

中文文本挖掘中特征抽取和表示

林鸿飞 战学刚 张跃 姚天顺

(东北大学计算机系, 沈阳 110006, ics@ramm.neu.edu.cn)

摘要: 文本挖掘是从非结构化的文本中发现潜在的概念以及概念间的相互关系。文本的特征是概念的表现形式, 特征抽取是文本挖掘的必要基础。鉴于中文文本的特点, 本文提出了基于结合性的中文姓名识别方法、数字特征的转换以及基于模糊语义的表示和检索。它们具有较强的适应性和良好的反映能力, 不依赖于具体的领域知识。

关键字: 文本挖掘 中文姓名识别 特征抽取 模糊语义表示

Features Extraction and Representation for Chinese Text Mining

Lin Hongfei, Zhan Xuegang, Zhang Yue and Yao tianshun

(Department of Computer, Northeastern University, Shenyang, 110006, ics@ramm.neu.edu.cn)

Abstract: This paper briefly describes the background of text mining and the main difficulties in Chinese text mining. It is well known that text features play an important role in text mining, so the paper presents the approach of identifying Chinese name, the approach for transforming digital features and fuzzy semantic representation of the features. They have good adaptability and representative ability, they also are independent of domains.

Keywords: Text Mining, Chinese Name Identification, Text Feature Extraction, Fuzzy Semantic Representation

1、引言

在信息技术高速发展的今天, 尤其是因特网的日益普及, 人们每天都要获取和处理大量的信息。如何帮助用户在日益增多的信息中自动发现新的概念并自动分析它们之间的关系, 使之能够真正地做到信息处理的自动化, 这已成为信息技术领域的热点问题。在这样的需求驱动下, 文本挖掘得到了长足的发展, 并取得了相当的成功。

文本挖掘不同于数据挖掘, 数据挖掘面对的是结构化数据, 采用的方法大多是非常明确的定量方法。其过程包括数据取样、特征提取、模型选择、问题归纳和知识的发现。而文本挖掘由于它处理的是非结构化的文本, 因此, 决定它采用的方法与数据挖掘不同。它经常使用的方法来自与自然语言理解和文本处理领域, 如文本摘要、文本分类、文本检索等技术。

对于中文文本的文本挖掘其难度较大, 体现为汉语分词问题, 建立完整的汉语概念体

系的困难和汉语语法、语义和语用分析的困难。我们在 CETRAN^[1]的词典、概念词典和汉语分析器的基础上,给出中文文本挖掘的特征抽取机制。首先建立一般特征项的抽取模型,然后重点讨论专有特征项的抽取问题,包括基于结合性的中文姓名识别、数字特征的转化及其基于模糊语义的表示。

2、一般特征项的抽取

一般特征项的选择根据阈值,将权重大于阈值的特征项列出。特征项权重函数定义如下:

$$f_w(t_i) = \frac{f_u(t_i) \log_2(1 + f_v(t_i))^l}{\sqrt{\sum (f_u(t_j) \log_2(1 + f_v(t_j)))^2}}$$

其中: $f_w(t_i)$ 表示特征项 t_i 的权重函数; $f_u(t_i)$ 表示特征项在文本内的频数; $f_v(t_i)$ 表示特征项 t_i 的段落频率,即包含 t_i 的段落数/文本总段落数; l 表示特征项 t_i 的长度。

这个公式实质上是著名的权重公式 $tf * idf$ 的扩展。权重函数的设计基于如下的事实:

特征项的段落频率越高,表明该特征项反映文本主题的能力越强,因此应赋予较大的权重。另外,短词具有较高的频率,更多的含义,是面向功能的;而长词的频率较低,是面向内容的。加大长词的权重,增强词汇的区分度。也可以减轻单个汉字成词的不稳定性。

