

中文信息学报

第 16 卷 第 2 期 JOURNAL OF CHINESE INFORMATION PROCESSING Vol. 16 No. 2

汉语情感意义的机器标注研究初探

应英¹ 周锋² 周昌乐²

(1. 浙江水利水电专科学校 杭州 310016 2. 浙江大学人工智能研究所 杭州 310028)

摘要: 本文将情感计算引入到汉语的机器理解中,在已有的汉语机器理解研究的基础上,采用多重松弛迭代计算方法,对汉语情感意义的标注问题进行了研究,通过语境信息的利用,构建了一个实验性系统并取得了较准确的词语情感标注,为后续的句子情感意义的理解提供了基础,拓宽了汉语机器理解的研究范围。

关键词: 汉语机器理解;情感标注;多重松弛迭代算法;学习及纠错机制

中图分类号: TP391.2

A Research on Emotion Tagging of Chinese Understanding by Designing an Experiment System

YING Ying¹ ZHOU Feng² ZHOU Chang-Le²

(1. Zhejiang Water Conservancy And Hydropower College 310016

2. Computer Department Xixi Campus Zhejiang University 310028)

Abstract: This paper introduces emotions and their computation to machine understanding of Chinese. Based on the existing research of the machine understanding of Chinese, the problem of the emotional meaning tagging of Chinese was studied by using the multiple relaxation alternate algorithm. We designed an experimental system and obtained accurate Emotion tagging matching by using the context information. It is the first step to the machine understanding of the emotional meaning of Chinese sentences, thus the foundation for the future research is laid.

Keywords: Machine understanding of Chinese; Emotion tagging; Multiple relaxation alternate algorithm; Learn and correct system

一、引言

自然语言作为人们交流的重要工具,不可否认地包含着交流者之间的感情投入。而现代汉语作为在世界上使用最为广泛的自然语言之一,其情感意义的计算机理解的研究在各方面都拥有很高的价值。本课题基于一个实验系统的研究,将汉语情感计算引入汉语机器理解,拓宽了汉语机器理解的研究范围。

1.1 现代汉语词语的理性意义和感性意义

我们知道一篇文章是由许多条句子构成的,而一条句子又是由许多个词语构成的,因此对整篇文章情感意义的理解归根结底是对构成该篇文章的词语的情感意义的理解。

人们通常理解某个词语的意义,就是知悉这个词语指什么样的对象,而且一般抽象从本质

收稿日期:2001-8-23

本文得到国家自然科学基金项目(项目号:69983006)和浙江省教育厅科研基金项目(项目号:20010335)支持。

作者应英,女,1960年生,讲师,在职研究生,研究方向为计算语言学、检测和信息处理、计算机程序设计。周锋,男,1979年生,本科生。周昌乐,男,1959年生,教授,博士生导师,主要从事人工智能领域的研究工作,研究方向为计算语言学、理论脑科学和认知逻辑学。

特点或区别性特点来把握该对象。这种理解、知悉和把握,是理性认识的过程;而被认知的意义本身,也形成在抽象、概括对象特点的认识基础上。因而这种意义是理性的。除此之外,词语意义中还有大量感性的部分存在。尽管它是附丽在词语理性意义上,并不成为一个词语的意义主体,但是正是由于这些感性意义的存在,才使得我们的语言如此地绚丽多彩、千变万化。在某种意义上,理解语言的感性意义比理解它的理性意义更为重要。

我们通常把词语的感性意义称为表达色彩。表达色彩的分类方法多种多样,根据刘叔新在《汉语描述词汇学》一书中提到的,可将其分为感情色彩、态度色彩、评价色彩、现象色彩、语体色彩、风格色彩、格调色彩、语气色彩等。而我们根据实际需要可以粗略地将其划分为感情色彩和非感情的语体及形象色彩,其中感情色彩正是我们此次研究的重点。

众所周知,一个词语的情感意义并不总是唯一的,要了解某个词的情感意义,必须要把它放到一个特定的语境中去综合理解。举个简单的例子,“戏子”一词在不同的语境中就存在不同的情感意义:

