可计算性逻辑中 CoL2 系统的可判定性分析

李兴香 栾峻峰
(山东大学计算机科学与技术学院 济南 250101)

摘 要 可计算性(Computability)即算法有解性,是数学和计算机科学领域中重要的概念之一。可计算性逻辑(Computability Logic,CoL)是关于可计算性的形式理论,是一种交互的资源逻辑。其中,CoL2 系统采用博弈的语义,是对经典命题逻辑的扩展,在经典命题逻辑的基础上添加了选择运算和一般原子,比经典命题逻辑更富有表达力,具有更广阔的应用前景,并且有较高的证明效率。分析了 CoL2 系统的可判定性,即通过提出一个算法来判断任意一个 CoL2 公式是否是可证明的,并且证明了该算法是多项式空间内运行的。

关键词 可计算性逻辑,CoL2,交互计算,博弈语义

中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.7.010

Research on Decidability of CoL2 in Computability Logic

LI Xing-xiang LUAN Jun-feng
(Department of Computer Science and Technology, Shandong University, Jinan 250101, China)

Abstract Computability or algorithm solvablity, is one of the important concepts in the field of mathematics and computer science. Computability logic (abbreviated as CoL) is a formal theory of computability and an interactive resource logic. CoL2 logic uses game semantics. CoL2 is an extension of classical propositional logic. What makes CoL2 more expressive than classical propositional logic is the presence of choice operators and general atom. CoL2 has a higher proof efficiency. In this paper, the decidability of CoL2 was presented, in other words, by presenting an algorithm whether any CoL2 formula is provable can be determined, and we proved that the algorithm runs in polynomial space.

Keywords Computability logic, CoL2, Interactive algorithms, Game semantics

1 引言

可计算性(computability)即算法有解性,是数学和计算机科学领域中重要的概念之一。可计算性逻辑(Computability Logic,CoL)是 G. Japaridze 在 2003 年提出的一种交互的资源语义,是关于可计算性的形式理论[1]。

CoL2 在语义上采用"博弈"语义,是相较于经典逻辑"真值"语义的一次质的改变。现实生活中的大多数计算问题是用户与计算机之间复杂的多输入输出的交互过程,可计算问题的解决意味着计算机最终赢得了这个博弈,即输出了正确的答案。因此,可将可计算问题等同于博弈问题[4,10]。可计算性是可计算问题的特性,问题的可计算性是指存在一台计算机赢得这个博弈。可计算问题也称为可计算任务,可以看作是两个角色(即计算机和外部环境)之间的博弈或对话。传统的问题可以看作是一个简单博弈,这类博弈是由提出问题(输入 input)和回答问题(输出 output)组成的两步交互对话。因为大多数计算机和计算机网络执行的任务都是交互任务,所以博弈的长度不一定是两步,可以是多步的[9]。

可计算性逻辑中的公式代表交互计算问题,逻辑运算代表在可计算问题上的交互操作,一个公式是有效的是指存在解决相应问题的方法(或者等价的,即存在赢得博弈的策略)。

经典逻辑是可计算性逻辑的一部分,这是因为经典逻辑的谓词就是一类特殊的可计算问题——零交互的问题,而且经典逻辑中的真(truth)的概念就相当于可计算问题中的可计算性的概念。因此可计算性逻辑是经典逻辑的扩展,而可计算性逻辑在应用理论方面更富有表达力和建设性,并且有较高的证明效率[11]。

文献[2,3,5,6]先后阐述了可计算性逻辑 CoL 的特殊部分 CoL1、CoL2、CoL3 和 CoL4,并且证明了它们的可靠性和完备性,这为可计算性逻辑的应用提供了逻辑基础。本文在此基础上给出了 CoL2 系统的可判定性证明。

