短文本聚合在元器件供方匹配中的应用与研究

魏自强,班元郎,徐　伟,王文玺
(贵州航天计量测试技术研究所,贵州 贵阳 550009)

摘　要:航天采购部门采购合格供方的元器件是保证航天用元器件可靠性的方法之一。确定供方是否在合格供方目录中,是航天元器件采购流程中的一个重要步骤。但由于航天各院所系统中对供方定义标准不一致,常以供方公司的别称、简称代替供方名称,这导致同一供方出现多种不同名称,这给如何匹配合格供方带来了挑战。针对航天各院所系统中的供方数据的特征,提出了一种结合 Jaro-Winkle 算法和 Levenshtein 算法的融合算法。该算法通过引入调整阈值及系数,将字符的位序、字符替换、添加、删除操作等因素纳入到供方名称的短文本相似度计算中,提高供方名称的短文本匹配准确率。通过在航天元器件合格供方匹配流程中的应用,该算法有效提高了供方的匹配准确率。

关键词:Jaro-Winkler 算法;Levenshtein 距离;短文本聚合模型;数据特征;供方匹配
中图分类号:TP315　　**文献标识码**:A　　**文章编号**:1673-629X(2022)07-0216-05
doi:10.3969/j.issn.1673-629X.2022.07.037

Application and Research of Short Text Aggregation in Component Supplier Matching

WEI Zi-qiang, BAN Yuan-lang, XU Wei, WANG Wen-xi
(Guizhou Aerospace Metrology and Testing Technology Research Institute, Guiyang 550009, China)

Abstract: The procurement of components from qualified suppliers by aerospace procurement department is one of the methods to ensure the reliability of aerospace components. Determining whether the supplier is in the list of qualified suppliers is an important step in the procurement process of aerospace components. However, due to the inconsistency of supplier definition standards in the systems of aerospace institutes, the supplier's nickname and abbreviation are often used to replace the supplier's name, leading to a variety of different names for the same supplier, which brings challenges to how to match qualified suppliers. According to the characteristics of supplier data in the systems of aerospace institutes, we propose a fusion algorithm combining Jaro-Winkle algorithm and Levenshtein algorithm. By introducing the adjustment threshold and coefficient, the algorithm integrates the character bit order, character replacement, addition, deletion and other factors into the short text similarity calculation of the supplier's name, so as to improve the short text matching accuracy of the supplier's name. Through the application in the qualified supplier matching process of aerospace components, the proposed algorithm can effectively improve the matching accuracy of suppliers.

Key words: Jaro-Winkler algorithm; Levenshtein distance; short text aggregation model; data characteristics; supplier matching

0　引　言

工业化和信息化的深度融合,信息技术在军工企业产业链中的应用越来越广。简洁的短文本,已然成为适应人们快速高效工作的信息载体[1]。例如元器件供方简称就常作为供方名称的短文本替代出现航天各个业务系统中。由于航天元器件信息来源于 ERP、TDM 等多个平台,其中元器件的厂商名称即供方名称是定义唯一一个元器件的标准之一。但各系统的供方定义标准不同、不同人员对供方数据的理解不同导致同一条供方数据出现多条不同的记录。最终导致在采购流程中,不同业务部门所提交的采购单中同一元器件供方信息数据出现不一致的情况。因此,在采购流程中,对供方数据和合格供方目录进行匹配是必要步骤之一。

供方数据和合格供方目录都是短文本数据。供方数据匹配可以看作是文本匹配问题。在对现有的 Jaro-Winkler 算法以及 Levenshtein(编辑距离)算法进行测试后,发现这两个算法在供方匹配应用中有各自的优势,但都不能很好地满足供方匹配需求。该文根据供方数据的特征,将 Jaro-Winkler 算法与 Levenshtein(编

收稿日期:2021-08-09　　**修回日期**:2021-12-09
基金项目:国防科工局基础科研项目(JSZL20191201ZL0002)
作者简介:魏自强(1990-),男,硕士研究生,研究方向为信息安全、数据挖掘。