要是在旧社会,你不就是个戏子吗? 句(1)
戏子的一生终究无戏。 句(2)

显然句(1)中的“戏子”包含的是轻蔑和嘲讽,而句(2)中则更多地能体会出一种悲哀。作为一个实验系统,我们从众多情感分类中挑选了最具代表的11种情感意义:喜悦(简称 XI,下同)、尊敬(ZHUN)、亲切(QIN)、恼怒(NAO)、悲哀(BEI)、沮丧(JVS)、惧怕(JVP)、嘲讽(FEN)、憎恶(ZEN)、惊奇(QI)、疑问(YI)和无明显情感意义(WU)作为这个实验系统的情感分布。

1.2 自然语言处理与情感标注

自然语言处理就是利用电子计算机对人类特有的书面形式和口头形式的自然语言的信息进行各种类型处理和加工的技术。自然语言处理是双向的,既包括了计算机从自然语言中获取信息,也包括了计算机通过自然语言输出信息。实质就是自然语言的机器理解问题。要让计算机理解自然语言,不仅要进行词法分析、句法分析,还要进行语义分析。目前的处理手段主要是“先分后合”:先让计算机自动分词,把词语作为研究的基本对象,再对词语进行自动标注,最后通过某种计算机模型将其合一,把多个词语的含义合成为一条句子乃至整篇文章的含义。目前自动标注仅限于两个方面:词性的自动标注和词义的自动标注。根据冯志伟的MMT(Multiple branched and Multiple labeled Tree Analysis 多叉多标记树形分析法)语法理论,在描述汉语句子时,除了给出其组成成份的词类或词组类型特征之外,还必需给出语义关系特征,才不致产生歧义,即多叉多标记树形分析法。感情色彩作为自然语言的一大内容,要实现它的计算机理解自然可以使用同一手法,我们设想把情感计算理论引入汉语的机器理解中,即把感情色彩也作为词语的标注之一,配合适当的合一算法,参与合一运算就可得语句的情感,其目的是拓宽汉语机器理解研究范围。本实验系统的构造就是对汉语情感意义的机器标注进行研究。

二、总体设计思想和工作原理

2.1 总体设计思想介绍

首先需要说明的是这个系统并不是一个独立的系统,确切的说,它只是整个汉语情感的机器理解系统的一个模块,其主要功能是在分词的基础上,确定每个词的情感意义,并为今后的语句乃至语篇的情感合一打下基础。

当计算机将一组已经完成分词计算的词语输入本系统,系统会逐一从基本词库中找到相应的静态信息,然后再以这些静态信息为基础,配合动态数据库,通过多重松弛迭代计算公式,计算产生动态信息,完成对词语的情感标注。其中,静态信息是指基本词库及词语情感关系数

据库,基本词库中收录词语的各种可能的情感意义及其可能性,即各词的情感分布;词语情感关系数据库中则收录同一语境中各词的情感之间的相互关系。这两者的内容是根据事实情况设定的。静态并不是指其内容是固定不变的,而是指它对每一次运算都能提供一个相对稳定的数据。动态则是指该系统的情感计算是动态的,不同的语境中,对同一个词的计算结果是不同的。本实验系统的基本处理过程如下:

首先是预处理,即根据用户的输入完成从外部数据库中提取有用信息,调入系统事先设置好的数据结构中,为以后的运算作好准备。

其次是系统的主体——运算部分。即通过多重松弛迭代计算公式:

$$q_{ij}^{(k)} = \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^m R_{ii}(j, j) P_{ij}^{(k)} \right) \quad \text{公式(1)}$$

$$P_{ij}^{(k+1)} = \frac{P_{ij}^{(k)} (1 + q_{ij}^{(k)})}{\sum_{j=1}^m P_{ij}^{(k)} (1 + q_{ij}^{(k)})} \quad \text{公式(2)}$$