2 可计算性逻辑基础

本节简单介绍可计算性逻辑中的一些基本概念。

可计算性逻辑中包含很多符号[7],这里主要介绍以下符号:(1)个体常量符:$a,b,c\cdots$;(2)个体变量符:$x,y,z\cdots$;(3)逻辑常量符:$\top$和$\bot$;(4)简单原子符:$p,q,r,s,\cdots$;(5)一般原子符:$P,Q,R,S,\cdots$;(6)连接词符:$\neg$,$\wedge$,$\vee$,$\sqcup$,$\sqcap$,$\rightarrow$。

逻辑常量"$\top$"表示总是可计算的简单问题,逻辑常量"$\bot$"表示总是不可计算的简单问题。简单原子代表简单的零交互问题,它与经典命题逻辑中的谓词相对应;一般原子代表复杂的交互问题。$\neg$称为否定运算符,是改变角色的操作:$\neg$

到稿日期:2014-07-15 返修日期:2014-10-25 本文受国家自然科学基金项目(61202014)资助。
李兴香(1988—),女,硕士生,主要研究方向为可计算性逻辑,E-mail:lixingxiang828@163.com;栾峻峰(1974—),男,博士,副教授,主要研究方向为可计算性逻辑、计算生物学。

将博弈中计算机(外部环境)的合法移动变成外部环境(计算机)的合法移动;若计算机(外部环境)是获胜者,则经否定运算后,外部环境(计算机)变成获胜者。$\wedge$和$\vee$称为并行运算符,博弈$A_1 \wedge A_2(A_1 \vee A_2)$表示两个博弈$A_1$和$A_2$同时进行,只有同时赢得这两个(至少一个)博弈,计算机才能最终赢得博弈$A_1 \wedge A_2(A_1 \vee A_2)$。其中否定运算和并行运算与其在经典命题逻辑中的含义基本相同。$\sqcup$、$\sqcap$称为选择运算符,选择合取$A_1 \sqcap A_2$的含义是:初始时由外部环境先选择其中的一个博弈A_i,然后继续进行博弈A_i;若初始时外部环境不进行选择,则计算机赢得这个博弈。选择析取$A_1 \sqcup A_2$的含义是:初始时由计算机先选择其中的一个博弈A_i,然后继续进行博弈A_i;若初始时计算机不进行选择,则计算机输。$\rightarrow$表示归约运算符,其定义为:$B \rightarrow A=(\neg B) \vee A$。由定义可以看出,当该运算符应用到简单博弈(只有一步输入和一步输出的博弈)时,归约运算与其在经典命题逻辑中的含义相同,相当于经典命题逻辑中的蕴涵。实际上,$B \rightarrow A$就是将A归约到B的问题:解决$B \rightarrow A$也就是把B当作可计算资源(computational resources)来解决问题A。资源(resources)与问题(problems)是对称的概念:对一个玩家来说,要解决的问题是另外一个玩家所利用的资源,反之亦然。由于在$(\neg B) \vee A$中B是否定的(否定意味着改变角色),因此在$B \rightarrow A$中B是计算机的一个资源而不是问题。

定义 1(项,term) (1)个体常量和个体常量均为项;(2)设f是n元函数符号,$t_1, t_2, \cdots, t_n$是项,则$f(t_1, t_2, \cdots, t_n)$也是项。

定义 2(原子公式,atom) (1)两个逻辑常量是原子公式,称为逻辑原子公式;(2)设L是简单或一般谓词符,$t_1, t_2, \cdots, t_n$是项,则$L(t_1, t_2, \cdots, t_n)$分别为简单原子公式或一般原子公式。

定义 3(公式) (1)原子公式是公式;(2)若$F_1, F_2, \cdots, F_n$是公式,则$\neg F_1, F_1 \wedge F_2 \wedge \cdots \wedge F_n, F_1 \vee F_2 \vee \cdots \vee F_n, F_1 \sqcap F_2 \sqcap \cdots \sqcap F_n, F_1 \rightarrow F_2$也是公式,其中$F \rightarrow G$为$(\neg F) \vee G$的缩写。