辑距离)算法进行结合与改进。改进的算法结合了两种算法的优势,在计算元器件供方名称与合格供方目录的相似度时,提高了匹配的准确率,满足了供方匹配应用的需求。

1　研究现状

1.1　短文本聚合模型

加入短文本聚合模型的定义(1 到 2 句话)在短文本聚合模型中,采用相似度算法对文本进行相似度匹配。计算两个字符串相似度算法主要可分三类[2]:基于字面、基于语义、基于统计关联的相似度算法。基于字面的相似度算法有编辑距离的方法和相同字或词的方法,代表性的有 Jaro-Winkler[3-4]、Levenshtein[5-6]算法、最长公共子串算法[LCS][7-8]、余弦相似度算法[9]。文献[10]通过计算前后非相邻字符间的交换操作,改进了编辑距离算法,实现了编辑操作的最小化。

1.2　Jaro-Winkler 算法

Jaro-Winkler 算法是用来计算 2 个字符串的相似度,由 Jaro 改进而来。该算法适合计算两个较短的字符序列的相似度,运算结果在 0 到 1 范围内。0 表示完全不匹配,1 表示完全匹配[11],运算的值越大表示相似度越高。

(1)Jaro 算法。

$$d_j=\begin{cases}0, & m=0\\ \frac{1}{3}\left(\frac{m}{|s_1|}+\frac{m}{|s_2|}+\frac{m-t}{m}\right), & m>0\end{cases} \tag{1}$$

其中,$|s_1|$ 和 $|s_2|$ 分别为字符串 s_1 和 s_2 的字符串长度,m 为匹配字符串个数,t 为换位数目。

(2)匹配窗口 MW(matching window)计算公式:

$$MW=\left\lfloor\frac{\max(|s_1|,|s_2|)}{2}\right\rfloor-1 \tag{2}$$

其中,$|s_1|$ 和 $|s_2|$ 分别为 s_1 和 s_2 的字符串长度,当字符串 s_1 中的一个字符在字符串 s_2 中,但位置不同,需要换位操作时,如果这 2 个字符的距离小于等于 MW,则表示这两个字符为匹配字符。统计所有能匹配的字符的所有换位操作数,记为 t_j,则换位的字符数目 t,记为:

$$t=\frac{t_j}{2} \tag{3}$$

(3)Jaro-Winkler 计算公式。

Jaro-Winkler 算法的相似度计算公式为:

$$d_w=d_j+(lp(1-d_j)) \tag{4}$$

式中,p 范围为(0 ~ 0.25),默认值为 0.1;l 是字符串 s_1 和 s_2 的前缀部分匹配长度。

1.3　Levenshtein(编辑距离)算法

编辑距离于 1965 年被提出[12],编辑距离[13]是由原字符串 S 转换成目标字符串 T 最少需要进行的编辑操作次数。编辑操作包含 3 种操作,分别是字符的替换、添加、删除,这 3 种操作次数的总和记为这 2 个字符串的编辑距离。编辑距离越小,相似度越高。

设字符串 $S=s_1s_2\cdots s_m$,$T=t_1t_2\cdots t_n$,建立 S 和 T 的 $(m+1)\times(n+1)$ 阶匹配关系矩阵 **LD**:

$$\mathbf{LD}_{(m+1)\times(n+1)}=\{d_{ij}\}(0\leqslant i<m,0\leqslant j\leqslant n) \tag{5}$$

按公式(6)初始填充矩阵 **LD**:

$$d_{ij}=\begin{cases}i, & j=0\\ j, & i=0\\ \min(d_{i-1,j-1},d_{i-1,j},d_{i,j-1})+a_{i,j}, & i>0,j>0\end{cases} \tag{6}$$

其中,

$$a_{i,j}=\begin{cases}0 & s_i=t_j\\ 1 & s_i\neq t_j\end{cases}(i=1,2,\cdots,m,\ j=1,2,\cdots,n) \tag{7}$$

矩阵 **LD**,右下角元素 $d_{m,n}$ 即为字符串 S 和字符串 T 之间的 Levenshtein 距离,也叫编辑距离,记为 ld。

根据编辑距离 ld,定义字符串 S 和 T 的相似度为[14]:

$$Sim=1-\frac{ld}{\max(n,m)} \tag{8}$$

式中,Sim 为最终的相似度计算结果。越相似的 2 个字符串,Sim 的值将越大。

2　改进 Jaro-Winkler 算法

2.1　供方数据特征

供方数据主要来源于贵州航天计量测试技术研究所的 ERP、TDM 等系统。在对多源数据进行汇集后,发现同一供方的名称数据有大量不一致的情况。

在航天元器件数据中,元器件供方名称和合格供方名称存在不一致问题,典型特征如下:

(1)合格供方名称和元器件供方名称,存在全称和简称情况。例如“中国电子科技集团有限公司第四十九研究所”和“中电四十九所”,“天水天光半导体有限责任公司”和“天水天光”。

(2)合格供方名称和元器件供方名称,存在总公司和子公司信息。例如“易讯科技股份有限公司”和“易讯科技股份有限公司哈尔滨分公司”,“中国航天科工集团有限公司”和“中国航天科工集团第二研究院”。

(3)合格供方名称和元器件供方名称,一个可以区分类别词、一个没有,例如“施耐德电气有限公司”和“施耐德”。

(4)合格供方名称和元器件供方名称相似但不是同一家公司。例如“中国航天科工集团有限公司”和

“中国航天科技集团有限公司”。

(5)合格供方名称和元器件供方名称相似,但类别词不同。例如“深圳海瑞达电子有限公司”和“深圳海瑞达时频设备有限公司”。

(6)合格供方名称和元器件供方名称字面很相似,但顺序不一致。例如“联创电子有限公司”和“创联电子有限公司”。

前三种情形为正例,名称有差异,但是应该匹配成功。后三种情况为反例,名称相似,但不应该匹配成功。

2.2 算法改进

Jaro-Winkler 算法中缺乏对相同字符在原字符串中的间隔问题的考虑[14],因此对相对相似的两个名称不能够有效地拒绝匹配,对于前缀部分相同的两字符,Jaro-Winkler 匹配效果相对比较好。因此 Jaro-Winkler 算法对特征(4)中的名称很相似的不同公司,错误的匹配成功;对特征(6)中公司名称明显错位,但依然匹配成正例。

Levenshtein 算法对 2 个字符串的长度、位序等相对比较敏感,因此对相似的反例能较准确匹配,而对长度差异相对较大的正例易出现匹配错误。因此 Levenshtein 算法对特征(4)到(6)的反例,容易匹配正确;因为全称和简称、总公司和子公司字符串的长度差异大,因此容易对特征(1)到(3)的正例拒绝匹配。

由于 Jaro-Winkler 算法能对前缀部分相同的字符串加分,因此 Jaro-Winkler 相似度算法对正例匹配效果比较好,但对字符位序、对相同字符之间的间隔没有处理,所以对反例的匹配效果并不好。而 Levenshtein 算法考虑了字符串的长度、位序等情况。因此,考虑通过引入调整阈值及系数融合 Jaro-Winkler 和 Levenshtein 算法,以提高整体的匹配正确率,计算公式如下:

$$\text{ana} = \partial d_w + \beta \text{sim} \quad (\partial + \beta = 1) \tag{9}$$

其中,d_w 为 Jaro-Winkle 算法计算的距离,sin 为 Levenshtein 算法计算的相似度。

3 改进算法在航天元器件合格供方中的应用

3.1 供方名称清洗及映射

采购单中的元器件数据来源于不同系统,而不同系统中的元器件供方名称的字段长度、类型存在不一致的问题。在对供方名称进行处理时,先对数据进行清洗能有效提高匹配正确率。数据清洗的对象主要是相似的记录、异常值等。以建立供方名称映射表方式,在不改变原始数据的情况下对供方名称进行清洗。映射表部分数据如表 1 所示。

表 1 供方名称映射表部分数据

序号	清洗前名称	清洗后名称
1	26 所	中国电子科技集团第二十六研究所(中电 26 所)
2	西安开容	西安开容电子技术有限责任公司
3	877 厂	安卫光半导体有限公司(877 厂)
4	元六鸿远	北京元六鸿远电子科技股份有限公司
5	北京元六	北京元六鸿远电子科技股份有限公司
6	北京元六鸿远电子技术有限公司	北京元六鸿远电子科技股份有限公司
7	北京元陆鸿远电子技术有限公司	北京元六鸿远电子科技股份有限公司
8	…	…

3.2 建立停用词字典

在匹配供方数据时,供方名称中的部分后缀文本属于冗余数据,如有限公司、研究院等。这些冗余数据在匹配过程中会影响匹配的准确率。通过建立停用词表,可减少冗余数据对匹配准确率造成的干扰。在匹配供方时,去除供方和合格供方名称中的停用词,可有效提升匹配的正确率。通过对供方数据的分析与梳理,构建了停用词表。停用词表部分数据如表 2 所示。