标题、副标题以及关键字表中出现的词汇和短语是当然的特征项。

3、中文姓名识别

中文姓名一般来说随意性较大,而且又不像英语中利用首字母大写等信息区分姓名,所以中文姓名的识别比较困难,除少数著名的人物收录在词典中,通过分词可以获得外,其他姓名的识别必须采用专门的方法来获得。

中文姓名的组成的统计结果^[3]提示我们,建立姓氏用字表(First Name List)和名字用字表(Last Name List)和常用姓名表(Common Name List),检测可能的姓名用字。

为了方便表述和处理,为文本中出现的字和词赋予相应的属性函数值 $ATTRIBUTE(x)$, x 为字符串, $x=c_1, c_2, \dots, c_n$, c_1, \dots, c_n 为单字。标点符号的属性值为 Sign。

定义 1: 姓氏用字表中的字称为姓氏用字。“王”,“林”等等。属性值为 Surname。

定义 2: 姓名中不使用的字或极少使用的单字称为名字禁用字。如“死”、“奸”、“吧”、“呢”等等。属性值为 Stop。

定义 3: 出现在常用姓名表中的词汇称为姓名用词。如“王雪松”。属性值为 Name。

定义 4: 姓名中出现的词汇称为姓名用词。如“高尚”、“方圆”。属性值为 Pass。

定义 5: 对于符合下列条件的双字词组称为普通用词。属性值为 Common。否则称为非普通用词,属性值为 None。

条件 1：出现在分词字典中的非姓名用词。如“翻阅”、“浏览”。

条件 2：首字为动词，尾字为虚词。如“笑了”、“跑着”。

条件 3：首字为虚字，尾字为动词。如“亦是”、“也算”。

条件 4：首字为动词，尾字为方位词。如“翻过”、“跳上”。

条件 5：首字为数词，尾字为量词。如“三台”、“八张”。

条件 6：不是多字词的前两个字。如“才华横溢”、“张牙舞爪”。

定义 6：有一些词或符号经常出现在姓名的左右，包括表示称谓的名词和指界动词。如“先生”、“省长”、“经理”、“指出”、“说”、“授予”等等。出现在姓名左边的称为前称谓词，属性值为 Left；出现在姓名右边的称为后称谓词，属性值为 Right。

定义 7：若 x 在文本 T 中确认为姓氏用字，则 First_Name(T,x) 为真，否则为假。

定义 8：若 y 在文本 T 中确认为名字用字，则 Last_Name(T,y) 为真，否则为假。

定义 9：姓名解析表达式 $Name_Describe(x,m,n) = e[0]f[m]...f[2]f[1]x b[1]b[2]...b[n]e[1]$ ，其中 $ATTRIBUTE(x) = Surname$ ， $ATTRIBUTE(e[0]) = Stop\ OR\ Sign$ ， $ATTRIBUTE(e[1]) = Stop\ OR\ Sign$ ， $m \geq 0$ ， $n \geq 0$ 。实际上是将含有姓氏用字的句子分段后的结果。

姓氏识别算法：

```
PROCEDURE Identification_Surname(Text T, Name_Describe(x,m,n) )
```

```
BEGIN
```

```
IF ATTRIBUTE(f[2],f[1]) = Left THEN First_Name(T,x) = TRUE
```

```
IF ATTRIBUTE(b[1],b[2])= Right OR ATTRIBUTE(b[2],b[3]) = Right OR
```

```
ATTRIBUTE(b[3],b[4]) = Right THEN First_Name(T,x) = True
```

```
IF ATTRIBUTE(f[1],x) = None AND ATTRIBUTE(x,b[1]) = None
```

```
THEN First_Name(T,x) = True ELSE
```

```
IF ATTRIBUTE(f[1],x) = Common THEN
```

```
IF AND ATTRIBUTE(f[2],f[1]) = None THEN First_Name(T,x) = False ELSE
```

```
IF ATTRIBUTE(f[3],f[2]) = None THEN First_Name(T,x) = True ELSE
```

```
MessageBox "Can't identify the Surname ! "
```

```
END.
```