公式F中运算符的数目称为F的复杂度,F中运算符的数目与一般原子的出现次数之和称为F的总复杂度(aggregate complexity)。

定义 4 若函数$*$满足以下两个条件,则称$*$为解释(interpretation):

(1)$*$把每个n元简单谓词p映射到简单问题$p^*(x_1, x_2, \cdots, x_n)$;

(2)$*$把每个n元一般谓词P映射到一般问题$P^*(x_1, x_2, \cdots, x_n)$。

每个解释$*$均可按经典的方法扩充到所有公式,即:$\perp^*=\perp; \top^*=\top; (\neg F)^*=\neg(F^*); (F_1 \wedge \cdots \wedge F_n)^*=F_1^* \wedge \cdots \wedge F_n^*$;运算符$\vee$、$\sqcap$、$\sqcup$类似。我们称$*$将$F$解释为$A$意味着$F^*=A$。通俗地讲,解释$*$给公式赋予了“实际意义”;一个公式$F$只是一串符号,而解释$F^*$是一个博弈问题。

定义 5 一个计算问题A是可完成的当且仅当存在一个M完成A,此M称为A的一个解决方案,记为$M \models A$。

定义 6 一个公式是有效的当且仅当对F的每个解释$*$,F^*是可完成的。

定义 7 F是统一有效的当且仅当存在一个M使得对F的任意解释$*$,$M \models F^*$,并且称M为F的统一解决方案。

公式F的统一解决方案是一个不依赖于解释而能保证完成F的策略。

3 CoL2 系统

前面介绍的逻辑基础是 CoL2 语言中所包含的。

CoL2 是经典命题逻辑的扩展,是可计算性逻辑中的命题逻辑部分[8]。CoL2 比经典的命题逻辑更富有表达力,这不仅是因为它增加了新的逻辑运算符,从而可表达更细致的知识,而且是因为 CoL2 中有两种原子:简单原子(elementary atom)和一般原子(general atom)。简单原子代表简单的零交互问题,通常用p、q、r、s等表示,它与经典命题逻辑中的谓词相对应,经典命题逻辑中的谓词可以看作是 CoL2 中的简单原子;一般原子代表复杂的交互问题,通常用P、Q、R、S等表示,因此,可以把可计算问题等同于博弈问题。

CoL2 中,一个子公式在偶数(奇数)个$\neg$辖域中出现称为正(或负)出现。子公式不在任何选择运算符的辖域中出现称为表面出现。不含一般原子和选择运算符的公式(即经典一阶谓词逻辑公式)称为简单公式表示。一个公式F的简化公式(用$\| F \|$表示)是将F中所有形如$G_1 \sqcap \cdots \sqcap G_n$的子公式的表面出现用$\top$代替,所有形如$G_1 \sqcup \cdots \sqcup G_n$的子公式的表面出现用$\perp$代替,并且每个一般原子的正表面出现用$\perp$代替,每个一般原子的负表面出现用$\top$代替后的公式。一个公式是稳定的(stable)当且仅当它的简化公式是经典重言式,即在经典命题逻辑中是可证的;否则它就是不稳定的。其中$F \rightarrow G$为$(\neg F) \vee G$的缩写。

CoL2 系统由下面 3 条规则给出:

规则 A:$\vec{H} \vdash F$,其中F是稳定的且$\vec{H}$是最小公式集满足:当F中含有子公式$G_1 \sqcap \cdots \sqcap G_n$(或$G_1 \sqcup \cdots \sqcup G_n$)的一个正(或负)表面出现,则对每个$i \in \{1, \cdots, n\}$,$\vec{H}$包含用$G_i$代替该出现后的公式。

规则 B:$H \vdash F$,其中H是将F中的子公式$G_1 \sqcap \cdots \sqcap G_n$(或$G_1 \sqcup \cdots \sqcup G_n$)的一个负(或正)表面出现用$G_i (i \in \{1, \cdots, n\})$代替后的公式。