表 2 停用词表部分数据

序号	停用词
1	有限公司
2	研究院

续表 2

序号	停用词
3	科技有限公司
4	有限责任公司
5	厂
6	研究所
7	…

3.3 合格供方匹配过程

首先对 TDM、ERP 等系统数据进行抽取,在将供方数据汇集到数据仓库过程中,对相似的、异常的供方名称值进行标记,形成供方名称映射表。采购部门用户导入采购单,系统对采购单进行分解,提取供方数

据。然后系统对供方和合格供方数据按照供方名称映射表进行清洗。清洗完成后对供方和合格供方数据进行停用词处理。处理停用词后,利用改进的算法对供方和合格供方进行匹配。最后利用堆排序,对每个供方选择出匹配度最高的 5 个合格供方,供专业人员对其进行判断,如图 1 所示。

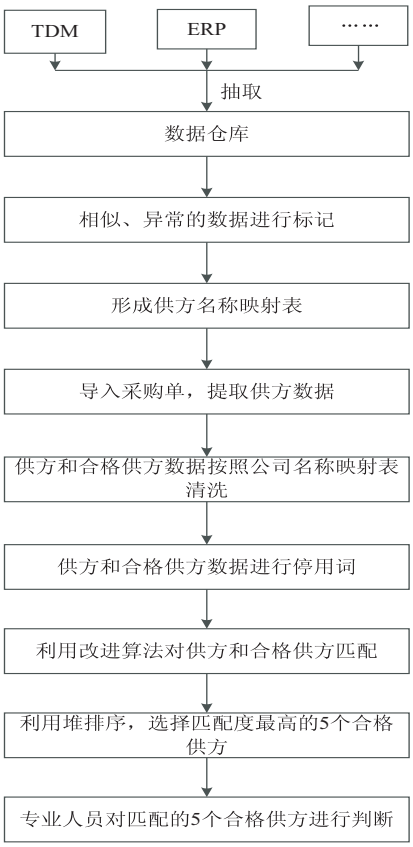

图 1　合格供方匹配过程

3.4　阈值及系数的确定

在进行供方匹配时,需要针对供方名称和合格供方名称按照相似度算法计算相似度,然后将结果与一个选择设定的相似度阈值进行比较,如果大于阈值,则认为该供方为合格供方,否则,判定为不合格供方。文献[1]对 Jaro-Winkle 和 Levenshtein 的最优阈值进行了分析,其中 Jaro-Winkle 的最优阈值为 0.709,其精确率为 85.9%,Levenshtein 的最优阈值为 0.662,其精确率约为 62.2%。因此设定改进算法的阈值为 Jaro-Winkle 的最优阈值和 Levenshtein 的最优阈值的平均值 0.68。

为了确定 ∂ 与 β 的值,随机从供方数据集中抽取三组数据,每组数据有 100 条数据,计算每组数据在不同 ∂ 与 β 值情况下的相似度。调整 ∂ 的步长为 0.1,$0 < \partial < 1$,$\partial + \beta = 1$。当 $\partial = 0.6$,$\beta = 0.4$ 时供方最高相似度多数在 0.6 至 0.8 之间,如图 2 所示。