来进行整体关联性的类属标注,即确定词语在特定语境中的情感取向的。其中 $P_{ij}^{(k)}$ 表示标注分析中第 i 个词语选择第 j 种情感属性在第 k 次迭代中的机率可能性, $R_{ii}(j, j)$ 为调整因子,也可称为相关系数,反映的是词语标注之间相互约束和作用的影响,另外, $R_{ii}(j, j)$ 也作为约束条件,保证了松弛迭代的有效收敛。这样可以产生整体一致性的最优情感标注分布结果。

最后是学习及纠错机制,即一方面根据运算结果,对外部数据库进行更新,提高以后运算的准确性。另一方面如果用户觉得结果与事实不符,还可以进行人为的修改,系统会根据用户的修改自动地更新数据库,使以后的类似运算得出正确的结果。

2.2 本系统的工作原理图

本系统的工作原理见图 1。

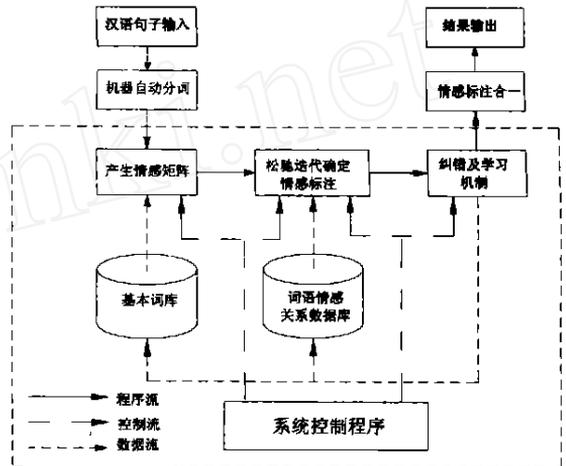


图 1 本系统的工作原理图

三、实验系统的详细设计

理论的正确性需要实践的证明,因此本节将构造一个相对简单的实验系统来证明上述理论的正确性。

3.1 数据结构

系统的外部数据结果是由两个 Paradox7 数据库(test. DB 与 rel. DB)和一个顺序存储文件 reldata 构成的。

test. DB 中存放单个词语的情感分布。它有 13 个数据段。其中除了 WORD 段是 16 个字节的字符串型外,其它 12 个数据段都是 0~1 的小数,且对于特定的某个词语,其 12 个数据段数据的和为 1。例如,“戏子”一词在 test. DB 中的存放情况如下:

| WORD | XI | ZHUN | QIN | NAO | BEI | JVS | JVP | FEN | ZENG | QI | YI | WU |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 戏子 | 0.00 | 0.00 | 0.00 | 0.00 | 0.42 | 0.00 | 0.00 | 0.50 | 0.00 | 0.00 | 0.00 | 0.08 |

我们从中不难发现“戏子”一词在三个数据段上有取值,它们分别是 BEI(悲哀)、FEN(嘲讽)、WU(无),这表示“戏子”一词在不同的语境中或表达悲哀或显示嘲讽或无明显的情感意义,而每种情感的可能性即为该数据段的数值,也就是最初的 P 值。

rel.DB 与 reldata 合起来构成了词语情感关系数据库。其中 rel.DB 是一个索引,我们通过它在顺序存储文件 reldata 中读写相关数据,它有 ADDRESS(Int)、WORD1(Alpha)、WORD2(Alpha)共 3 个数据段,ADDRESS 中存放 WORD1 与 WORD2 的情感相关系数表在 reldata 中的偏移量。顺序存储文件 reldata 则是存储每两个词的相关系数 R。例如,“戏子”与“无戏”两词的相关信息存储情况如下:

