知识组织与
知识管理

信息集成中的字符串匹配技术研究*

孙海霞 成 颖

（南京大学信息管理系 南京 210093）

【摘要】 匹配是信息集成的核心技术之一。论述基于编辑距离、基于标记以及 N 元文法等为代表的字符串匹配技术的研究现状，指出其存在的不足并提出改进思路。

【关键词】 匹配 信息集成 字符串匹配 【分类号】 TP393

Study on String – based Matching of Information Integration

Sun Haixia Cheng Ying

(*Department of Information Management*, *Nanjing University*, *Nanjing 210093*, *China*)

【**Abstract**】 Matching is one of the most important techniques of information integration. In this paper, string – based matching algorithms, mainly distance – based, token – based and the N – gram are elucidated. The deficiencies and research directions are also outlined.

【**Keywords**】 Matching Information integration String – based matching

信息集成是指综合运用查询处理、中间件、包装器等技术，把相互关联的分布式异构信息源集成在一起，实现变异构为同构，最终实现信息语义的统一[1,2]，从而有效地实现信息的共享[3]。在信息集成中，一个关键问题就是不同数据源模式中对等实体的识别，即匹配问题[4,5]。

1 匹配的内涵

匹配的目标是发现不同模式结构相关实体之间的映射关系，基本思想是：首先通过对标识结点的分析推得相关结点间的对应关系，然后根据获得的对应关系，通过运用各种筛选技术（Filtering）来确定最终的映射集。

一般匹配系统中输入的是两个分布在不同信息源中的实体（如表格、XML 元素、属性、规则、断言等），而输出的则是这些实体间所蕴含的关系（如相等、包含、兼容等）[4]，其结果可以采用一个四元组的映射 < id, e, e′, n > 进行描述。其中，n 是反映两个模式不同实体（e, e′）间的对应关系的相似系数（Coefficients），其大小直接反映了两个模式不同实体间的相似程度。当 n 为 1 时，表示 e 和 e′完全相同；当 n 为 0 时，表示 e 和 e′完全不同[4,5]，n 的值域为[0,1]。

至今，人们已经提出了多种匹配算法，每种算法都各有优缺点。许多匹配系统都是对它们的综合使用，如 Cupid[6]、COMA[7]、S – Match[8]、SF（Similarity Flooding）[9]、Artemis、OLA 等[4]。Shvaiko 和 Euzenat[4] 吸收了 Rahm 和 Bernstein[5] 的分类思想，提出了基于粒度/输入解释的分类（Granularity/Input Interpretation Classification）和基于输入信息类别的分类（Kind of Input Classification）的综合分类体系。下面，笔者将对 Shvaiko & Euzenat 分类体系中元素层匹配算法中的基于字符串的匹配技术进行分析，具体包括基于编辑距离、基于标记和 N 元文法的匹配算法。

2 字符串匹配技术

字符串匹配技术的基本思想是：将字符串看作是字符序列，字符串间相同的字符越多，则表明这两个字符串越相似，即两个实体间的相似系数就越大。字符串匹配

收稿日期：2007 – 06 – 01
收修改稿日期：2007 – 06 – 11
* 本文系南京大学人文社会科学项目“网络环境下异构信息检索标准体系研究”的研究成果之一。

技术本身可以细分为词缀匹配法(前缀匹配法和后缀匹配法)、基于编辑距离的匹配法、N 元文法以及基于标记的相似度法[10]等。

2.1 词缀匹配法

词缀匹配法是字符串匹配中最简单的方法,可分为前缀匹配法和后缀匹配法。前缀匹配法(Prefix)[4,12,13]的基本思想是:先输入源字符串和目标字符串,然后检测源字符串是否以目标字符串开头。如果是,则返回相等关系,或相似性系数为 1;否则返回 idk(I don't know) 或相似性系数为 0。

后缀匹配法(Suffix)[4,12,13]的基本原理和前缀匹配法相似,不同的是它检测的是源字符串是否以目标字符串为结尾。如果是,则返回相等关系,或相似性系数为 1;否则返回 idk 或相似性系数为 0。

