高技术通讯 2022 年 第32 卷 第6 期:597-606

doi:10.3772/j.issn.1002-0470.2022.06.005

面向中文电子病历的属性挖掘①

费超群②* ** *** 张书涵* ** *** 李阳阳**** *****

(*智能信息处理重点实验室 北京 100190)

(**中国科学院计算技术研究所 北京 100190)

(***中国科学院大学 北京 100049)

(****管理、决策与信息系统重点实验室 北京 100190)

(*****中国科学院数学与系统科学研究院 北京 100190)

摘 要 电子病历(EMR)的属性挖掘任务旨在从一组同一科室下的病历文本中抽取该科室医学检查项目。传统的频繁项或序列挖掘技术并不能直接用于该任务。本文提出一种新的不需要人工干预的属性挖掘框架,并借助无标注技术来处理这一难题,即将属性挖掘问题形式化为半结构化的频繁子序列挖掘任务,并提出一种有效的算法从电子病历中挖掘候选的词模式。在中文电子病历上进行的各项综合实验,证明了本文提出的方法可以有效处理属性挖掘任务。

关键词 属性挖掘;电子病历(EMR);频繁子序列挖掘;词模式;频繁词模式

0 引 言

电子病历(electronic medical record, EMR)已被现代医院广泛使用,但是许多电子病历并未被规范保存。大多数病历还是以无结构化形式被存储在电子文档中。这样的病历常存在以下问题:(1)病历文本中由不同医生对同一症状的表达存在差异;(2)病历文本中常出现不规范表达和噪音等现象。这种病历不便于被再次利用。而将无结构的电子病历转化为结构化的"〈检查项目,检查结果〉对"能有效解决这些问题。这个"对"的第 1 项被称为电子病历的属性,第 2 项称为该属性对应的属性值。

属性挖掘是属性值对抽取任务中的重要部分,本文的工作就是要从某科室的电子病历中抽取该科室电子病历的属性。

属性与现有工作中的指示性短语不同[1-4],区别包括:(1)它们的目标和任务不同。指示性短语用来表示一组特定文档的特征,而属性则用于表示这些文档中所讨论对象的共同特征。(2)它们在子句中的位置不同。指示性短语出现在子句中任何位置,而属性常出现在子句开头或结尾。(3)属性一定是指示性短语,但是指示性短语不一定是属性。(4)属性在单个文本中不常见,但对于一组文本来说是很常见的。与其不同的是,指示性短语在单个文本和多个文本中都很常见。(5)它们的应用场景不同。指示性短语抽取通常用于信息检索、文本聚类等任务。属性挖掘更适用于知识库构建、领域文本摘要生成等任务。

属性挖掘问题是基于这样的假设,即所有属性,特别是重要属性,会经常出现在一组电子病历中。直观上这是合理的,因为可以作为属性的词或短语在语料库中出现的频率是比较高的。为此,本文提出一种频繁属性挖掘框架。从文本中提取出频繁词模式并对它们进行分组,使用候选属性得分函数,从

① 国家自然科学基金(61232015,61472412,61621003),国家重点研发计划(2016YFB1000902),中国博士后科学基金(2020TQ0341),北京科技项目和清华-腾讯-AMSS 联合项目资助。

② 男,1992 年生,博士生;研究方向:知识图谱,知识工程,自然语言处理;联系人,E-mail: feichaoqun@ict.ac.cn。

(收稿日期:2021-04-12)

候选频繁词模式集合中选择得分最高的词模式作为属性。该方法避免了使用监督学习和无监督学习来提取信息的一些缺点。基于中文电子病历的实验结果表明,该方法能够有效地处理属性挖掘问题。

1 相关工作

现有工作中已有一些从类似病历的文本中抽取属性或属性值对的任务。文献[5]从产品描述中抽取“〈产品属性,属性值〉对”,把该抽取任务形式化为多分类问题,并提出半监督学习算法 co-EM 来解决此问题。文献[6]从关于某主题的短广告语料库中提取“〈键,值〉对”,将该任务形式化为序列标注问题,并基于人工标注训练集利用监督学习的方法来处理这个问题。文献[7]将产品属性提取问题形式化为命名实体识别任务,并分别借助有监督和半监督方法从 eBay 商品标题中提取产品属性。上述工作中提出的方法都在文献[8]归纳的信息抽取方法框架内。而现有的有关属性或属性值对抽取的工作[9-10],也基本按照文献[8]所归纳的几种信息抽取的思路进行,即将属性抽取任务看成是序列标注问题或分类问题。这些工作与本文的工作相似,但他们的研究对象和解决问题的方法与本文不同。

另一种相关工作是类属性提取,其目标集中在构建类的“骨架”上。文献[11]利用引导式的方法,基于种子属性集同时从 Web 文档和查询日志中提取类属性。文献[12]引入轻监督的方法,用于从与特定主题相关联的网络搜索查询中提取实体属性。但上述工作在目标和任务上与本文不相同。

频繁模式挖掘的思想在其他领域也有很广泛的应用。文献[13]应用序列模式挖掘来解决广义中心字符串挖掘问题。文献[14]应用序列模式挖掘来处理事件日志中的挖掘。从电子病历中抽取信息的工作最近也有一些研究[15-17],但是这些工作均不涉及属性挖掘,目前还没有专注于基于电子病历的属性挖掘工作。

2 问题定义

序列模式挖掘问题由 Agrawal 和 Srikant[18] 在 1995 年提出。他们认为,“序列模式”是纯粹的句法概念,即如果一个符号序列出现得足够频繁,则可称其为序列模式。序列越长,模式就越有价值。本文提出的“属性”概念不仅具有句法特征还有特有的语义规则约束。