名字识别算法：

```
PROCEDURE Identification_Name(Text T, Name_Describe(x,m,n) )
```

```
BEGIN
```

```
IF n = 1 THEN Last_Name(T,x,b[1]) = True ELSE
```

```
IF n = 2 THEN Last_Name(T,x,b[1],b[2]) = True ELSE
```

```
IF ATTRIBUTE(b[1],b[2])=Right THEN Last_Name(T,b[1],b[2]) = True ELSE
```

```
IF ATTRIBUTE(b[3],b[4])=Right THEN Last_Name(T,b[1],b[2]) = True ELSE
```

```
IF ATTRIBUTE(b[2],b[3])=Right THEN Last_Name(T,b[1]) = True ELSE
```

```
IF ATTRIBUTE(b[2],b[3])=Pass THEN Last_Name(T,b[1],b[2]) = True ELSE
```

```
IF ATTRIBUTE(x,b[1],b[2])=Name THEN Last_Name(T,b[1],b[2]) = True ELSE
```

```
IF ATTRIBUTE(b[3],b[4])=Common THEN Last_Name(T,b[1],b[2]) = True ELSE
```

```
IF ATTRIBUTE(b[3])=Stop THEN Last_Name(T,b[1],b[2]) = True ELSE
```

```

IF ATTRIBUTE(x,b[1])=Name THEN Last_Name(T,b[1]) = True ELSE
IF ATTRIBUTE(b[2],b[3])=Common THEN Last_Name(T,b[1]) = True ELSE
IF ATTRIBUTE(b[2])=Stop THEN Last_Name(T,b[1]) = True ELSE
Last_Name(T,b[1],b[2]) = True

```

END.

例如：他 f[3]打 f[2]开 f[1]张(x)扬(b[1])先(b[2])生(b[3])的(b[4])礼 b[5]品 b[6]盒 b[7]。识别姓名所采用的规则如下：

```

ATTRIBUTE(f[1],x) = Common ---> ATTRIBUTE(f[2],f[1]) = Common
---> ATTRIBUTE(f[3],f[2]) = None ---> First_Name(T,x) = True (识别姓氏为“张”)。
ATTRIBUTE(b[2],b[3]) = Right ---> Last_Name(T,b[1]) = True(识别名字为“扬”)。
最终识别出来的姓名为张扬。

```

对于系统未能确定的姓氏，采用上下文信息加以进一步的区分。我们设立了候选名字表，共有两个域，一个为候选姓名，另一个为频率信息。每当有候选姓名出现，如果它出现在常用姓名表中，则直接认定为姓名。如果它不出现在常用姓名表中，此时若具有良好的边界特征，也可直接认定为姓名，并送往常用姓名表；否则，加入候选姓名表。若已存在，则频率计数加1。当频率计数超过阈值时，可认定为姓名，并送往常用姓名表。

对《人民日报》94年800余篇语料处理结果，表明查全率和正确率分别为98.62%和81.94%，而且误判率大于漏判率。

4、数字特征的转换和表示

每个民族都有自己的数字表示法，但数的概念是独立于任何具体的语言。对于数词而言，按照语义分类则分成系数词和位数词。系数词是数字的名字，位数词是数字所处位置的指称。位数词又可以分成层位数词和子位数词，由于大多数语言采用分层读数法，每层配有一个位数词来标记该层的位值，这类位数词称为层位数词。英语数字每三位一层，层位数词有 thousand, million 等。汉语数字是每四位一层，层位数词为“万”、“亿”等，为了叙述简便，本文限定层位数词最高为“万亿”。子位数词用于标记每一层内各系数词的位值，汉语有“千”，“百”，“十”。假定数字只有一层时，即小于一万的数字其层位数词为 Ω （其字符值为“”），小于十的数字的子位数词也为 Ω 。则数词结构定义如下：

```

数词 ::= {子数词块 + 层位数词}^n,
子数词块 ::= {子系数词 + 子位数词},
层位数词 ::= {万亿、亿、万、 $\Omega$ },
子系数词 ::= {零、一、二、三、四、五、六、七、八、九},
子位数词 ::= {千、百、十、 $\Omega$ }

