数据集成中数据项与数据元匹配算法①

文必龙，付 玥

(东北石油大学 计算机与信息技术学院，大庆 163318)

摘 要：近年来，随着数据元标准的建立，数据元在各行各业的数据集成过程中担任着重要角色，用于规范数据库、报表、文档中的数据项，实现各种数据源之间的映射。分析数据元的结构，提出一种数据项与数据元匹配算法，该算法基于编辑距离算法，融合最长公共子序列、权重、词语重心后移等思想，实现数据项与数据元字典中数据元的相似度计算，利用排列组合原理对匹配速度进行优化。以中石化标准数据元为实验数据进行实验，验证了该匹配算法的有效性。

关键词：编辑距离；最长公共子序列；相似度计算；数据元；权重

Matching Algorithm Between Data Item and Data Element During Data Integration

WEN Bi-Long, FU Yue

(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: In recent years, with the establishment of data element standard, data element plays important role during data integration in many enterprises. Data element may standardize dataitems of databases, reports and documents. It may help mapping between data sources. Analyzing the compositions of data element and putting forward a kind of matching algorithm between dataitem and data element. The matching algorithm is based on levenshtein distance and fused the thought of longest common subsequence, weight and backward focus. It realizes similarity calculation between dataitem and data element of data element dictionary. It uses the permutation and combination principle to optimize matching speed. The experiments have proved that the matching algorithm was right through using the standard dataitems of China Petroleum and Chemical data element dictionary as experimental data.

Key words: levenshtein distance; longest common subsequence; similarity computation; data element; weight

随着企业的不断进步与发展，企业内部数据的存储和表示呈现出分布性、异构性的特点，不仅包括企业内、外关系数据库等传统结构化数据，还包括 Excel、Xml、Html 等半结构化数据，以及声音、图像、视频等非结构化数据。由于各种类型数据的分布位置不同，数据源采用的存储方式和表现形式不同，需要使用不同的概念、属性、关系来描述数据[1]。不同业务数据信息系统虽然能够满足业务数据存储和管理要求，但在许多情况下，这些信息系统已经开始制约企业的数据共享，为避免企业数据在信息化建设中形生“数据孤岛”，很多企业都开始实施数据集成，并把它当作首要解决的问题。数据集成可以把不同来源、格式、特点性质的数据在逻辑上或物理上有机地集中[2]，实现企业数据共享，为“数据孤岛”架一座桥梁。

数据集成过程中，数据层面上的数据异构是导致系统之间难以实现数据共享的主要原因，包括：名称不一致、定义不一致、数据类型不一致、数据表示不一致等等[3]。为解决数据层面上的数据异构，需要确立不同数据源之间数据项的对应关系以实现两者之间映射。传统的解决方式是手工建立一个同义词库，在同义词库中记录数据项之间的对应关系，当数据项增加或者改变时就要同时更新同义词库，十分麻烦。

① 基金项目:国家科技重大专项(2008ZX05023-05-05)
收稿时间:2011-07-08;收到修改稿时间:2011-07-30

随着数据元标准的建立，数据元在企业数据集成过程中的地位变得非常重要。数据元作为企业数据的标准，通过分别计算不同数据源中数据项与数据元的相似度，完成数据项与数据元的匹配，就可以自动匹配数据源之间的数据项，实现双方数据项之间的映射，方便企业数据集成的实施。

1 数据元分析

数据元是用一组属性描述其定义、标识、表示和允许值的不可再分的基本数据单元。

数据元可以看作由以下四部分组成：对象类、特性、表示、限定词。①对象类是现实世界中的想法、抽象概念或事物的集合，其特性和行为遵循同样的规则而能够加以标识，例如原油、油相等；②特性是对象类的所有个体所共有的可以区别于其他成员的某种性质，例如原油密度、油相渗透率等；③表示是值域、数据类型的组合，必要时也包括度量单位或字符集等；④限定词是帮助定义和呈递唯一性概念的术语，在数据元语义结构中处于被限定主要成份的前面[4]。表 1 显示数据元组成成分示例，“[]”符号内的词为限定词。