频繁的属性与频繁的序列模式有一些相似之处,但两者之间也存在本质区别,特别是在电子病历上。

针对电子病历的文本结构,本文列出基本定义如下。(1)每个文本都是由句号分隔的句子序列。(2)每个句子都是用逗号或分号分隔的子句序列。(3)每个子句都是由词、数字和冒号(最多 2 个)组成的序列。(4)每个词在语法上都是汉字字符序列,在语义上是基本意义单元。(5)如果词 u 的字符序列是词 v 的字符序列的子序列,则 u 称为 v 的子词。(6)如果 u 是 v 的(字符或词)子序列,则 v 是 u 的(字符或词)超序列。(7)如果 u 是 v 的(字符或词)子序列但不等于 v,则 u 是 v 的(字符或词)真子序列。(8)属性是词序列,用来表示一组文本共同包含的对象的特征。(9)属性值是用于评估属性的数字或文本描述。(10)病历文本是半结构化文本,其中的一些词用来为文本中描述的对象的属性赋值。(11)如果文本的某些部分必须遵循某些预先指定的句法规则,则将其称为半结构化文本。(12)电子病历的大多数子句中,每个子句都由一个属性及其相关的属性值组成。

定义 1 给定一组电子病历和用户指定的最小支持度阈值,频繁属性挖掘旨在根据一些启发式规则以频繁出现的词序列作为候选属性,并从中选出最合理的属性。

在上述定义中,启发式规则是对电子病历语义的非正式描述,它既包含医学领域知识,又包含医生书写病历的惯例。

词数据库 S 是元组 $<sid,s>$ 的集合,其中 sid 是包含词序列 s 的记录的序列号。S 中不同元组具有不同的序列号 sid。如果词序列 α 是 s 的词子序列,则称元组 $<sid,s>$ 包含词序列 α。S 中词序列 α 的支持度是 S 中包含 α 的元组数,表示为 $support_S(\alpha)$。

定义 2 设 α 是病历数据库 R 中的一个词序列。给定正整数 ξ 作为支持度阈值,如果 $support_R(\alpha) \geqslant \xi$,则称 α 是 R 中的词模式。

3 语料研究

实验用的中文电子病历由某口腔医院提供,它们分别来自修复科、正畸科和颌面外科。

3.1 属性分类

根据"〈属性,属性值〉对"中相关属性值的类型,可以将属性分为 3 类,如表 1 所示。

表 1 属性类型及示例

类型	示例
二元型	缺失(是/否),触痛(有/无)
描述型	口腔卫生状况无明显异常
度量型	松动:2 度,骨板厚度:约 8 mm

二元型属性是指该属性在文本中的取值只有存在或不存在 2 种情况;描述型属性是指其属性值是一段描述性的文本;度量型属性的属性值由数字和度量单位组成。

在某些情况下,属性值会插入属性的词序列中。例如,在子句"牙龈无红肿"中,"牙龈红肿"是属性,而"无"是其属性值。对于具有这种结构的子句,可以通过删除插入的属性值并连接剩余的单词以获取属性词。这种插入型的属性值称为插入词。本文所提出的算法也能够适用这种情境。

3.2 属性和属性值的位置关系

属性和属性值在文本中的位置关系可以分为 3 类。

(1) 属性和值是连续的,如"牙槽嵴吸收轻度,咬合无明显异常"。

(2) 属性和值被冒号分开,如"触痛:无,弹响:无"。

(3) 两层属性,如"中线:上中线:左偏 1 mm;下中线:右偏 2 mm"。其中,"中线"是父属性,"上中线"和"下中线"是子属性,父属性和子属性通常由冒号隔开。

对于第 2 类和第 3 类有明显句法规律的情形,仅使用一些句法规则就能抽取到属性,不属于本文所考虑的情形。本文的算法主要面向第 1 类文本,即没有句法规律的文本。

4 属性挖掘算法 Attrimining

本节介绍面向中文电子病历属性挖掘的算法 Attrimining。它包含词模式挖掘、词模式分组和词模式组属性选择 3 个阶段。

4.1 词模式挖掘

当前,用于频繁子序列挖掘的主要方法有 2 种[19],一种是基于 Apriori 的方法,例如 Apriori-All[18]、广义序列模式(GSP)[20];另一种是模式增长类的方法,例如 FreeSpan[21]、PrefixSpan[22]。

由于属性和频繁子序列之间的差异,这些算法都无法直接应用于属性挖掘任务。频繁子序列是自由符号序列,而属性是半结构化的词序列。此外,前者是纯粹的句法概念,而后者在语义上要求领域相关。有证据表明,每个属性都必须由领域专家最终确认。

为应对电子病历中挖掘属性的挑战,本文提出一种新算法 Attrimining(见算法 1),该算法能够挖掘满足最小支持阈值 ξ 的属性。Attrimining 有 3 个核心概念:前缀、投影和后缀,见定义 3 ~5。

定义 3 给定词序列 W,词序列 W' 被称为 W 的前缀,当且仅当存在词序列 U(空或非空)满足 $W = W' \circ U$,其中 $\circ$ 是词序列连接符。

定义 4 给定词序列 W,词序列 W' 被称为 W 的后缀,当且仅当存在词序列 V(空或非空)满足 $W = V \circ W'$,其中 $\circ$ 是词序列连接符。