```

下面讨论汉语数字转化的规则。首先将层位数词、子位数词和子系数词转化为相应的十进制数字，特别地将 Ω 转化为 10^0 。根据层位数词分段，得到如下的数字特征解析式：

$$Digit(x) = (x_{n3}y_3 + x_{n2}y_2 + x_{n1}y_1 + x_{n0}y_0)w_n + \dots + (x_{03}y_3 + x_{02}y_2 + x_{01}y_1 + x_{00}y_0)w_0$$

其中 n 称为层数, $y_3 = \text{千}$, $y_2 = \text{百}$, $y_1 = \text{十}$, $y_0 = \Omega$, $x_{ij} (i = n, n-1, \dots, 0)$ 为子系数词, $w_i (i = n, n-1, \dots, 0)$ 为层位数词。则转化的数字为:

$$\text{Digit}(x) = \sum_{i=0}^n \left(\sum_{j=0}^3 x_{ij} 10^j \right) 10^i .$$

如果有小数, 通过“点”或其它标志分段, 对于小数部分单独处理后, 再与整数部分相加, 小数部分数字解析式 (不包括小数点), $\text{Digit}(x) = z_1 z_2 \dots z_m, z_i (i = 1, \dots, m)$, 是子

系数词。则转化的数字为: $\text{Digit}(x) = \sum_{i=1}^m z_i 10^{-i} .$

除了上述的单纯数字转化外, 对于与数字紧密结合的修饰词也必须加以分析, 以获取范围信息。如超过五十吨、少于一百万张、一万多元等等, 诸如“增加”、“少于”、“多”、“余”等词汇都是重要的范围提示信息。

识别出的数字还应考虑相应的种类特征, 即分成日期、时间、数字和货币等加以处理。这些种类的其主要特点是相应的词组一般由数词和各种特征词构成, 如年、月、日、元、角、美元, 马克等; 数词表现方式比较复杂: 有汉字, 有阿拉伯数字, 数字间可能存在其它字。如二十八岁、50 马克、五月六日、一元八角、1234.01、四分之三、百分之四十五等等。

对于日期特征, 存在三种日期形式: 一是绝对日期, 如一九九九年六月八日; 另一种是相对日期, 即相对于某日期原点的日期, 若给定日期原点值, 则可以转化成绝对日期, 如三年前; 最后一种是伪日期, 无法转化为绝对日期, 如几天前, 若干年后。为了统一日期表示, 便于检索处理和统计需求, 将各类日期转化为如下格式: (日期原点为 d_origin)

一九九九年六月八日 \rightarrow 1999-06-08 五月八日 \rightarrow $d_origin + 0000-05-08$
 二十四日 \rightarrow $d_origin + 0000-00-24$ 三年前 \rightarrow $d_origin - 0003-00-00$
 几天前 \rightarrow [$d_origin - 0000-00-09, d_origin - 0000-00-01$]

对于时间特征, 往往与日期同时出现, 这里在日期的基础上, 单独考虑时间因素。它同样存在三种时间形式: 一是绝对时间, 如三点五十分三十一秒; 另一种是相对时间, 即相对于某时间原点的的时间, 若给定原点值, 则可以转化成绝对时间, 如二小时前; 最后一种是伪时间, 无法转化为绝对时间, 如几分钟, 几小时后。相仿日期表示, 将各类时间转化为如下格式: (时间原点为 t_origin)

三点五十分三十一秒 \rightarrow 03:50:31 十八时五分 \rightarrow 18:05:00
 十点整 \rightarrow 10:00:00 三小时前 \rightarrow $t_origin - 03:00:00$
 六分钟后 \rightarrow $t_origin + 00:06:00$ 几小时前 \rightarrow [$t_origin - 12:00:00, t_origin - 01:00:00$]