在匹配同根字符串的应用中,词缀匹配法非常有效。但要注意的是,词语形式上的一致不等于其语义上的一致性,如单词 hot 与 hotmail, myself 与 yourself 等;同理,词语形式上的不一致不等于其在语义上的不一致。因此,需通过其它匹配技术来进一步确认,以提高准确率,这也是出现混合式匹配系统(Hybrid Matcher)和组合式匹配系统(Composite Matcher)的原因之一[6,7]。在实际应用中,无论是前缀法还是后缀法,都要考虑词缀的相对长度以及词缀在整个字符串中的比例,通过设置阈值来控制结果的输出。

2.2 基于编辑距离(Edit Distance)的匹配法[4,12]

编辑距离即将一个字符串转换成另一个字符串所需要进行插入、删除、替换等相关编辑操作的次数。基于编辑距离的匹配法是先输入两个字符串,然后计算两个字符串间的编辑距离,计为 d。m 取较长的字符串的长度,即 m = Max(字符串 1,字符串 2),n 为 d 与 m 的比值,即:n = d/m。当 n 小于阈值时,则返回相等关系或 1,否则返回 idk 或 0。

在基于编辑距离的匹配法中,编辑距离的计算是关键,编辑距离计算的准确与否直接关系到匹配结果准确率的高低,实践中通常采用基于矩阵的方法进行计算。设有字符串 s, t, s_i表示字符串 s 中的第 i 个字符,t_j 表示字符串 t 中的第 j 个字符,D(s, t, i, j)则表示 s 中前 i 个子串和 t 中前 j 个子串间的编辑距离,是矩阵 D 中的一个元素。置初值 D(s,t,0,0)为 0,则有下列计算公式(Levenshtein Distance 算法)[10]:

$$D(s,t,i,j)=\mathrm{Min}\begin{cases} D(s,t,i-1,j-1) & s_i \text{ 与 } t_j \text{ 相同时} \\ D(s,t,i-1,j-1)+1 & \text{在 s 中用 } t_j \text{ 替换 } s_i \text{ 时} \\ D(s,t,i,j-1)+1 & \text{在 s 中插入字符 } t_j \text{ 时} \\ D(s,t,i-1,j)+1 & \text{在 s 中删除字符 } s_i \text{ 时} \end{cases} \quad (1)$$

显然,这是一个递归计算过程,可以通过动态规划技术来实现。

Sellers 算法是对 Levenshtein 算法的改进,也称为 Needleman – Wunch Distance,其主要的改进是修改了权值的计算方法,即在对字符串进行替换、插入和删除时,每操作一次的权值 W 并不固定为 1,而是根据相关参数来确定。只有当 W 被赋予 1 时,才和 Levenshtein 算法一致。

Smith 和 Waterman[15]则在 Levenshtein 算法的基础上,提出了从不同字符串间求最大相同子字符串的算法:

$$D_{i,j}=\mathrm{Max}\begin{cases} D_{i-1,j-1}+s(s_i,t_j) & s_i \text{ 与 } t_j \text{ 相同时} \\ \mathrm{Max}(D_{i-k,j}-W_k) & \text{以 } s_i \text{ 为结尾的 k 个字符被删除或插入时} \\ \mathrm{Max}(D_{i,j-1}-W_q) & \text{以 } t_i \text{ 为结尾的 q 个字符被删除或插入时} \\ 0 & \text{确保结果非负,此时无相同子串} \end{cases} \quad (2)$$

其中,$s(s_i,t_j)$是序列元素间的相似值,W 是进行相关操作的权值,通常建立在先前统计的基础上,k 是进行相关操作的子字符串的长度,$D_{i,j}$表示以 s_i,t_j 为结尾的子串间的最大相似度值,k = i≤length(s),1≤j≤length(t)。具有最大相似度值 $D_{i,j}$的子串便是目标相似子串。如 $D_{i,j}$为 0,表示字符串 s 与 t 间没有相似子串。

Jaro[16]提出的 Jaro 矩阵法是另一种有效的相似度值计算方法,该算法考虑到字符串间相同字符的顺序位置与个数,其表达式如下:

$$\mathrm{Jaro}(s,t)=\frac{1}{3}\times\left[\frac{|s'|}{|s|}+\frac{|t'|}{|t|}+\frac{|s'|-Ts',t'/2}{s'}\right] \quad (3)$$