定义 5 给定词序列 W 和 P,W 的词子序列 W' 被称为 W 关于 P 的投影,当且仅当 P 是 W' 的前缀;W' 是 W 的后缀。

例如,给定词序列 w_1 =〈"口腔","卫生","状况","良好"〉。则〈"口腔"〉和〈"口腔","卫生"〉都是 w_1 的前缀。〈"卫生","状况","良好"〉和〈"状况","良好"〉都是 w_1 的后缀,同时〈"卫生","状况","良好"〉是 w_1 关于前缀〈"口腔"〉的投影。

定义 6 设 w 是子句数据库 C 中的词模式。则

$C|_w$ 表示元组 $<proj, cid, rid>$ 的集合,其中,rid 表示某病历文本的序号(唯一且可代指该病历),cid 是该病历文本中某子句的序号(可代指该子句),$proj$ 是 w 关于病历 rid 中子句 cid 的投影。

Attrimining 的 3 个步骤:挖掘词模式(算法 1 中的第 2 ~ 11 行)、对词模式进行分组(算法 1 中的第 12 行)以及从每个组中选择词模式作为属性(算法 1 中的第 13 行)。在本节中只介绍第 1 阶段,后 2 个阶段将分别在第 4.2 节和第 4.3 节中详细说明。

给定病历记录数据库 R,在 R 中,每条文本记录都已通过分词器被分割为单词列表。首先通过子例程 *ObtainClauses* 获取子句数据库 C。子句分隔符,即标点符号逗号“,”、分号“;”、冒号“:”、顿号“、”等是划分子句的依据,可以用于提取第 3.2 节中提到的第 2 类或第 3 类属性,以及帮助构建用于抽取第 1 类属性的子句数据库 C。

对于 C 中的每个子句 c 中的每个词 w_i,如果 w_i 满足一定条件,则 w_i 是 c 的前缀,表示为 $prefix_i$,而剩下的词将作为 $prefix_i$ 相应的候选后缀。然后,调用 *GetPostfix* 以获取 $prefix_i$ 的真实后缀 $postfix_i$(指去掉插入词和标点后的候选后缀)。同时,构建 $prefix_i$ 的投影数据库 $S|_{prefix_i}$。最终,可以获取所有长度为 1 的前缀。

算法 1 Attrimining

输入:病例数据库 R;最小支持度 ξ

输出:病历属性集

初始化:$wp\_all \leftarrow \{\}$ /*词模式集合*/

$prefixList\_1 \leftarrow \{\}$ /*长度为 1 的前缀集合*/

$AETrieList \leftarrow \{\}$ /*AETrie 集合*/

$AttributeSet \leftarrow \{\}$ /*属性集合*/

1 调用 *ObtainClauses*(R)获取子句数据库 C

2 **for** C 中的子句 c **do**

3 **for** c 中的词 w_i **do**

4 如果 w_i 满足:w_i 不以数字开头和结尾;w_i 不包含任何标点;w_i 不是插入词,则:

5 词序列为 $ws_i = w_i \cdots w_{C.length-1}$ 为 w_i 的前缀,记为 $prefix_i$;调用 *GetPostfix*(ws_i)获取 $prefix_i$ 的后缀 $postfix_i$;添加 $prefix_i$ 到 $prefixList\_1$ 中;添加元组 $<postfix_i, cid, rid>$ 到 $prefix_i$ 的投影数据 $S|_{prefix_i}$ 中

6 **end for**

7 **end for**

8 **for** $prefixList\_1$ 中的前缀 $prefix_j$ **do**

9 如果 $support_R(prefix_j) \geqslant \xi$,则:

10 添加 $prefix_j$ 到 wp_all;基于 $prefix_j$ 创建 *AETrie* 节点 *AENode*;添加 *AENode* 到 *AETrieList*;调用 $WPFinding(prefix_j, 2, S|_{prefix_j}, AENode, wp\_all)$

11 **end for**

12 合并 *AETrieList* 并将 wp_all 根据 *AETrieList* 中的 *AETrie* 进行分组

13 从每组中选择 *AScore* 得分最高的词模式作为属性,并将其添加到 *AttributeSet*

14 **return** *AttributeSet*

下一步,检查 $prefixList\_1$ 中的每个前缀 $prefix_j$。如果 $support_R(prefix_j) \geqslant \xi$,即未重复的 rid 数目大于 ξ,则表示 $prefix_j$ 是词模式,将其添加到 wp_all 中。然后调用 *WPFinding*(见子例程 1)基于前缀 $prefix_j$ 及其投影数据库 $S|_{prefix_j}$ 来抽取长度大于 1 的词模式。同时,基于 $prefix_j$ 创建一个 *AETrie*(将在 4.2 节中进行说明)节点 *AENode* 并将其添加到 *AETrieList*,以记录基于前缀递归生成词模式的过程。

接下来,举例演示 Attrimining 如何抽取词模式并构建 *AETrie*。假设有表 2 中所示的病历记录数据库 R,令 $\xi = 2$。

通过调用 *ObtainClauses*(R)从 R 中获取子句数据库,其中的部分展示在表 3 中。以表 3 中的几个子句为例,从中挖掘词模式的过程呈现在图 1 中。从表 3 中,可以挖掘到的词模式包括〈“口腔”〉,〈“口腔”,“卫生”〉,〈“口腔”,“卫生”,“状况”〉,〈“口腔”,“卫生”,“状况”,“良好”〉,〈“卫生”〉,〈“卫生”,“状况”〉,〈“卫生”,“状况”,“良好”〉,〈“状况”〉,〈“状况”,“良好”〉。