表 1 数据元组成成分

序号	名称	对象类词	特性词	表示词
1	石油储量	石油	储(藏)	量
2	原油密度	原油	密度	(值)
3	油相渗透率	油相	渗透率	(率)
4	热采单元年产油总量	[热采]单元	[年]产油	[总]量

2 基于字面相似匹配方法

基于字面相似的计算方法是根据字面相似性原理，即汉语中绝大多数同义词、相近词都含有相同语素这一突出特点，通过计算含有相同语素的个数以及它们在各词中的位置来确定两者之间的关联程度[5]。

2.1 编辑距离算法

编辑距离(Levenshtein Distance)由 Levenshtein 于 1966 年在文献中提出，通过编辑距离计算源字符串 S 与目标字符串 T 的相似度[6]。方法为：对于两个字符串 S、T，将 S 转换成 T 所需要的操作步骤的总数量（删除、插入、替换）叫做从 S 到 T 的编辑路径，所有编辑路径中最短的编辑路径就是字符串 S 与字符串 T 的编辑距离，编辑距离越小则表示两个字符串的相似度越高。

设 n、m 分别表示源字符串 S（s1…si…sn）和目标字符串 T（t1…tj…tm）的长度，编辑距离 LD 的计算方法为：

$$LD=\begin{cases}0, \text{当} n=0 \text{且} m=0 \text{时}\\ n, \text{当} n>0 \text{且} m=0 \text{时}\\ m, \text{当} n=0 \text{且} m>0 \text{时}\\ D(n,m), \text{当} n>0 \text{且} m>0 \text{时}\end{cases}$$

其中 D(n，m)为（n+1）*（m+1）阶矩阵：

$$D(i,j)=\begin{cases}j, i=0\\ i, j=0\\ \min\begin{cases}D(i-1,j)+1\\ D(i,j-1)+1\\ D(i-1,j-1)+cost\end{cases}, \text{其它}\end{cases}$$

cost 表示 S 中第 i 个字符 si 转换到 T 中第 j 个字符 tj 所需要的操作次数，如果 si 与 tj 相等，则不需要任何操作，此时 cost=0；否则，需要替换操作，此时 cost=1。

2.2 加权相似度算法

在词汇字面相似度计算上，王源提出在计算中考虑匹配字数和词汇结构这两方面因素，宋明亮利用词汇字面相似性原理计算词间相似度进行词汇归类[7]，吴志强在前面两种思想基础上提出了加权相似度算法，将汉语重心后移的思想加入其中。

加权相似度算法主要内容为：表达某一具体专指概念的词语，该词语的中心词往往在词语的后半部分，组成该词语的语素越靠后，它所起的作用就越大，基于此，对词语中各个语素进行量化和加权处理，语素越靠前，它的作用越小，为它分配的权值也越小，语素越靠后，它的作用越大，为它分配的权值也越大[8]。

3 算法详解

3.1 算法设计

对数据元组成成分与结构进行分析，符合重心后移的特点，所以在数据项与数据元匹配的过程中主要坚持三项原则：两者在词语结构上越相近，相似度越高；两者包含相同语素越多，相似度越高；两者包含的相同语素的位置越靠后，相似度越高。

按照这三项原则，制定的具体算法为：

(1) 通过距离编辑算法，计算数据项与数据元在结构上的相似程度，并且将结果转换为 0、1 之间的数。

(2) 查找数据项与数据元包含相同语素的个数。

(3) 为数据项与数据元的每个语素按照重心后移的原则分配权重，分别计算这两个词语的权重总和。

(4) 查找数据项与数据元的最长公共子序列 LCS，按照为语素分配的权重值，分别计算它们的最长公共子序列中包含语素的权重和。

(5) 通过大量的实验，确定公式中的每个系数。

3.2 算法实现

本文以中石化数据元字典中 14000 个数据元作为实验数据，在数据元字典中查找与某一数据项相似的数据元。选取以数据项“层位日产气量”作为源字符串 S、以数据元“层位日产水量”作为目标字符串 T 为例，具体实现过程如下：