其中,s′是 s 中与 t 中的一个子字符串匹配的字符串,t′是 t 中与 s 中的一个子字符串匹配的字符串,|s′|,|s|,|t′|,|t|分别表示各自的串长,Ts′,t′是将 s′转换成 t′所需要操作的次数。在 Jaro 矩阵 M 中,如 s_i 与 t_j 相同,且两者距离足够近,即|i – j|小于 Min(|s|,|t|)/2,则 $M_{i,j}$的值为 1,否则为 0。

William Winkler 在 Jaro 矩阵法的基础上对含有相同前缀的匹配字符串的权值计算进行了修正,提出了 Jaro – Winkler 矩阵法,使得最终的输出结果值为:

JaroWinklerScore(s,t) = JaroScore(s,t) + (prefixLength × PREFIXSCALE × (1.0 – JaroScore(s,t))) (4)

其中,prefixLength 是相同前缀的长度,PREFIXSCALE 是一个常量,可根据相同前缀的长度进行调整。

2.3 基于标记(Token – based)的相似度法[4,8,10]

基于标记的相似度匹配法的基本思想是:将源字符串和目标字符串分别看成是由非停用词构成的文献向量 S 和 T,然后计算出两向量间的相似度值 Sim,当 Sim 大于系统设定的阈值时,则这两个字符串是相匹配的,系统返回相等关系或 1,否则不相匹配,返回 idk 或 0。

该算法思想的核心是标记相似度值 Sim 的计算,它直接决定着匹配结果的准确率。在信息检索领域中,计算相似度的方法有很多,如基于项匹配个数的、基于"距

离”的、基于概率的、内积法以及余弦相似度法等。

Matching 系数法(Matching Coefficient)是基于项匹配个数的最简单的一种算法,直接把 S 和 T 中相同的词项(Term)数作为相似度值。Dice 系数法(Dice Coefficient)对相似度值的定义作了改进,把相似度值限定在 0 - 1 间:

$$Sim(S, T) = \frac{2|S \cap T|}{|S| + |T|} \quad (5)$$

Dice 系数法提高了相似度值的精确率,但其在针对具有较少相同词项的匹配问题时,就显得不足,Jaccard 系数法(Jaccard Coefficient)填补了这一缺陷,其算法公式为:

$$Sim(S,T) = \frac{|S \cap T|}{|S| + |T| - |S \cap T|} \quad (6)$$

Overlap 系数法(Overlap Coefficient)与 Dice 系数法相近,将相似度值定义为 $|S \cap T|$ 和 $Min\{|S|, |T|\}$ 的比值。当一个字符串是另一个字符串的子串时,该法较为实用。

公式(5)、(6)中:$|S \cap T|$ 表示源字符串向量 S 和目标字符串 T 中包含的相同非零项的个数,$|S|$ 表示向量 S 中非零项的个数,$|T|$ 表示向量 T 中非零项的个数。

内积相似度法(Inner Product)和余弦相似度法(Cosine Coefficient)都引入了权重的思想,弥补了上述算法中不能区分不同词项在不同文献中重要程度的缺陷。内积相似度法算法可用下面的公式来表达:

$$Sim(S, T) = \sum_{i=1}^{n} SW_i \times TW_i \quad (7)$$

余弦相似度法(Cosine Coefficient)实际上是对内积相似度法的一种规范,其规范化的基础是向量的欧氏长度(L_2)[13](其本身即是一种基于距离的计算相似度值的方法),在实践中使用较为广泛。其算法公式为:

$$Sim(S,T) = \frac{\sum_{i=1}^{k} SW_i \times TW_i}{\sqrt{\sum_{i=1}^{k} SW_i^2} \times \sqrt{\sum_{i=1}^{k} TW_i^2}} \quad (8)$$

公式(7)、(8)中:SW_i 表示词项 W_i 在源字符串向量 S 中的权重,TW_i 表示词项 W_i 在目标字符串向量 T 中的权重,W_i 是两个字符串中的相同词项,n 表示源字符串和目标字符串中相同词项的个数,k 表示向量 S 和 T 的维数。

内积相似度法和余弦相似度法中权重的选取是关键。目前,权值计算的方式主要有:

(1)1 - 0 法,即词项出现,权值为 1,否则为 0。该加权法的缺陷在于否认了词项出现的次数对权值的影响。

(2)按词项出现的次数来确定,分别记为 0,1,2,…。此方法的缺陷在于没有考虑到词项出现的次数还与字符串的长短有关。

(3)TF - IDF 加权法:$weight_w = TF_w \times IDF_w$,这是针对前两种方法存在的缺陷提出来的,考虑到了词串的长短对此项频率的影响以及区分词的重要性。其中:

TF_w 表示词项 w 在字符串中出现的频率,其大小反映了 w 对整个字符串的重要性程度,值越大,越重要。

DF_w 表示含有词项 w 的字符串个数,其值反映了 w 在衡量字符串间相似性时作用的大小,值越大,作用越小。

IDF_w 与 DF_w 成反比关系,$idf_w = \log\left[\frac{N}{id_w}\right]$,其大小反映了 w 对字符串进行区别时的重要性。

在对简单的名称字符串进行匹配时,可以使用 Jaccard 系数法,它不仅实现起来简单,而且准确率也较高。但在对复杂名称、长字符串甚至实例进行匹配时,实践证明采用余弦相似度法与 TF - IDF 加权法相结合会更好。其中,可以根据具体的信息集成对象或匹配对象采用具体的方式(如最大 tf 规范、对数 tf 规范和余弦规范化等[14])对 tf 进行规范,以提高余弦相似度法的准确性。

Raymond Mooney[10] 给出了另外两种基于 TF - IDF 加权法的名称间相似度计算方法,即 IF - IDF 法和 SoftTF - IDF 法:

$$IF - IDF(S,T) = \sum_{w \in S \cap T} V(w,S) \times V(w,T) \quad (9)$$

$V(w,S) = V'(w,S) / \sqrt{\sum_{w'} V'(w,S^2)}$,$V'(w,S) = \log(TF_{w,s} + 1) \times \log(IDF_w)$,$V(w,T)$ 的定义同上。$TF_{w,s}$ 表示词项 w 在词串 S 中出现的频率,IDF_w 表示包含词项 w 的字符串的逆文献频率。

$$SoftTF - IDF(S,T) = \sum_{w \in CLOSE(\theta,S,T)} V(w,S) \times V(w,T) \times N(w,T) \quad (10)$$

公式(10)是对公式(9)的改进,认为两个字符串间的相似性程度不仅受相同词项的影响,而且还受相似词项的影响。设 Sim(w, v)是求短字符串间相似度量值的函数,如使用 Jaro 矩阵法,$w \in S$, $v \in T$;则 CLOSE(θ,S,T)是满足 Sim(w, v)大于阈值 θ 的词项 w 的集合,$N(w, T) = MAX_{w \in T} Sim(w, v)$。

基于概率的相似度计算法是建立在信息熵统计的基础上的,其中,文献向量中的元素是字符串中非停用词在文献中出现的概率,其基本原理是不同字符串间的匹配程度可决定它们之间信息熵差的大小 β(S,T)。即字符串 S 和 T 之间的相似度值可定义为[14]:

$$Sim(S,T) = 1 - \beta(S,T) \quad (11)$$

这一类的算法研究成果也很多,如 Bhattacharyya 法等[21]。

2.4 N 元匹配法 (N - gram)[4,12,17]

将文本 W 看成是一线性字符序列,将长度为 n 的窗口从文本的第一个字符处开始,自左向右连续移动,每次移动的步长为 1 个字符,窗口中出现的 n 个字符即为 n -

gram。也就是说,将文本 W 分解成若干个长度为 n 的子字符串 $w_1, w_2, w_3 \cdots$ 所组成的向量空间 D。如 W 的三元向量空间 D 可定义为:$D = \{w_{aaa}\ w_{aab}\ w_{aac}\ w_{abc}\ w_{abb} \cdots w_{acc}\}$,其中,$w_{aaa}$表示字符串 aaa 在向量空间中出现的频率。当 n = 3 时,英文 informational 的三元语法为:{inf nfo for orm rma mat ati tio ion ona nal },其向量空间 D 为 = {1,1,1,1,1,1,1,1,1,1,1,1}。

目前,N - gram 法在匹配技术中运用较为广泛,基本思想[9]是:先输入两个字符串,即源字符串 s 和目标字符串 t,列出源字符串和目标字符串所有的 n - gram,然后计算同现的 n - gram 的频率 ΣCst(n - gram),ΣCs(n - gram)为源字符串 s 中 n - gram 的数量,ΣCt(n - gram)为目标字符串中 n - gram 的数量,则可以计算 m:

$$m = \Sigma Cst(n-gram) / MAX(\Sigma Cs(n-gram), \Sigma Ct(n-gram)) \quad (12)$$