子例程 1 $WPFinding(wp, l, S|_{wp}, node_{wp}, wp\_all)$

参数:wp:词模式;l:本次挖掘的词模式的长度;$S|_{wp}$:wp 的投影库;$node_{wp}$:wp 对应在 *AETrie* 上的节点

初始化:$prefixList\_l \leftarrow \{\}$ /*长度为 l 的前缀集合*/

1 **for** $S|_{wp}$ 中的元组 $<postfix_i, cid, rid>$ **do**

2 连接 wp 和 $proj[0]$ 以组成新词序列 seq_new;将 $proj$ 中除 $proj[0]$ 以外的词作为 seq_new 的后缀 $post\_new$;添加元组 $<postfix\_new, cid, rid>$ 到 seq_new 的投影数据 $S|_{seq\_new}$ 中;添加 seq_new 到 $prefixList\_l$ 中

3 **end for**
4 **for** *prefixList _ l* 中的前缀 $prefix_j$ **do**
5 如果 $support_R(prefix_j) \geqslant \xi$, 则:
6 添加 $prefix_j$ 到 *wp _ all*; 基于 $prefix_j$ 创建 *AETrie* 节点 *AENode*, 并将 *AENode* 作为 $node_{wp}$ 的子节点;调用 $WPFinding(prefix_j, l+1, S|_{prefix_j}, AENode, wp\_all)$
7 **end for**
8 **return**

表 2 病历数据库 *R*

病历	*rid*
松动 3 度,开合度正常,口腔卫生状况一般	1
松动 2 度,开合度 4 cm,口腔卫生状况良好	2
松动 3 度,开合度 3 cm,口腔卫生状况良好	3
松动 2 度,开合度 3 cm,口腔卫生状况较差	4

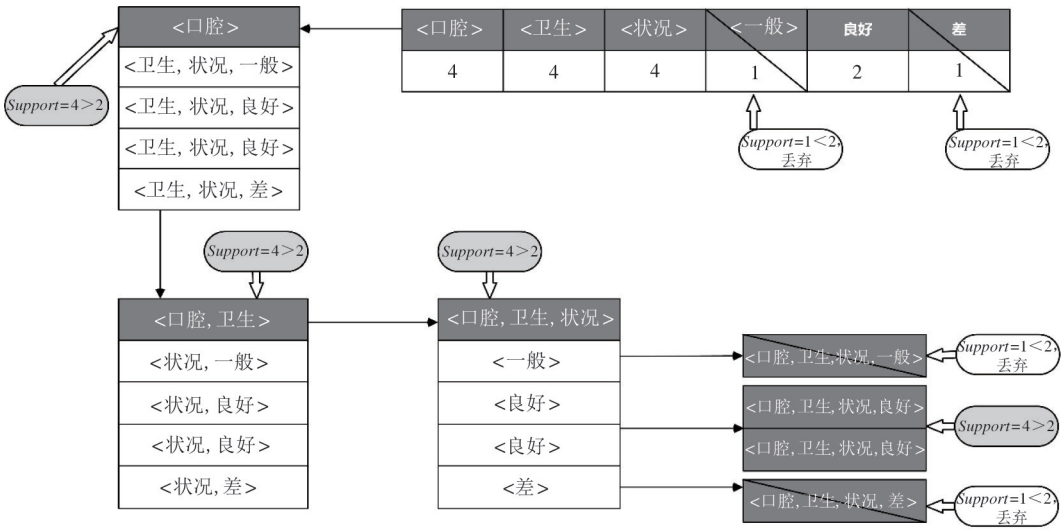


图 1 词模式生成过程图解

表 3 病历子句数据库

病历	*cid*	*rid*
口腔卫生状况一般	3	1
口腔卫生状况良好	6	2
口腔卫生状况良好	9	3
口腔卫生状况较差	12	4

4.2 基于 *AETrie* 的词模式分组

Attrimining 的第 2 个阶段是词模式分组,即根据 *AETire* 将 *wp _ all* 中的词模式进行分组。*AETire* 的构造是伴随着词模式的生成一起进行的。分组的目的是从组中选择最合理的候选属性,并排除不能作为属性的词模式。

AETire 具有以下特性:

(1) 所有节点都是长度为 1 且满足 ξ 的词模式。

(2) 从节点的任意祖先到其自身的路径也是词模式。

(3) 一条路径可表示的词模式的最大数量为 $\frac{n(n+1)}{2}$, n 为路径中节点的数量。

(4) 根节点到叶节点的路径上的所有子路径可表示的词模式都被划分到同一组。

假设有一个 *AETrie*, 见图 2。根节点是〈“口腔”〉,叶节点是〈“情况”〉,〈“一般”〉,〈“良好”〉。

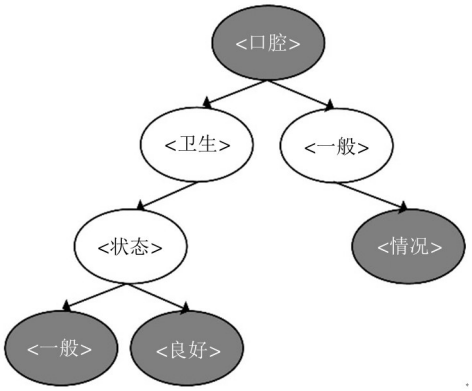


图 2 *AETrie* 示例

根据 *AETrie* 的性质,在路径〈“口腔”〉→〈“一般”〉→〈“情况”〉中,可以得到以下词模式:〈“口腔”〉,〈“一般”〉,〈“情况”〉,〈“口腔”,“一般”〉,〈“一般”,“情况”〉,〈“口腔”,“一般”,“情况”〉。这些词模式将被划分在相同的组。