规则 C:$F \vdash F'$,其中F'是将F中的某个一般原子的两个(一正一负)表面出现的同时用一个未在F中出现的简单原子代替后的公式。

规则 A 中的前提$\vec{H}$是空集{}时,F是稳定的且不含选择运算,则它推出的F就是经典逻辑中的定理,这些定理就是 CoL2 中的公理。当一个公式F能够由 CoL2 的 3 条规则推导出时,我们称F是可证明的(provable),记为 CoL2 $\vdash F$。

下面是 CoL2 公式$P \sqcup Q \rightarrow P \vee Q$的证明:

(1)$p \rightarrow p \vee Q$	{},规则 A
(2)$P \rightarrow P \vee Q$	(1),规则 C
(3)$q \rightarrow P \vee q$	{ },规则 A
(4)$Q \rightarrow P \vee Q$	(3),规则 C
(5)$P \sqcup Q \rightarrow P \vee Q$	{(2),(4)},规则 A

文献[3]中已经证明了下面两个定理,说明了 CoL2 系统是可靠的和完备的。

定理 1 对任意的公式F,CoL2 $\vdash F$当且仅当F是有效的。并且:

a) 可靠性:存在一个有效的程序能够根据F的 CoL2 证明构造出一个F的统一解决方案M,即对于给定的解释$*$,

M 能够计算出 F^*；

b) 完备性：若 CoL2 $\nvdash F$，则将 F 中所有简单原子解释为有限元的谓词，将 F 中所有一般原子解释为一般问题的某个解释 $*$，F^* 是不可计算的。

定理 2 一个 CoL2 公式 F 是有效的当且仅当它是统一有效的。

4 CoL2 的可判定性

CoL2 系统是可判定的是指存在一个有效的算法来判断任意一个 CoL2 公式是否是可证明的。本节将给出判断 CoL2 公式的可证明性的算法，并且分析其复杂性。下面给出判断 CoL2 公式的可证明性问题的形式定义。

实例：给定 CoL2 公式 F。

询问：F 是不是可证明的，即 CoL2 $\vdash F$?

定理 3 判断 CoL2 公式的可证明性问题是多项式空间内可判定的。

证明：令 F 为任意一个 CoL2 公式。F 是可证明的当且仅当 F 可以由某些可证明的公式通过规则 A、B 或 C 推导出来。判断 CoL2 $\vdash F$ 这个问题的复杂度可以通过对 F 的总复杂度(F 中运算符的数目与一般原子的出现次数之和)的归纳进行证明。

下面首先给出解决这个判定问题的算法。这是一个递归的算法，递归的次数与公式 F 的总复杂度的值相等。

算法 判断 CoL2 公式的可证明性问题

输入：CoL2 公式 F

输出：若 F 是可证明的，输出 yes；否则，输出 no

算法：递归地调用下面 3 个步骤：

1. 测试规则 A：这一测试包含下面两个步骤
   (1)检查 F 是否是稳定的，即 $\|F\|$ 是否是经典永真的。若是稳定的，则进入(2)，否则进入 2；
   (2)然后对于 F 中的子公式 $G_1\sqcap\cdots\sqcap G_n$(或 $G_1\sqcup\cdots\sqcup G_n$)的每个正(或负)表面出现，对于每个 $i\in\{1,\cdots,n\}$，检查 H_i 是否是可证明的，其中 H_i 是用 G_i 代替 F 中上述出现后的公式。若所有的 H_i 都是可证明的，则返回 yes，否则进入 2；
2. 测试规则 B：对 F 中的子公式 $G_1\sqcap\cdots\sqcap G_n$(或 $G_1\sqcup\cdots\sqcup G_n$)的每个负(或正)表面出现，检查 H_i 是否是可证明的，其中 H_i 是用 G_i 代替 F 中上述出现后的公式。若某个 H_i 是可证明的，则返回 yes，否则进入 3；
3. 测试规则 C：对 F 中某一个一般原子 P 的每对(一正一负)表面出现，用未曾出现在 F 中的简单原子 q 代换，检查代换后的公式 H 是否是可证明的。若 H 是可证明的，则返回 yes；否则返回 no。