(1) 首先，通过编辑距离的计算方法，得到数据项“层位日产气量”与数据元“层位日产水量”的编辑距离 LD，设 S 与 T 的最长长度为 len =max（n，m），则参数 1 为 $parameter1=\frac{len-LD}{len}$：其中：

$$D(i,j)=\begin{pmatrix}0&1&2&3&4&5&6\\1&0&1&2&3&4&5\\2&1&0&1&2&3&4\\3&2&1&0&1&2&3\\4&3&2&1&0&1&2\\5&4&3&2&1&1&2\\6&5&4&3&2&2&1\end{pmatrix},$$

LD=D（6,6）=1，

len=max（6,6）=6，

$parameter1=\frac{6-1}{6}=\frac{5}{6}$。

(2) 查找 S 与 T 这两个词语包含的相同的语素的个数 num，具体方法为：设参数 mark=0，逐一取 S 中的语素，比较在 T 中是否包含该语素，如果在 T 中包含，则 mark=mark+1，将 mark 的最终结果赋给 num。字符串 S 与 T 包含的相同语素为{层，位，日，产，量}，则 num=5。

(3) 为 S 与 T 的每个语素按照重心后移的原则分配权重，S 中（s1…si…sn）的权重分别为（1…i…n），T 中（t1…tj…tm）的权重为（1…j…m），分别计算 S 与 T 中各语素的权重之和 QS、QT：

$$S=\begin{pmatrix}层&位&日&产&气&量\\1&2&3&4&5&6\end{pmatrix},$$

$$T=\begin{pmatrix}层&位&日&产&水&量\\1&2&3&4&5&6\end{pmatrix},$$

QS=1+2+3+4+5+6=21，

QT=1+2+3+4+5+6=21。

(4) 查找 S 与 T 的最长公共子序列 LCS，依据的原理为：如果设 L（n，m）为 n*m 阶矩阵，判断 si 与 tj 是否相同，相同则 L(i-1，j-1)=1，那么在 S 与 T 所包含的相同语素的个数不为 0 的情况下，总可以找到一条值为 1 的不连续的最长对角线，这条对角线对应语素组成的序列就是 S 与 T 的最长公共子序列：

$$L(i,j)=\begin{pmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&1\end{pmatrix}。$$

根据这一原理，制定的具体算法为：定义一个结构体 LCS{C1，C2，C3，C4，C5}，C1 记录行号值，C2 记录列号值，C3 记录 S 中子串（s1…si）与 T 中子串（t1…tj）的最长公共子序列 lsc 的长度，C4 记录 lsc 中语素在 S 中对应位置的权重和，C5 记录 lsc 中语素在 T 中对应位置的权重和。当 si 与 tj 相同时，在结构体 LCS（C1，C2，C3，C4，C5）中查找满足条件（C1<i 并且 C2<j）的 C3 的最大取值所对应的位置，设该位置为 LCS [x]，设此时结构体的位置为 LCS [now]，将 i 值赋值给 LCS [now].C1，将 j 值赋值给 LCS [now].C2，LCS [x].C3 加 1 的值赋值给 LCS [now].C3，LCS [x].C4 加 i 的值赋值给 LCS [now].C4，LCS [x].C5 加 i 的值赋值给 LCS [now].C5：

LCS[0]={0，0，0，0，0}，

LCS[1]={1，1，1，1，1}，

LCS[2]={2，2，2，3，3}，

LCS[3]={3，3，3，6，6}，

LCS[4]={4，4，4，10，10}，

LCS[5]={6，6 ，5，16，16}。

由 LSC[5]得到 S 与 T 的最长公共子序列的长度值为 5，那么计算此公共子序列在 S 中对应的权重和为 QS（LCS）=1+2+3+4+6=16、在 T 中对应的权重和为 QT（LCS）=1+2+3+4+6=16。由此得到第二个参数：

$$parameter2=\frac{num}{n}*\frac{Q_S(LCS)/Q_S+Q_T(LCS)/Q_T}{2},$$

$$parameter2=\frac{5}{6}*\frac{16/21+16/21}{2}。$$

(5) 通过不同的系数取值以及大量的实验，确定系数 A 与 B 的取值，得到 S 与 T 的相似度计算公式：P（S,T）=A*parameter1+B* parameter2，其中系数 A 与 B 的和为 1。