当一个 *AETrie* 是其他 *AETrie* 的子树时,将它们进行合并。

4.3 基于 *AScore* 的词模式组属性选择

Attrimining 的第 3 个阶段是从词模式组中选择最合理的词模式作为属性。

根据对病历数据库的以下观察:(1)如果词模式的支持度越大,它越有可能是属性;(2)如果词模式的长度越长,它越有可能是属性;(3)如果词模式越接近子句的最左边,它越有可能是属性。因此,影响词模式 wp 作为属性的 3 个因素分别为 wp 的支持度 $support(wp)$ 、wp 的长度 $len(wp)$ 、以及 wp 的平均最左距离(average most left distance,AMLD) $AMLD(wp)$。综合这 3 个因素,本文设计了得分函数 $AScore(wp)$,用于计算词模式 wp 作为属性的得分,如式(1)所示。

$$AScore(wp) = \frac{support(wp) \times len(wp)}{AMLD(wp)} \quad (1)$$

其中,*AMLD* 是 wp 在其关联子句中的平均最左距离(most left distance,MLD)。$MLD(wp)$ 是指词模式最左边的词与原始子句中最左边的词之间的距离。设最小 *MLD* 值为 1,$len(wp)$ 表示 wp 中词的数量。

假设有子句 c =〈“口腔”,“卫生”,“状况”,“良好”〉,从 c 中抽取的词模式中有:〈“口腔”,“卫生”,“状况”〉,〈“卫生”,“状况””〉,〈“状况”,“良好”〉,它们对应的 *MLD* 分别为 1、2、3。

从表 3 所示的子句中,可以得到一个词模式组,其中的部分见表 4。以第 2 个词模式 wp_2 =〈“口腔”,“卫生”〉为例,来计算 wp_2 的 *AScore* 得分。wp_2 的支持度是 4,长度是 2,在它原始子句中的所有 *MLD* 都是1。根据式(1),$AScore(wp_2)$ =4×2/((1+1+1+1)/4) =8。在这组中,〈“口腔”,“卫生”,“状况”〉的 *AScore* 得分最高。因此,“口腔卫生状况”就被选择作为该组的属性。

表 4 词模式和 *AScore* 得分

词模式	*AScore*
〈口腔〉	4×1/((1+1+1+1)/4) =4
〈口腔,卫生〉	4×2/((1+1+1+1)/4) =8
〈口腔,卫生,状况〉	4×3/((1+1+1+1)/4) =12
〈口腔,卫生,状况,良好〉	2×4/((1+1+1+1)/4) =8
〈卫生,状况〉	4×2/((2+2+2+2)/4) =4
〈卫生,状况,良好〉	4×3/((2+2+2+2)/4) =6

另外,由于中文的书写惯例,一些词模式不应作为候选属性。为此,Attrimining 中也添加了一些启发式规则,如下:

(1) 长度为 1 且只包含一个字符的词模式不能作为属性。

(2) 长度为 1 且 *AMLD* 值较高的词模式不能作为属性。

(3) 如果词模式在 *AETrie* 中是根节点,且有多个子节点,则不能作为属性。

4.4 时间和空间复杂度分析

Attrimining 算法的主要时间消耗在前缀生成阶段。假设病历数据库中子句的数量为 N_C,每个子句的平均单词数是 $\overline{N_{CW}}$,则算法的时间复杂度为 $O\left(N_C \times \frac{\overline{N_{CW}}(\overline{N_{CW}}+1)}{2}\right)$。在实际场景下,病历数据库中,每个子句所含有的词数一般都比较少,也即 $\overline{N_{CW}} << N_C$。

Attrimining 在运行时的内存消耗主要在前缀投影库的构建和 *AEtrie* 的构建过程中。前缀和其投影数据库的构建的空间复杂度为 $O(N_C \times \overline{N_{CW}})$。假设构建的前缀数量为 N_{prefix},显然 $N_{prefix} < N_C \times \overline{N_{CW}}$,即构建 *AETrie* 的空间复杂度小于 $O(N_C \times \overline{N_{CW}})$,据此推断算法整体的空间复杂度为 $O(N_C \times \overline{N_{CW}})$。

5 实验和评估

本节通过几组实验来验证 Attrimining 算法的有效性,以及影响其性能的关键因素。实验平台为 Windows 10,内存为 20 GB;处理器型号为 Intel Core (TM) i7-4990,算法实现语言和版本为 Python 2.7。

5.1 病历库预处理

原始电子病历存在以下问题。(1)表达不统一。如数字“1”存在多个形式的表达:“1”、“I”和“一”等。(2)测量单位不统一。如单位“度”存在有“度”、“°”等表示形式。(3)中文和英文标点符号混合。如括号中文形式“(”和英文形式“(”。(4)“\t”、“\n”或其他特殊符号出现在不合适的位置。这些问题会影响分词和抽取性能,因此必须对语料库进行规范化:包括统一单位、统一表达、统一中英文标点等。

5.2 实验结果与评估

本文使用 Jieba(http://pypi.org/project/jieba/)分词器在中文口腔医学词典的帮助下对电子病历进行分词;再使用 Attrimining 从病历库中提取属性。实验对象为分别从修复科、正畸科和颌面外科中随机抽取的各 200 个病历。在口腔医生的帮助下,对实验病历进行标注,得到黄金数据集,也即每个科室的“黄金”属性。首先展示从 3 个科室抽取的 *AScore* 得分排行前 10 的属性,如表 5 所示。