< rel.DB >

| ADDRESS | WORD1 | WORD2 |
|---------|-------|-------|
| 1 | 戏子 | 无戏 |

< reldata >

| | 喜悦 | 尊敬 | 亲切 | 恼怒 | 悲哀 | 沮丧 | 惧怕 | 嘲讽 | 憎恶 | 惊奇 | 疑问 | 无 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|
| 喜悦 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 尊敬 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 亲切 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 恼怒 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 悲哀 | 0.00 | 0.00 | 0.00 | 0.00 | 0.87 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | 0.00 | 0.20 |
| 沮丧 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 惧怕 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 嘲讽 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | 0.45 | 0.00 | 0.00 | 0.00 | 0.00 |
| 憎恶 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 惊奇 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 疑问 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 无 | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 |

词一为“戏子”,词二为“无戏”。从表中可以看出“戏子”一词的各种情感标注与“无戏”一词的各情感标注之间的相互关系,即 R 值。R 值的取值范围为 [0,1],“1”表示两种标注相伴出现,“0”表示两种标注相互排斥,不会一起出现,“(0,1)”表示两种标注之间的关系程度。

除了外部数据结构,我们还须为系统运算提供所需的内部数据结构。在此不详述。

3.2 详细设计

1. 预处理

由于本系统是建立在分词完毕的基础上的,所以在这里要求用户输入时须用空格将词与词分隔开,为了处理方便,我们把标点符号也作为一种特殊的词来处理。例如,句(2)“戏子的一生终究无戏。”就应如下输入:

戏子 的 一 生 终 究 无 戏 。 (这里 表示空格)

面对这样的输入先把头尾的空格去掉(如果有的话),然后第一次遍历整个句子,统计出有多少的词语,分配出相应大小的内存区域,接着第二次遍历,把各个词语的具体内容放入对应的内存区域中。存放单个词语的数据结构如下:

```
typedef enum { XI,ZHUN,QIN,NAO,BEI,JVS,JVP,FEN,ZENG,QI,YI,WU} word-aff;
struct ciyu
{ int no;
  char na[16];
  word-aff aff;
```

}

no 为词语在句子中的位置;na 为词语的内容;aff 则是一个枚举变量,其内容决定词语的情感含义。其中前两项的内容可以在这里生成,而第三项则须通过之后的运算产生,在这里统一赋初值“WU”。例如,句(2)“戏子的一生终究无戏。”在此阶段的存储情况如下(P 为存储各词语的数组名):

| | | | | | | | |
|-----|------|------|------|------|------|------|-----|
| | 1 | 2 | 3 | 4 | 5 | 6 | |
| ... | 戏子 | 的 | 一生 | 终究 | 无戏 | 。 | ... |
| | WU | WU | WU | WU | WU | WU | |
| | P[0] | P[1] | P[2] | P[3] | P[4] | P[5] | |

接着建立相应的情感矩阵和相关系数数组。通过 P 数组中存放的词语内容在基本词库中查找词语对应的情感数据,从而构成一个 $n \times 12$ 的情感矩阵(n 为词数)。同样可以在词语情感关系数据库中找到相应的数据,构造出一个 $(n \times 12) \times (n \times 12)$ 的相关系数四维数组。还是用句(2)“戏子的一生终究无戏。”一句来举例:

<情感矩阵>

| | XI | ZHUN | QIN | NAO | BEI | JVS | JVP | FEN | ZEN | QI | YI | WU |
|----|------|------|------|------|------|------|------|------|------|------|------|------|
| 戏子 | 0.00 | 0.00 | 0.00 | 0.00 | 0.32 | 0.00 | 0.00 | 0.54 | 0.00 | 0.00 | 0.00 | 0.14 |
| 的 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| 一生 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| 终究 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| 无戏 | 0.00 | 0.00 | 0.00 | 0.00 | 0.61 | 0.00 | 0.00 | 0.21 | 0.00 | 0.00 | 0.00 | 0.18 |
| 。 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