最后将 m 与设定的阈值比较,如果大于阈值,则返回相等关系或 1,否则返回 idk 或 0。

HYLTON[18]的研究中采用的 N - gram 匹配法与上述算法稍有区别,其基本思想为:输入两个字符串 W_1 和 W_2,获得 W_1 和 W_2 的 n - gram 向量空间 D_1 和 D_2,然后根据差异向量空间 $|D'| = D_1 - D_2$ 计算出两个向量空间的差异值 d,再将差异值 d 与阈值 t 相比较,如果 d 小于 t,则表明 W_1 和 W_2 是相似的。

差异值 d 的计算公式:

$$d = \sqrt{\sum_{i}^{n} (w_{1i} - w_{2i})^2} \quad (13)$$

阈值 t 的计算公式为:T = 2.486 + 0.025n

w_{1i}和 w_{2i}是 W_1 和 W_2 的 n - gram 子字符串,n 表示向量空间的最大维数,2.486 是一个常量,可根据具体的系统设定。

以 room1 和 room2 为例,它们的 Tri - gram 向量空间分别为:D_1 = { roo oom om1 } = {1,1,1,0} 和 D_2 = { roo oom om2} = {1,1,0,1},它们的差异向量空间为:$|D'| = D_1 - D_2$ = {om1 om2},$d = \sqrt{1^2 + 1^2} = 1.414$,阈值 t = 2.486 + 0.025 × 4 = 2.536。因差异值 d 小于阈值 t,所以可判定这两个字符串是相似的。

3 结 语

基于编辑距离的匹配法对 Wordnet[19]中未登录词(Unknown Word)和缩写词的匹配较为有效,一般在数据库和 XML 模式匹配中用的较多。但也会出现歧义等问题,如 Cool 与 Cook,根据前面的阈值进行计算,返回的是相等关系,但它们实质上是无关的;而 Hill 与 Mountain 在一定程度上却是相似的。

基于 N - gram 的匹配法与基于编辑距离的匹配法一样,对简写词、缩写词和形态上有细微差别的字符串进行匹配时较为有效,目前,被广泛用于针对数据库和 XML 模式的匹配系统。但大量实验证明,与基于编辑距离的匹配法相比,N - gram 法的结果较为“悲观”。表 1 是 S - Match[8]系统对这两种算法的比较。

表 1 N - gram 与基于编辑距离的匹配法的比较

Match name	Source label	Target label	Relation	Similarity value
Edit distance	fin	find	=	0.75
N - gram	fin	find	idk	0.54
Edit distance	PO	PO1	=	0.66
N - gram	PO	PO1	idk	0.5
Edit distance	impact	compact	=	0.71
N - gram	impact	compact	idk	0.58

基于标记的匹配法的优势在于:不像基于编辑距离的匹配法和 N 元文法那样受字符串顺序的约束,而且基于 TF - IDF 加权法的相似度计算法还同时考虑到了词频、字符串的长度和区分词之间的关系,有效提高了相似度计算的准确性,进而提高了匹配的准确性。例如,要求识别字符串 S,T,W 之间的匹配关系,S = “name of the student”, T = “age of the student”,W = “student_name”。如果用 Jaccard 相似度计算法,可得结果:<1,S,T,1>,<2,S,W,0>;用基于 TF - IDF 加权法的相似度计算法则可得结果:<1,S,T,0>,<2,S,T,0>。显然,后者的结果是正确的。

从上面几种字符串匹配方法中可以看到,字符串匹配方法实际上是以词形为切入点的,它实现的是名称或名称描述在形式上的匹配,因此实现起来较为简单;但字符串匹配技术过于“机械”,即使在后来的研究中提出了权重的概念,但依然是建立在词项本身的基础上。自然语言中大量同义词、多义词的存在,导致了不同实体的命名即使在词形上存在对应关系也不等于其语义上也存在相应的对应关系,因此,仅仅根据字符串匹配方法所获得的匹配结果是不可靠的,还应综合考虑这些实体的其它相关属性,其中最重要的是结构属性,也就是说,要考虑这些实体在不同模式中的具体位置信息,即语境信息。因此,字符串匹配方法需要结合其它匹配方法,诸如结构匹配方法(Structure - based)或各种基于语义的匹配方法(Semantic Matching)等,从而实现对匹配结果进行进一步的考证与综合应用,以提高准确率[12]。因此,如何从自然语言处理的角度,综合运用查询优化、相关反馈等技术,从实体概念的本身,也就是从语法、语义以及语用的