表 5 3 个科室 *AScore* 得分前 10 的属性

颌面外科	修复科	正畸科
松动	口腔卫生状况	口腔卫生状况
牙龈红肿	缺失牙	上牙弓弯曲
叩诊	松动	下牙弓弯曲
冠周溢脓	叩诊	触痛
牙冠完整	牙垢	X 线片检查结果
X 线片	牙龈退缩	自发性疼痛
牙冠埋伏	根尖周异常	全景 X 线片
临床邻牙检查	近远中距离	X 线头颅侧位片
开口度	近远中间隙	牙列
埋伏牙	骨高度	折断

本文使用精确度(P)、召回率(R)和 $F1$ 值作为基本的评估标准。并考虑以下因素:(1)抽取的候选属性与标准属性之间存在部分匹配。例如“形态对称”是抽取出来的候选属性,但其标准属性是“髁突形态对称”。此种情况下,该候选属性被判定为是部分正确的。(2)存在无效属性。当黄金标准集中的某些属性在实验数据集中出现的次数小于阈值 ξ 时,Attrimining 显然无法适用,这类属性被视为无效属性。为评估因素 1 中“部分匹配正确”的属性,本文设计一些“部分(Part)”评估指标,如表 6 中的“完全和部分准确率”所示。该指标表示部分和完全正确的候选属性数量与标准属性数量的比值。为评估因素 2 中的无效属性,本文设计“高频(Frequent-most)”指标,即只与高频的黄金属性(在病历中出现次数大于等于阈值)进行比较,来评估 Attrimining 在高频场景中的表现。如表 6 中的“高频完全准确率”所示。该指标表示完全正确的属性数量占高频标准属性数量的比例。最终,本文提出 12 个指标作为本次实验的评估标准,见表 6。

表 6 评估标准

完全准确率(FP)	完全召回率(FR)	完全 $F1$ 值($FF1$)
完全和部分准确率(FPP)	完全和部分召回率(FPR)	完全和部分 $F1$ 值($FPF1$)
高频完全准确率(FFP)	高频完全召回率(FFR)	高频完全 $F1$ 值($FFF1$)
高频完全和部分准确率($FFPP$)	高频完全和部分召回率($FFPR$)	高频完全和部分 $F1$ 值($FFPF1$)

表 7 呈现 3 个科室病历的实验评估结果。在所有实验中设置支持度 $\xi=2$。结果表明,本文算法在不同科室的实验数据集上都有很好的效果,尤其是在颌面外科数据集上。

表 7 3 个科室评估结果

	颌面外科			修复科			正畸科		
	P	R	$F1$	P	R	$F1$	P	R	$F1$
%F	80.70	61.33	69.70	53.51	53.98	53.74	55.21	56.38	55.79
%FP	96.67	77.33	85.93	81.48	77.88	79.64	90.22	89.36	89.79
%FF	80.70	83.64	82.14	53.51	84.72	65.59	55.21	77.94	64.63
%FFP	96.67	96.36	96.51	81.48	94.44	87.49	90.22	98.53	94.19

%F:完全;%FP:完全和部分;%FF:高频完全;%FFP:高频完全和正确。

5.3 与基准方法的比较

由于未在相关文献中找到应用于和本文类似的中文数据集的属性挖掘任务的方法,为此,本文设计一种基于规则的属性抽取方法,即使用正则表达式来抽取属性,作为基准比较方法。

在基于规则的实验中首先统计词频,然后结合高频词和语法结构设计抽取规则。其中使用的几条规则如下:“ < 位置 > * ? [无 | 不] < 症状 > + ”,“ < 项目 > :”,“ < 项目 > [约] * ? mm”。

修复科和颌面外科的病历被用于此次的比较实验,对比结果见表 8。

表 8 与基准的比较评估结果

	颌面外科				修复科			
	FP	*FPP*	*FR*	*FPR*	*FP*	*FPP*	*FR*	*FPR*
基准/%	75.82	85.93	46.36	58.28	54.31	73.63	47.78	76.11
本文/%	80.70	96.67	61.33	77.33	53.51	81.48	53.98	77.88

在完全准确率(*FP*)、部分和完全准确率(*FPP*)、完全召回率(*FR*)、部分和完全召回率(*FPR*)等指标上与基准方法的比较结果显示,Attrimining 在多数指标上都优于基准方法。

5.4 算法运行时间耗费评估

本文使用正畸科的数据集来评估 Attrimining 算法在不同大小数据集下的时间耗费,结果见图3。当实验数据较少时(< 140),时间耗费呈多项式的变化;而当实验数据较大时(> 140),时间耗费呈接近线性的变化。从算法时间复杂度公式可以看出,当 $N_{\overline{CW}}$ 和 N_C 差距不大时,时间耗费应该呈现多项式的变化;但是当 $N_{\overline{CW}}$ 和 N_C 差距很大时,也即 $N_{\overline{CW}} << N_C$ 时,时间耗费的决定性影响因素是 N_C,时间耗费曲线会呈现准线性的变化。