根据上述算法可以得出，CoL2 系统中的公式 $(p\sqcap Q)\vee(p\sqcap R)\rightarrow p\sqcap(Q\vee R)$ 的证明可以用图 1 的树来表示。

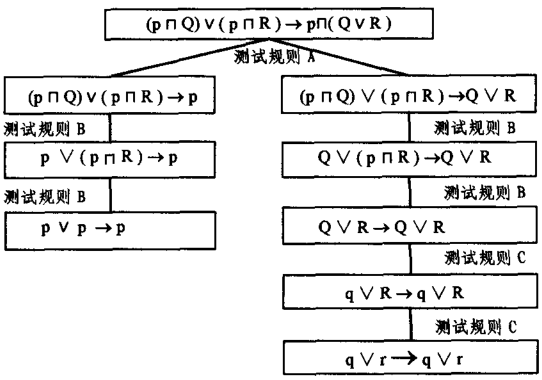

图 1 公式 $(p\sqcap Q)\vee(p\sqcap R)\rightarrow p\sqcap(Q\vee R)$ 的证明树

根据上述证明树的构造过程，可以得出在 CoL2 公式中 $(p\sqcap Q)\vee(p\sqcap R)\rightarrow p\sqcap(Q\vee R)$ 的证明：

(1) $p\vee p\rightarrow p$ {}，规则 A

(2) $p\vee(p\sqcap R)\rightarrow p$ (1)，规则 B

(3) $(p\sqcap Q)\vee(p\sqcap R)\rightarrow p$ (2)，规则 B

(4) $q\vee r\rightarrow q\vee r$ {}，规则 A

(5) $q\vee R\rightarrow q\vee R$ (4)，规则 C

(6) $Q\vee R\rightarrow Q\vee R$ (5)，规则 C

(7) $Q\vee(p\sqcap R)\rightarrow Q\vee R$ (6)，规则 B

(8) $(p\sqcap Q)\vee(p\sqcap R)\rightarrow Q\vee R$ (7)，规则 B

(9) $(p\sqcap Q)\vee(p\sqcap R)\rightarrow p\sqcap(Q\sqcap R)$ (3)(8)，规则 A

下面分析这个算法的正确性及其复杂度。对于算法步骤 1 中的(1)，判断 F 是否是稳定的，即 $\|F\|$ 是否是经典永真的，这个问题显然是在有限时间内可判定的；在步骤(2)中，每个 H_i(其中 $i\in\{1,\cdots,n\}$)的总复杂度都小于 F，因此由归纳假设判断 CoL2 $\vdash H_i$ 只需要有限的时间。F 由规则 A 推导出当且仅当步骤 1 返回 yes。因此若步骤 1 返回 yes，意味着 F 是稳定的，且所有的 H_i 满足规则 A 中的条件。类似地，也可以在有限的时间内完成步骤 2。F 由规则 B 推导出当且仅当步骤 2 返回 yes。在步骤 3 中，替换后的公式 H 的总复杂度显然小于 F 的总复杂度。由归纳假设，H 的可证明性是可以判定的，并且由于 F 中一般原子的数目是有限的，q 的选择显然也是无关紧要的(这是因为如果选择了另一个没有出现在 F 中的简单原子 q'，输出结果也是一样的)，因此步骤 3 也可以在有限的时间内完成，并且步骤 3 输出 yes 当且仅当 F 是由规则 C 推导出来的。同时由上述分析可以看出，这个算法是可以在多项式空间内完成的[12]。

对于任意给定的一个 CoL2 公式，都可以用该算法判断此 CoL2 公式是否是可证明的，即此 CoL2 公式是否可由 CoL2 的 3 条规则推导出。因此，可以得出 CoL2 系统是可判定的。