角度进一步完善和发展信息集成的匹配技术,应该是信息集成研究的一个重要发展方向[4,12,20]。

参考文献:

[1] 陈跃国,王京春. 数据集成综述[J]. 计算机科学, 2004,31(5):48 - 51.

[2] Maurizio L. Data Integration: A Theoretical Perspective[C]. In: Proc. of the ACM SIGACT—SIGMOD —SIGART Symposium on Principles of Database Systems,2002:233 - 246.

[3] 吴昊,邢桂芬. 基于本体的信息集成技术研究[J]. 计算机应用,2005,25(2):456 - 458.

[4] Shvaiko P,Euzenat J. A survey of Schema - based Matching Approaches[J]. Journal on Data Semantics,LNCS 3730,2005:146 - 171.

[5] Rahm E,Bernstein P. A Survey of Approaches to Automatic Schema Matching[J]. The International Journal on Very Large Data Bases (VLDB),2001,10(4):334 - 350.

[6] Madhavan J,Bernstein P,Rahm E. Generic Schema Matching With Cupid[C]. In: Proceedings of the Very Large Data Bases Conference (VLDB),2001:49 - 58.

[7] Do H H,Rahm E. COMA - A System for Flexible Combination of Schema Matching Approaches[C]. In: Proceedings of the Very Large Data Bases Conference (VLDB), 2001: 610 - 621.

[8] Giunchiglia F,Shvaiko P,Yatskevich M. S - Match: An Algorithm and an Implementation of Semantic Matching[C]. In: Proceedings of the European Semantic Web Symposium (ESWS),2004: 61 - 75.

[9] Melnik S,Garcia - Molina H,Rahm E. Similarity Flooding: A Versatile Graph Matching Algorithm[C]. In: Proceedings of the International Conference on Data Engineering (ICDE), 2002:117 - 128.

[10] Ilenko B,Cohenw M R,et al. Adaptive Name Matching in Information Integration [J]. IEEE Intelligent Systems, 2003,18 (5):16 - 23.

[11] Geng J F, Yang J. AutoBib: Automatic Extraction and Integration of Bibliographic Information on the Web[C]. In: Proceedings of the 29th VLDB Conference. Berlin, Germany, 2003:193 - 204.

[12] Giunchiglia F, Yatskevich M. Element Level Semantic Matching [C]. In: Proceedings of Meaning Coordination and Negotiation Workshop at the International Semantic Web Conference (ISWC), 2004:61 - 75

[13] Giunchiglia F,Shvaiko P,Yatskevich M. Semantic Schema Matching [R]. Technical Report DIT - 05 - 014, University of Trento, 2005:347 - 365.

[14] 孙建军,成颖. 信息检索技术[M]. 北京:科学出版社. 2004: 53 - 71,232 - 242.

[15] Smith F, Waterman M S. Identification of Common Molecular Subsequences[J]. Journal of Molecular Biology,1981(147): 195 - 197.

[16] Jaro M A. Advances in Record Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida[J]. Journal of American Statistical Association, 1989,86(406):414 - 420.

[17] 程国达,邹亚会,朱静. 一种自适应信息集成方法[J]. 计算机应用, 2005,25(3):666 - 669.

[18] Hylton J A. Identifying and Merging Related Bibliographic Records [D]. MIT Institute of Technology,1996.

[19] Miller A G. WordNet: A Lexical Database for English[J]. Communications of the ACM,1995,38(11):39 - 41.

[20] Madhavan J, Bernstein P, Doan A, et al. Corpus - based Schema Matching[C]. In: Proceedings of the International Conference on Data Engineering (ICDE),2005:57 - 68.

[21] Similarity Metrics[EB/OL]. [2007 - 01 - 10]. http://www.dcs.shef.ac.uk/~sam/stringmetrics.html.

(作者 E - mail:sunyiqin1984@yahoo.com.cn)