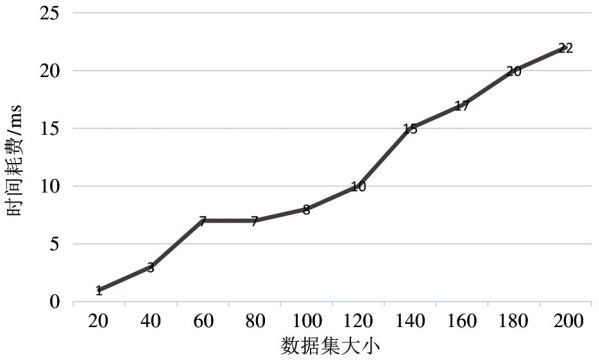


图 3 不同数据集大小下的时间耗费

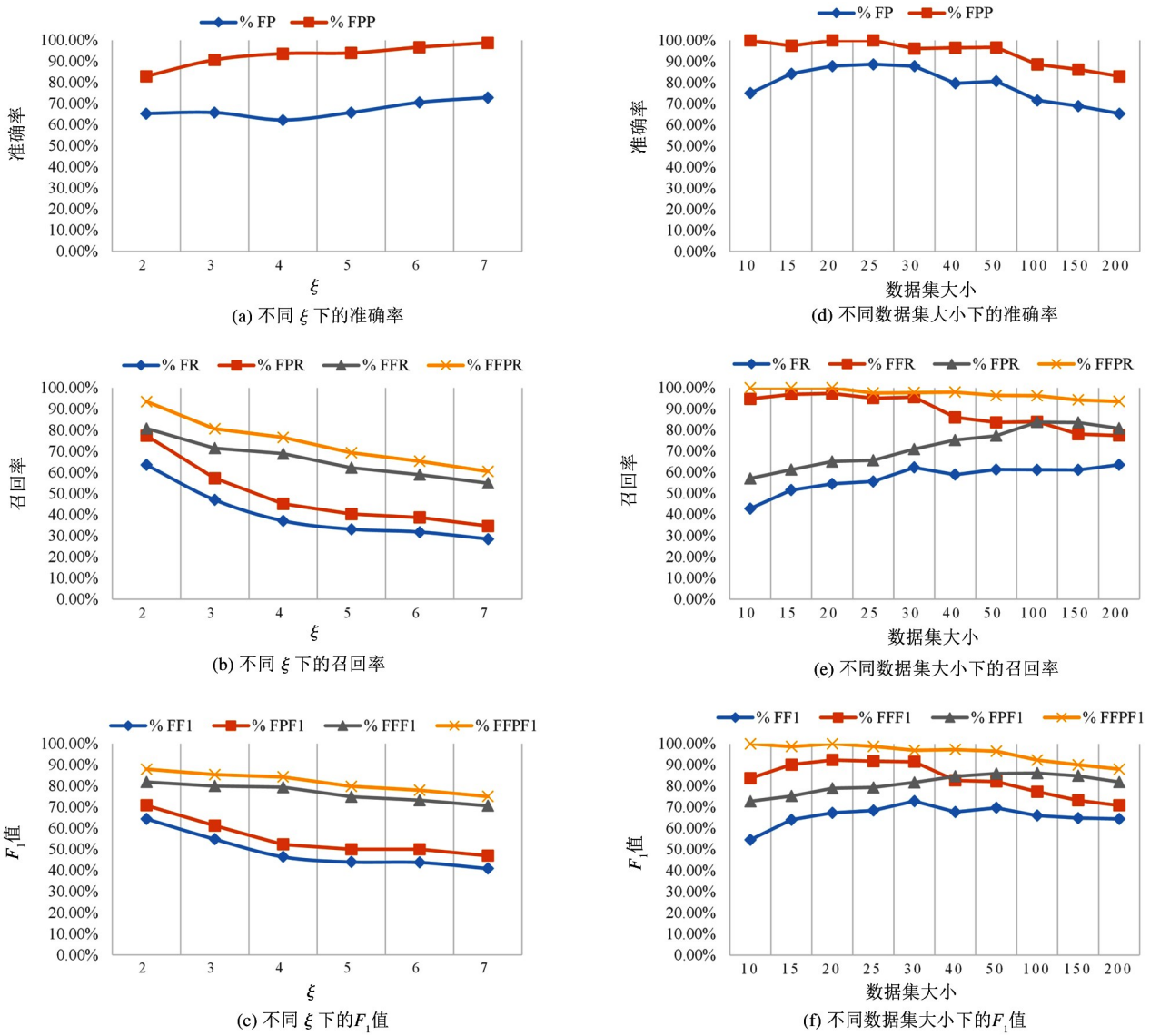
5.5 影响频繁属性挖掘的因素分析

最小支持度阈值 ξ 显然是影响属性挖掘的关键因素之一。此外,算法的效果也会受到数据集大小的影响。本文设计 2 组实验来验证其影响。在 2 组实验中,都以颌面外科的数据作为研究对象。在第 1 组实验中,利用 200 个病历作为实验数据,ξ 分别设置为 2、3、4、5、6、7。第 2 组实验中,设置 ξ =2 来观察不同大小数据集下算法的性能变化。2 组实验结果如图 4 所示。

第 1 组实验结果表明,随着 ξ 的增加,准确率逐渐提高,召回率和 *F*1 值逐渐降低。*F*1 值在 ξ =2 时取得最大值,这说明 Attrimining 算法在较低的支持度下有更好的表现。第 2 组实验结果显示,准确率曲线、召回率曲线和 *F*1 值曲线都有先上升后下降的趋势。这些曲线反映适当增加训练数据可以提高算法性能。但是,当数据规模超过一定阈值时并不会改善抽取效果。一方面,说明 Attrimining 算法在少量的数据集情形下能取得良好的结果。另一方面,实验结果也表明,对于大规模属性抽取任务,对数据采取增量式分批抽取的策略能够达到更好的效果。

6 结论

本文形式化属性挖掘问题,并设计一种有效的算法。同时,设计了一种有效的候选属性得分函数获取最可信的属性。在真实数据集下进行实验,实验结果证明了本文所提方法的合理性和有效性。未来将使用提取到的属性来提取“〈属性,值〉对”,构