结束语 可计算性逻辑是 Japaridze 在 2003 年提出来的。目前对可计算性逻辑研究还处于初级阶段，可以说是冰山一角，需解决的问题的数量远远超过已解决的问题的数量，还有很多未开垦的领域和未解决的问题，为那些对逻辑和逻辑在计算机方面的应用，以及对计算人工智能理论感兴趣的人们提供了很多研究和发现的好机会。我们未来的工作将围绕可计算性逻辑在应用领域中展开，主要是它在知识库系统、规划和行为系统等应用系统的应用。我们可以将可计算性逻辑的知识应用到系统中，并且用可计算性逻辑准确地表达实际问题作为未来的研究目标，真正实现将可计算性逻辑作为在计算机应用领域的逻辑基础。

参 考 文 献

[1] Japaridze G. Introduction to computability logic [J]. Annals of Pure and Applied Logic, 2003, 123: 1-99

[2] Japaridze G. Propositional computability logic I [J]. ACM Transactions on Computational Logic, 2006, 7(2): 302-330

[3] Japaridze G. Propositional computability logic II [J]. ACM Transactions on Computational Logic, 2006, 7(2): 331-362

[4] Japaridze G. Computability logic: a formal theory of interaction. Interactive Computation: TheNew Paradigm [C] // Goldin D, Smolka S, Wegner P, eds. Springer Verlag, Berlin, 2006: 183-223

[5] Japaridze G. From truth to computability I [J]. Theoretical Computer Science, 2006, 357: 100-135

[6] Japaridze G. From truth to computability II [J]. Theoretical Computer Science, 2007, 379: 20-52

[7] Japaridze G. In the beginning was game semantics. Games: Unifying Logic, Language and Philosophy[C]// Majer O, Pietarinen A-V, Tulenheimo T, eds. Springer, 2009: 249-350

[8] Japaridze G. The propositional logic of elementary tasks [J]. Notre Dame Journal of Formal Logic, 2000, 41(2): 171-183

[9] Sipser M. Introduction to the Theory of Computation [M]. Thompson, 2006

[10] Xu W, Liu S. Deduction theorem for symmetric cirquent calculus [J]. Advances in Intelligent and Soft Computing, 2010, 82: 121-126

[11] Blass A. A game semantics for linear logic[J]. Annals of Pure and Applied Logic, 1992, 56: 183-220

[12] 朱大铭,马绍汉. 算法分析与设计[M]. 北京:高等教育出版社, 2009
Zhu Da-ming, Ma Shao-han. Design and Analysis of Algorithms [M]. Beijing: Higher education press, 2009

(上接第 18 页)

4.2 实验结果

本实验使用 3 个指标来衡量命名实体识别的性能:正确率、召回率、F-值。其计算公式如下:

$$正确率(P)=\frac{系统正确识别的实体个数}{系统识别的实体个数}\times 100\%$$

$$召回率(R)=\frac{系统正确识别的实体个数}{文档中的实体总数}\times 100\%$$

$$F\text{-}值=\frac{2\times P\times R}{P+R}\times 100\%$$

对实验 1、实验 2、实验 3 的命名实体识别结果进行正确率、召回率、F-值的计算,结果如表 3 所列。

表 3 实验结果

实验	文本类型	实体总数	识别个数	正确个数	正确率/%	召回率/%	F-值/%
CRF	军用文书	663	644	559	86.80	84.31	85.54
	网络军事文本	1045	1019	869	85.28	83.16	84.21
CRF+Self-Training	军用文书	663	647	579	89.49	87.33	88.40
	网络军事文本	1045	1017	892	87.71	85.36	86.52
CRF+Self-Training+两种校正方法	军用文书	663	646	602	93.19	90.80	91.98
	网络军事文本	1045	1027	929	90.46	88.90	89.67

实验 1 仅使用了条件随机场模型,在军用文书中识别的正确率、召回率、F-值仅为 86.8%、84.31%、85.54%,在网络军事文本中识别的正确率、召回率、F-值仅为 85.28%、83.16%、84.21%,效果不太理想,没有达到进行军事应用的标准。