<相关系数四维数组>

| | 戏子 | 的 | 一生 | 终究 | 无戏 | 。 |
|----|-----|-----|-----|-----|-----|-----|
| 戏子 | ... | ... | ... | ... | ... | ... |
| 的 | ... | ... | ... | ... | ... | ... |
| 一生 | ... | ... | ... | ... | ... | ... |
| 终究 | ... | ... | ... | ... | ... | ... |
| 无戏 | ... | ... | ... | ... | ... | ... |
| 。 | ... | ... | ... | ... | ... | ... |

| | 喜悦 | 尊敬 | 亲切 | 恼怒 | 悲哀 | 沮丧 | 惧怕 | 嘲讽 | 憎恶 | 惊奇 | 疑问 | 无 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|
| 喜悦 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 尊敬 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 亲切 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 恼怒 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 悲哀 | 0.00 | 0.00 | 0.00 | 0.00 | 0.87 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | 0.00 | 0.20 |
| 沮丧 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 惧怕 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 嘲讽 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 | 0.45 | 0.00 | 0.00 | 0.00 | 0.00 |
| 憎恶 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 惊奇 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 疑问 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 无 | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 |

值得注意的是,这里所列出的数据只是本次运算时的数据,由于引入了学习及纠错机制,这些数据是在不断动态变化、完善的。

2. 多重松弛迭代运算

松弛迭代是一种广泛应用于模式识别和计算机视觉等各项技术中的计算方法,主要通过使用局部的不一致性和多次重复迭代来获得一个完整的全局分析。这里,每个词语的情感分布通过迭代渐渐收敛到最可能值。前面已经介绍了松弛迭代非线性概率模型的迭代算子,见公式(1)、(2)。现在,将运算所需的所有数据(情感矩阵 P 和相关系数四维数组 R) 代入算式一步步地进行迭代,直至情感矩阵中每一行的数据中都有一个唯一的值变为 1,即这条句子的每个词语都有了唯一确定的情感意义。另外,为了提高收敛速度,我们选择了一个阈值 K (如 $K = 0.98$),当 $P_{ij} > K$ 时,有

$$(\forall j \neq j_0) (P_{ij} < 1 - K)$$

这时就可认为迭代结束。

一旦迭代结束,就遍历整个情感矩阵,将每个词语行中数值为 1 的对应情感填入最初存放各词语信息的数组。如例句此时该数组的存储情况为:

| | | | | | | | |
|-----|------|------|------|------|------|------|-----|
| | 1 | 2 | 3 | 4 | 5 | 6 | |
| ... | 戏子 | 的 | 一生 | 终究 | 无戏 | 。 | ... |
| | BEI | WU | WU | WU | BEI | WU | |
| | P[0] | P[1] | P[2] | P[3] | P[4] | P[5] | |

至此,本系统的基本功能已经完成。

3. 学习及纠错机制对系统的完善

一个词语的某种情感每出现一次,就意味着这个词语的这个情感的出现概率增加一点,

同样,两个词语情感在一起出现一次,也就意味着这两个词语情感的联系加深一点。正是出于这样的考虑,在每次成功地对一句话中各词语进行情感标注之后,都应该对外部的基本词库和词语情感关系数据库进行相应的更新,这样如果下次碰到类似的运算,就会更快地得出结果。

运算后 <基本词库>

| | XI | ZHUN | QIN | NAO | BEI | JVS | JVP | FEN | ZEN | QI | YI | WU |
|----|------|------|------|------|------|------|------|------|------|------|------|------|
| 戏子 | 0.00 | 0.00 | 0.00 | 0.00 | 0.36 | 0.00 | 0.00 | 0.52 | 0.00 | 0.00 | 0.00 | 0.12 |
| 的 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| 一生 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| 终究 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| 无戏 | 0.00 | 0.00 | 0.00 | 0.00 | 0.65 | 0.00 | 0.00 | 0.19 | 0.00 | 0.00 | 0.00 | 0.16 |
| 。 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