(a) 不同 ξ 下的准确率　(d) 不同数据集大小下的准确率

(b) 不同 ξ 下的召回率　(e) 不同数据集大小下的召回率

(c) 不同 ξ 下的F_1值　(f) 不同数据集大小下的F_1值

图 4　Attrimining 在不同 ξ 和不同大小数据集下的性能变化

建口腔领域的中文口腔知识图谱,并将其应用到辅助治疗、自助注册、医嘱自动录入、临床决策支持、人机交互咨询等医学应用中。

参考文献

[5] GHANI R, PROBST K, LIU Y, et al. Text mining for product attribute extraction[J]. *ACM SIGKDD Explorations Newsletter*, 2006, 8(1): 41-48

[6] CHAKRABORTY S, SUBRAMANIAN L, NYARKO Y. Extraction of (key, value) pairs from unstructured ads [C]// AAAI Fall Symposium Serie, Quebec, Canada, 2014: 10-17

[1] WU Y F B, LI Q, BOT R S, et al. Domain-specific keyphrase extraction[C]// Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 2005:283-284

[2] LIU Z, HUANG W, ZHENG Y, et al. Automatic keyphrase extraction via topic decomposition[C]// Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, USA, 2010:366-376

[3] STERCKX L, DEMEESTER T, DELEU J, et al. Topical word importance for fast keyphrase extraction[C]// Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 2015: 121-122

[4] TOMOKIYO T, HURST M. A language model approach to keyphrase extraction[C]// Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, Sapporo, Japan, 2003: 33-40

[7] PUTTHIVIDHYA D, HU J. Bootstrapped named entity recognition for product attribute extraction[C]// Proceed-

ings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 2011: 1557-1567
[8] MOONEY R J, BUNESCU R. Mining knowledge from text using information extraction[J]. *ACM SIGKDD Explorations Newsletter*, 2005, 7(1): 3-10
[9] 马进,杨一帆,陈文亮. 基于远程监督的人物属性抽取研究[J]. 中文信息学报, 2020, 34(6): 64-72
[10] 叶大枢, 黄沛杰, 邓振鹏, 等. 限定领域口语对话系统中的商品属性抽取[J]. 中文信息学报, 2016, 30(6): 67-74
[11] REISINGER J, PASCA M. Bootstrapped extraction of class attributes[C]// Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 2009: 1235-1236
[12] BELLARE K, TALUKDAR P P, KUMARAN G, et al. Lightly-supervised attribute extraction for web search[C]// Proceedings of NIPS 2007 Workshop on Machine Learning for Web Search, Vancouver, Canada, 2007: 44-68
[13] LU R, JIA C, ZHANG S, et al. An exact data mining method for finding center strings and all their instances [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2007, 19(4): 509-522
[14] LEEMANS M, VAN DER AALST W M. Discovery of frequent episodes in event logs[C]// Proceedings of International Symposium on Data-Driven Process Discovery and Analysis, Milan, Italy, 2014:1-31
[15] RINK B, HARABAGIU S, ROBERTS K. Automatic extraction of relations between medical concepts in clinical texts[J]. *Journal of the American Medical Informatics Association*, 2011, 18(5): 594-600
[16] ROLLER R, USZKOREIT H, XU F, et al. A fine-grained corpus annotation schema of German nephrology records[C]// Proceedings of the Clinical Natural Language Processing Workshop, Osaka, Japan, 2016: 69-77
[17] SOYSAL E, CICEKLI I, BAYKAL N. Design and evaluation of an ontology based information extraction system for radiological reports[J]. *Computers in Biology and Medicine*, 2010, 40(11-12): 900-911
[18] AGRAWAL R, SRIKANT R. Mining sequential patterns [C]// Proceedings of the 11th International Conference on Data Engineering, Taibei, China, 1995: 3-14
[19] HAN J, CHENG H, XIN D, et al. Frequent pattern mining: current status and future directions[J]. *Data Mining and Knowledge Discovery*, 2007, 15(1): 55-86
[20] SRIKANT R, AGRAWAL R. Mining sequential patterns: generalizations and performance improvements[C]// Proceedings of International Conference on Extending Database Technology, Avignon, France, 1996: 1-17
[21] HAN J, PEI J, MORTAZAVI-ASL B. FreeSpan: frequent pattern-projected sequential pattern mining[C]// Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, USA, 2000: 355-359
[22] HAN J, PEI J, MORTAZAVI-ASL B. PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth[C]// Proceedings of 17th International Conference on Data Engineering, Heidelberg, Germany, 2001: 215-224

Attribute mining from Chinese electronic medical records

FEI Chaoqun[*,**,***], ZHANG Shuhan[*,**,***], LI Yangyang[****,*****]
([*] Key Laboratory of Intelligent Information Processing, Beijing 100190)
([**] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)
([***] University of Chinese Academy of Sciences, Beijing 100049)
([****] Key Lab of Management, Decision and Information Systems, Beijing 100190)
([*****] Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190)

Abstract

The task of mining frequent attributes from the electronic medical record (EMR) aims at extracting medical examination items from a group of diagnosis records produced by the same clinic. The traditional frequent item set or sequence mining techniques can not be directly applied to the case. This paper proposes a tag free technique to overcome the problem and presents a novel extraction framework which can avoid manual tagging. Namely, it formulates the attribute mining problem as a task of semi-structured frequent subsequence mining problem and proposes an effective algorithm to mine candidate word patterns from written EMRs. Comprehensive experimental results on Chinese EMRs show that the proposed method can tackle attribute mining problem effectively.

Key words: attribute mining, electronic medical record (EMR), frequent subsequence mining, word pattern, frequent word pattern