对系数 A、B 按照表 2 进行取值计算，发现在 A=0.4、B=0.6 与 A=0.3、B=0.7 时最满足预期结果，重新取值计算。

表 2 系数选取 1

A	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
B	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9

按照表 3 进行取值计算，发现在 A=0.35、B=0.65 时最满足预期匹配结果，最终确定为 A=0.35，B=0.65，P（层位日产气量,层位日产水量）= 0.7043650794。

表 3 系数选取 2

A	B	A	B	A	B
0.3	0.7	0.34	0.66	0.38	0.62
0.31	0. 69	0.35	0.65	0.39	0.61
0.32	0.68	0.36	0.64	0.4	0.6
0.33	0.67	0.37	0.63		

3.3 算法优化

在算法实现的过程中，采取以下两种方式用来提高算法的匹配效率：

(1) 在计算数据项 S 与数据元 T 编辑距离过程中，同时计算 S 与 T 包含的相同语素个数 num，以及计算 S 与 T 的最长公共子序列对应语素的权重和，这样，只需要创建一个（n+1）*（m+1）阶矩阵和一个二维循环就可以完成。通过将算法融合在一起，减少操作。

(2) 在数据项与所有数据元进行匹配之前，先将无关的数据元筛去，只留下与数据项有一定关系的数据元用于与数据项进行匹配，提高匹配速度。利用排列组合的原理，从 S 中提取出子串，将这些子串组成查询条件，用来筛选数据元。

通过上述操作实现匹配效率的提高。从包含 14000 个数据元的数据元字典中找出与数据项匹配的数据元，优化前所用时间（单位：ms）、后所用时间（单位：ms）以及匹配到的结果（单位：个）如表 4 所示：

表 4 优化前后时间对比

序号	数据项	优化前	优化后	结果
1	作业施工深度	1156.25	250.875	887
2	层位日产气量	985.61	163.25	580
3	气举压力	645.875	120.5	435
4	气体最小流量	570.25	103.375	405
5	注气泵压	250.5	55.75	189
6	拐点坐标	107.75	20.5	67

3.4 实验结果

数据项“层位日产气量”在中石化数据元字典中匹配到的前 20 个数据元以及相似度结果如表 5 所示：

表 5 优化前、后时间对比

序号	匹配数据元	匹配相似度
1	层位日产气量	1
2	层位日产水量	0.7043650794
3	层位日产液量	0.7043650794
4	层位日产油量	0.7043650794
5	试油层日产气量	0.6868551587
6	单井日产气量	0.6047619048
7	井段日产气量	0.6047619048
8	井组日产气量	0.6047619048
9	折算日产气量	0.6047619048
10	采气井日产气量	0.5559523810
11	单井初增日产气量	0.5171957672
12	单井设计日产气量	0.5171957672
13	单元标定日产气量	0.5171957672
14	实际平均日产气量	0.5171957672
15	试采单井日产气量	0.5171957672
16	试气单井日产气量	0.5171957672
17	预计单元日产气量	0.5171957672
18	单井报废前日产气量	0.4857142857
19	单井措施后日产气量	0.4857142857
20	单井措施前日产气量	0.4857142857

4 结语

以数据元作为媒介，在数据元字典中查找到与双方数据项匹配的数据元之后，直接选择相似度最高的数据元，通过数据元实现对数据项的规范，并且自动实现异构数据源中数据项之间的映射，降低