< rel data >

| | 喜悦 | 尊敬 | 亲切 | 恼怒 | 悲哀 | 沮丧 | 惧怕 | 嘲讽 | 憎恶 | 惊奇 | 疑问 | 无 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|
| 喜悦 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 尊敬 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 亲切 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 恼怒 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 悲哀 | 0.00 | 0.00 | 0.00 | 0.00 | 0.88 | 0.00 | 0.00 | 0.09 | 0.00 | 0.00 | 0.00 | 0.18 |
| 沮丧 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 惧怕 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 嘲讽 | 0.00 | 0.00 | 0.00 | 0.00 | 0.09 | 0.00 | 0.00 | 0.45 | 0.00 | 0.00 | 0.00 | 0.00 |
| 憎恶 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 惊奇 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 疑问 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 无 | 0.00 | 0.00 | 0.00 | 0.00 | 0.18 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.10 |

显然,如果我们通过更新词频来完成这一任务是非常合适的,但是,统计词频的工作量是巨大的,在这样一个实验系统中,我们采用了一个相对简单的做法,即一个词语的某种情感每出现一次,就将基本词库中该词语的该情感数值加上一个数值,而该词语的其它情感数值则相应地减去一个数值;类似地,当两个词语情感在一起出现,就将 *reldata* 文件中对应的关系表的对应数值 m 作 $0.9m + 0.1$ 运算,而其它同行同列的数值 n 作 $0.9n$ 运算。以下是句(2“戏子的一生终究无戏。”运算后的数据存储情况(*reldata* 只列举了“戏子”与“无戏”的情感关系表):

3.3 本实验系统的一些不足之处

(1)通过观察,不难发现情感关系数据库中的绝大多数数据都是0,这些冗余无疑是对存储空间巨大浪费,在实际应用中应该运用某种压缩算法在空间与时间之间找到最佳支点。

(2)在本实验系统中,关键的相关系数 R 值是直接从数据库中读取的,而较为科学的做法则是通过某个结合多个参数的算式计算得出 R 值。

(3)本系统所考虑的语境信息仅仅是以句子为单位的,而现实的语境往往是以段落甚至整篇文章为单位的,如果我们要实现这样大篇幅的语境,一方面会大大增加数据统计的复杂性,另一方面也会使系统运算的速度大受影响。而且即使是在一个句子中,我们也仅仅只考虑了每两个词语之间的一对一的相互联系,这就无法体现现实情况中多词之间的联系;另外词语在句中所处的相对位置对该词语所蕴含的情感意义也有很大的影响,这在本系统中也没有得到体现。

四、结束语

本系统首次将情感计算引入到汉语的计算机理解当中,拓宽了汉语机器理解的研究范围;将语境信息数字化,使得计算机有了分析不同语境中单个词语情感意义的基础;借用松弛迭代技术,取得在某一特定语境中词语的最佳情感标注。本课题的研究初有成效,这对于汉语机器理解新内容和新思路的发展,都是有着建设性的科学研究意义和重要的学术价值的。

现代汉语是当今世界最复杂的自然语言之一,尤其是汉语中所固有的歧义现象,其语境的复杂程度可想而知。把情感计算引入到汉语的机器理解中,理论上可以帮助计算机消除在汉语理解上的歧义问题,但另一方面由于大量的词语存在一词多义的现象,在进行情感标注时也不可避免地存在情感歧义的现象,本系统尚在本课题研究的初步阶段,这一问题还有待深入研究。

参 考 文 献

- [1] 詹人凤. 现代汉语语义学. 北京:商务出版社,1997. 10
- [2] 刘叔新. 汉语描述词汇学. 北京:商务印书馆,1990. 11
- [3] 冯志伟. 自然语言的计算机处理. 上海:上海外语教育出版社,1996. 10
- [4] 秦莉娟,周昌乐. 一种引入多重松弛算法的汉语词类标注方法. 南京大学学报,2000,36(10)
- [5] 赵巍. 一种基于语境信息利用的汉语自动分词研究. 杭州大学计算机系硕士学位论文. 1997,12
- [6] M. G. Dger. *Emotions and their computations: Three Computer models, cognition and Emotion*,1987 1(3):323 - 347
- [7] N. H. Frijda and J. Swagerman, *can computers feel? Theory and design of an emotional system.* *Cognition and Emotion*,1987 1(3):235 - 257