实验 2 使用了 Self-Training 算法对条件随机场模型进行迭代学习,在军用文书中识别的正确率、召回率、F-值分别为 89.49%、87.33%、88.4%,在网络军事文本中识别的正确率、召回率、F-值为 87.71%、85.36%、86.52%,该方法解决了大量标注语料难以获取的问题,提高了识别的性能。

实验 3 在实验 2 的基础上依次采用基于词典的方法和基于规则的方法对识别结果进行校正,在军用文书中最终识别的正确率、召回率、F-值可达到93.19%、90.8%、91.98%,在网络军事文本中最终识别的正确率、召回率、F-值可达到 90.46%、88.9%、89.67%,使得在军事文本中的命名实体识别性能达到与通用领域相当的水平。实验 3 的方法在继承了基于统计学习模型方法的基础上同时吸收了基于词典和基于规则方法的优点,其在军用文书中识别的正确率、召回率、F-值相比于实验 2 分别提高了 3.7%、3.47%、3.58%,在网络军事文本中识别的正确率、召回率、F-值相比于实验 2 分别提高了 2.75%、3.54%、3.15%,训练语料相对于海量的军事文本来说数量明显不足。采用两种校正方法可准确识别出更多的复合词和嵌套词,使识别的正确率和召回率都有大幅度的提高。

上述 3 组对比实验中,对军用文书识别的性能始终略高于对网络军事文本识别的性能,其主要原因在于军用文书文体较为规范,而网络军事文本中普遍存在较多口语化、网络化的词汇,因而为命名实体识别造成较大困难。

结束语 本文对军事文本和军事命名实体进行定义和研究,并针对其特点提出了一种面向军事文本的命名实体识别方法。该方法不仅同时使用多种原子特征模板和复合特征模板进行特征的表达,还采用了半监督学习 Self-Training 算法对条件随机场模型进行迭代学习。通过对比在人工搜集的军事文本上进行的 3 组实验结果,表明该方法能够有效解决军事文本中的命名实体识别问题,并能获得较好的识别性能。如何进一步提高命名实体识别的速度,是下一步研究的重点。

参 考 文 献

[1] Tjong Kim Sang E F, De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition[C]// Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003: 142-147

[2] McCallum A, Li W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons[C]// Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003: 188-191

[3] 向音. 军用文书的语篇特征初探[J]. 办公室业务, 2011, 10: 19-20
Xiang Yin. The preliminary study about discourse features in the military document[J]. Office Operations, 2011, 10: 19-20

[4] 邱泉清,苗夺谦,张志飞. 中文微博命名实体识别[J]. 计算机科学, 2013, 40(6): 196-198
Qiu Quan-qing, Miao Duo-qian, Zhang Zhi-fei. Named entity recognition on chinese microblog[J]. Computer Science, 2013, 40(6): 196-198

[5] 张晓艳,王挺,陈火旺. 命名实体识别研究[J]. 计算机科学, 2005, 32(4): 44-48
Zhang Xiao-yan, Wang Ting, Chen Huo-wang. Research on named entity recognition[J]. Computer Science, 2005, 32(4): 44-48

[6] 李航. 统计学习方法[M]. 北京:清华大学出版社, 2012: 191-210
Li Hang. statistical learning methods[M]. Beijing: Tsinghua University Press, 2012: 191-210

[7] Nadeau D, Sekine S. A survey of named entity recognition and classification[J]. Lingvisticae Investigationes, 2007, 30(1): 3-26

[8] 黄鸿,李见为,冯海亮. 基于半监督流形学习的人脸识别方法[J]. 计算机科学, 2009, 35(12): 220-223
Huang Hong, Li Jian-wei, Feng Hai-liang. Face recognition based on semi-supervised manifold learning[J]. Computer Science, 2009, 35(12): 220-223