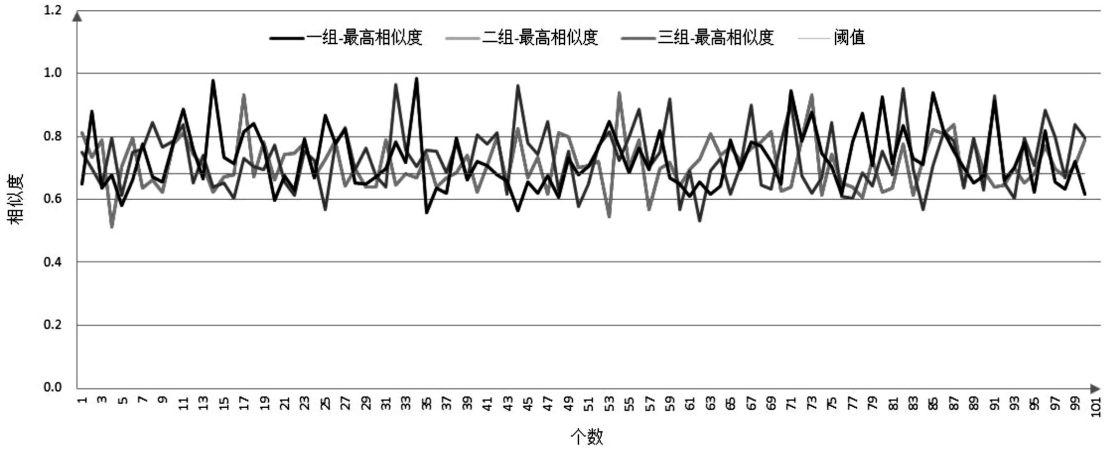

图 2　相似度-阈值曲线

对三组数据精确率进行判断,其中第一组数据中 62 个正例 58 个匹配正确,38 个反例 28 个拒绝匹配;其中第二组数据中 66 个正例 60 个匹配正确,34 个反例 28 个拒绝匹配;其中第三组数据中 58 个正例 51 个匹配正确,42 个反例 34 个拒绝匹配;三组平均精确率为 86.3%。因此改进的算法阈值设定为 0.68,系数设定为 $\partial = 0.6$,$\beta = 0.4$ 时,精确率略高于 Jaro-Winkle 算法,如表 3 所示。

表 3　改进算法的匹配情况

组号	正例匹配	反例拒绝	正确匹配
1	58/62	28/38	86/100
2	60/66	28/34	88/100
3	51/66	34/42	85/100

3.5　合格供方匹配结果分析

在业务系统中测试合格供方匹配功能,结果如图 3 所示。用户导入的采购单中有 500 条随机选取供方数据。导入完毕后系统自动进行匹配工作,匹配结果

正确率相对较高。因此,改进的算法能够很好地完成 合格供方匹配。

图 3 合格供方匹配

4 结束语

建立供方名称清洗表、停用词表,基于 Jaro-Winkler 算法并对其改进,改进后算法能很好地实现供方名称和合格供方名称的匹配,实现批量合格供方匹配的自动化,能极大地提高匹配效率,有助于航天元器件的采购工作,有效保证航天元器件的可靠性。

参考文献:

[1] 刘 震,陈 晶,郑建宾,等. 中文短文本聚合模型研究[J]. 软件学报,2017,28(10):2674-2692.

[2] HARITAOGLU I, HARWOOD D, DAVIS L S. W4:real-time surveillance of people and their activities[J]. IEEE TPAMI,2000,22(8):809-830.

[3] NIKOLOV A,UREN V,MOTTA E,et al. Integration of semantically annotated data by the KnoFuss architecture[C]// Knowledge engineering: practice & patterns. Acitrezza: Springer,2008:265-274.

[4] HERZOG T H, SCHEUREN F, WINKLER W E. Record linkage[J]. Wiley Interdisciplinary Reviews: Computational Statistics,2010,2(5):535-543.

[5] 姜 华,韩安琪,王美佳,等. 基于改进编辑距离的字符串相似度求解算法[J]. 计算机工程,2014,40(1):222-227.

[6] OKUDA T,TANAKA E,KASAI T. A method for the correction of garbled words based on the Levenshtein metric[J]. IEEE Transactions on Computers,2009,C-25(2):172-178.

[7] 吴凌芬,杨小渊,叶添杰,等. 改进 Jaro-Winkler 算法在迎宾机器人语音交互中的应用[J]. 现代计算机:普及版,2015(3):8-13.

[8] 何 锋,谷锁林,陈彦辉. 基于编辑距离相似度的文本校验技术研究与应用[J]. 飞行器测控学报,2015,34(4):389-394.

[9] 赵作鹏,尹志民,王潜平,等. 一种改进的编辑距离算法及其在数据处理中的应用[J]. 计算机应用,2009,29(2):424-426.

[10] TASI C S,HUANG Y M,LIU C H,et al. Applying VSM and LCS to develop an integrated text retrieval mechanism[J]. Expert Systems with Applications,2012,39(4):3974-3982.

[11] HUNT J W,SZYMANSKI T G. A fast algorithm for computing longest common subsequences[J]. Communications of the ACM,1977,20(5):350-353.

[12] YE J. Cosine similarity measures for intuitionistic fuzzy sets and their applications[J]. Mathematical and Computer Modelling,2011,53(1-2):91-97.

[13] KUMARSHRIVASTAVA S, RANA J L, JAIN R C. Text document clustering based on phrase similarity using affinity propagation[J]. International Journal of Computer Applications,2013,61(18):38-44.

[14] LEVENSHTEIN V. Binary codes capable of correcting deletions, insertions and reversals[J]. Soviet Physics Doklady, 1966,10(1):845-848.