对于数字统一转化为####,###,###,###.##格式, 加上相应的量词或货币名称构成数字词组, 表示单纯数量特征。

一万两千三百四十美元 \rightarrow 1,234.00 美元 三十六吨 \rightarrow 36.00 吨
 六百八十多元 \rightarrow [680 - 690] 元 高于 1600 转/分 \rightarrow [1600 --∞]转/分

值得指出的是将伪日期, 伪时间和不定数量词的模糊语义数量化, 转化为相应的区段, 即模糊区间数, 在文本挖掘中具有重大意义。采用的基本思想是设 U 是线性有序的论域,

这里 U 可以是所有合法日期的集合、所有时间的集合和数字的集合, $[a,b]/p$ 表示一个模糊区间数, $a,b \in U$, $0 < p \leq 1$, $[a,b]$ 称为区间, p 称为可能度。区间数表示某个模糊数落在区间 $[a,b]$ 的可能度。在日期特征中,指某个具体日期落在日期区段的可能性;对于数字,表示数字落在该数字区间的可行性。对于出现的伪日期、伪时间和不定数量词,将其区间加大到足够大范围,使其能够包括通常意义下的所有可能取值,此时 p 的值为 1,因此对于常见的伪日期、伪时间和其它不定数量的词汇说明相应的区间。对于确切的日期和数字认为它们也是的模糊区间数, $[a,b]/1$, $a=b$ 。在检索操作时,两个模糊数之间的语义距离为:

$$S(x,y) = \sqrt{w_1|a_1 - a_2|^r + w_2|b_1 - b_2|^r + |p_1 - p_2|^r},$$

$$x = [a_1, b_1] / p_1, y = [a_2, b_2] / p_2$$

这里 $p_1 = p_2 = 1, r = 2, w_1 + w_2 + w_3 = 1, w_1 = w_2 = 0.5, w_3 = 0$ 。所以有:

$$S(x,y) = \sqrt{0.5 * |a_1 - a_2|^2 + 0.5 * |b_1 - b_2|^2}$$

如果 $S(x,y)$ 小于指定的阈值,则可以认为符合检索条件。

5、结束语

如何在浩如瀚海的文本流中到挖掘到潜在的概念以及概念间的相互关系,是十分关键的问题。特征抽取是文本挖掘的必要基础,由于中文文本的特殊性,在特征抽取时,存在一定的难度,体现为建立完整的汉语概念体系的困难和中文姓名、机构名和数字特征识别及其表示等方面的问题。针对这些问题,本文提出了基于结合性的中文姓名识别算法、数字特征的转换和模糊语义表示。

在设计文本挖掘模型过程中,面对海量文本,采用统计方法有着较强的适应性和良好的反映能力,不依赖于具体领域知识。但是,随着需求的深入,引入基于自然语言理解的语法分析、语义分析和语用分析势在必行,以便挖掘更深的知识。下一步的重点是研究更有效地把自然语言理解技术应用到文本挖掘中,提高挖掘的精度和效率。

参考文献

- [1] 姚天顺等,自然语言理解,清华大学出版社,1995
- [2] 麻志毅,林鸿飞,姚天顺,基于情境的文本中时间信息分析,东北大学学报, Vol. 3, No. 6, 1999
- [3] 吴立德等,大规模中文文本处理,复旦大学出版社,1997
- [4] 刘开瑛等,中文文本中抽取特征信息的区域与技术,中文信息学报, Vol. 12 No. 2, 1998
- [5] Udo,Klemens and Schnattinger, Deep Knowledge Discovery from Natural Language Texts, In Proceedings of the 3rd Conference on Knowledge Discovery and Data Mining,1997,175-178.
- [6] G.Salton,J.Allen and C.Buckley. Automatic Structuring and Retrieval of Large Text Files. Communications of the ACM, Vol.37 No.2, February 1993, 97-108.