（下转第 231 页）

SCADE 自动完成，所以自动隐去了。

第二个输入（现在从外部看起来是第一个输入）的赋值根据算法来决定，在本算法中，sin 函数的泰勒级数第 0 级值为 x，所以这里也设置成了 x；BasicSeries 的计算也是在 x 的基础上，从第一级开始累加，这样做是为了避免除 0 的麻烦。

其它的输入都需要设置成数组，且数组大小须与迭代次数保持一致。

用户在使用时可以根据需求来决定迭代次数的设置和精度的设置。其它的用户不需要修改。

2.2 Cos_SCADE 等

其它几个三角函数、反三角函数的 SCADE Suite 实现方法与正弦函数类似，由于篇幅的原因，本文不再一一介绍。

3 测试

为了与编译器中自带的三角函数进行比较，专门在 SCADE Suite 中建立用于测试的节点，并利用 SCADE Suite 的仿真工具完成测试：

如,Compare_sin 节点是用于测试 sin 函数，并与 VC 库中的 sin 函数结果对比。

测试结果与 Vc 库函数结果相对比后，显示：

(1) 对于 sin、cos、tan：

使用泰勒级数法（迭代设置成最多 10 次的情况下）可保证精度，在输入为（-2PI，2PI）时，精度高处可达 E-17,精度低处也达到了 E-9。

(2) 对于 asin、acos：

使用泰勒级数法，在输入值比较小时精度高；在输入值大时，如在区间[0.95，1]时，精度低，在区间[E-7,E-1]。

(3) 对于 atan，泰勒级数法很难收敛，几乎不好用。

4 结语

这里仅介绍了使用 SCADE Suite 实现三角函数、反三减函数的泰勒级数展开法，其它方法并不需要在 SCADE Suite 环境下详细介绍。

每种算法都有各自的优势和劣势，用户需要根据需求和自己的情况决定使用何种算法，及怎么使用算法。本文尚有须完善之处，敬请提出宝贵意见。

参考文献

1 马士超,王贞松.基于 DSP 的三角函数快速计算.计算机工程,2005,31(22):12−14.

2 史万明,吴裕树,刘玉树.反正切函数的快速计算方法.北京理工大学学报,1995,15(15):6−9.

3 吴成富,王睿,陈怀民,等.基于 SCADE 实现的三余度飞控计算机系统任务同步.航空计算技术,2009,39(1):107−110.

4 张杰,宋志刚.基于模型的软件开发技术在型号软件研制中的应用.科学技术与工程,2008,8(15):4152−4157.

5 张合军,陈欣.基于 SCADE 的无人机自主导航飞行软件设计.计算机测量与控制,2007,15(10):1400−1402.

（上接第 243 页）

人为操作产生的误差，免去建立同义词库的需要，在节省时间与人力的同时，提高数据集成过程的效率与正确性。

参考文献

1 鱼滨,郑娅峰.基于本体的异构数据集成方法及其实现.计算机应用与软件,2007,24(9):30−32,65.

2 熊曾刚,张学敏,陈建新.基于 XML 的信息系统集成的研究.情报杂志,2005,6:25−27.

3 刘庆河,郝文宁,韩宪勇,陈兴建,吴可嘉.基于数据元的数据交换规范研究.电脑知识与技术,2010,6(10):2309−2310.

4 吴波,李建,伍东.数据元标准化在石油数据中的研究与实现.山西电子技术,2006,5:86−89.

5 章成志.一种基于语义体系的同义词识别研究.淮阴工学院学报,2004,13(1):59−62,67.

6 赵作鹏,尹志民,王潜平,许新征,江海峰.一种改进的编辑距离算法及其在数据处理中的应用.计算机应用,2009,29(2):424−426.

7 Lu Y, Hou HQ. Automatic recognition and mining of Chinese synonyms for information retrieval. Information Studies: Theory & Application, 2006,29(4):472−475.

8 朱毅华,侯汉清,沙印亭.计算机识别汉语同义词的两种算法比较和测评.中国图书馆学报,2002,4:82−85.