

## 计算语言学相关课程的设置和教学<sup>1</sup>

刘海涛 侯敏 李晓华 冯志伟  
中国传媒大学应用语言学系，北京 100024  
{byliuhaitao, byhoumin, bylixiaohua}@cuc.edu.cn

**摘要:** 本文介绍了中国传媒大学应用语言学系计算语言学方向有关课程的设置和教学情况，提出计算语言学课程的设置应该突出学科的交叉性特点，同时要注意可接受性及适用性。

**关键词:** 计算语言学 课程设置 教学

## Teaching Computational Linguistics

LIU Haitao HOU Min LI Xiaohua FENG Zhiwei  
Communication University of China, Beijing 100024  
{byliuhaitao, byhoumin, bylixiaohua}@cuc.edu.cn

**Abstract:** This paper presents curriculum and teaching related with computational linguistics at Communication University of China. The authors propose that the curriculum should be intersecting, acceptable and suitable.

**Keywords:** computational linguistics curriculum teaching

一般认为“计算语言学”的历史始于1947年，这一年人们开始考虑使用刚诞生不久的计算机进行自动翻译的问题<sup>2</sup>。机器翻译的研究奠定了计算语言学作为一个研究领域得以存在的基础。著名机器翻译历史的研究者 John Hutchins 认为：机器翻译的先锋们所做的工作不仅仅有益于机器翻译这一领域本身的研究和发展，而且对于计算语言学、自然语言处理以及从人工智能的角度研究语言问题，都有极大的意义<sup>[1]</sup>。经过50年的发展，今天的计算语言学几乎已经成为一门显学了。越来越多的大学开始设立计算语言学的专业，培养各个层次的专门人才。接下来的问题是，如果我们在大学设立了计算语言学专业，我们应该教学生些什么呢？如何设置课程才能让学生掌握基本的计算语言学知识，而这些知识又有益于学科的建设与学生未来的发展呢？这一问题虽然不是什么计算语言学的前沿课题，却是一个不容忽视的问题，因为对于任何称得上是学科的研究领域而言，其高等教育的相关课程大致形成了该学科的理论和研究基础，因此也是非常重要的。这一问题，也正引起越来越多同行的注意和兴趣，如2002年在美国宾州大学召开了题为“Effective Tools and Methodologies for Teaching NLP and CL”的研讨会；计算语言学的领头羊机器翻译在2003年第九届机器翻译峰会(MT Summit)期间也召开了题为 T<sup>4</sup> (Teaching Translation Technologies and Tools) 的研讨会，而类似的有关机器翻译教学的会议此前已召开过两次。

中国传媒大学从2000年开始招收应用语言学专业的本科学生，目前已形成一个计算语言学方向本科生、硕士生和博士生的层级培养体系。本文介绍了我们对于计算语言学一些基本

<sup>1</sup> 虽然本文题目中用了“计算语言学”的字样，但文章内容对于自然语言处理、自然语言理解、中文信息处理等课程和专业的设置也有参考意义。因为文章的重点在于如何针对学科的交叉性设置课程与教学。

<sup>2</sup> 有关机器翻译的早期历史参见 Hutchins(2000)。

概念的理解以及在课程设置和主要课程教学方面的一些想法和做法。我们希望本文一方面能得到各兄弟院校、各位专家的批评指导，另一方面能引起人们对文科背景计算语言学专业课程的设置和各级计算语言学人才培养计划制定的重视，当然也希望能对各兄弟院校相关专业的课程设置有些参考作用。

## 1. 计算语言学的基本定义

什么是计算语言学？这是一个至今没有一致答案的问题。对于研究用计算机处理自然语言具体应用的学者而言，可以完全不必纠缠于这个问题。但要在大学开设计算语言学专业，授予各级学位，则不能回避这个问题。好在我们有诸如 Robert Dale/Moisl/Somers<sup>[2]</sup> 和 Mitkov<sup>[3]</sup>等权威的计算机语言学和自然语言处理的手册可供参考，也可以从 Carstensen 等<sup>[4]</sup>和 Jurafsky/Martin<sup>[5]</sup>编的大部头教科书中提取必要的信息。

简单说来，计算语言学是一门研究用计算机处理语言的交叉性学科<sup>[3]</sup>。也有人说“**计算语言学**是研究用机器来处理自然语言的学科。它是由信息技术和语言学交叉而成的”<sup>[4]</sup>。这两个定义虽然简单，但道出了计算语言学的特殊之处，它是一门交叉学科，其内容涉及语言学、信息技术、哲学（特别是语言哲学和逻辑学）、人工智能、认知科学、数学（数理逻辑、自动机和形式语言、图论、统计学等）<sup>[4]</sup>。关于计算语言学学科的交叉性，我国学者也多有论述，冯志伟在自己的多部著作中都认为语言的计算机处理涉及语言学、计算机技术和数学等领域，从而强调培养跨学科人才的重要性<sup>[6]</sup>。学科的这种交叉性给计算语言学专业课程的设置和教学带了困难和问题。德国出版的一本《计算语言学和语言技术》把目前计算语言学领域的研究分为四种路向<sup>[4]</sup>：

- 作为语言学的分支（如同社会语言学，心理语言学等），它只研究语言及语言处理与计算相关的方面，而不管其在计算机上的具体实现。语法形式化理论的研究便是其重要的研究领域之一。这类似于英语术语 **computational linguistics**（计算语言学）的涵义。
- 作为开发语言学相关程序以及语言学数据处理的学科。这一方向近年来随着大规模语料库的出现，越来越重要。
- 作为在计算机上实现自然语言能力的学科（德语区一般称为“机器的语言处理”，对应于英语的“自然语言处理”）。这一方向的研究与人工智能以及认知科学关系密切。
- 作为面向实践的、工程化的语言软件开发。也称语言技术。

以上细分较好地解决了我们对学科定义的困惑。从大的方面看，这四种研究方向都属于计算语言学，只不过侧重点不同。我们一直认为，没有理论的实践是盲目的，对于计算语言学这样的前沿领域更是如此。结合上述四种计算语言学的研究路向来看，其中的第一、三项是基础，没有这两项研究的支撑，第四项的研究不会有什么真正意义上的突破。任何急于求成，只看到短期效益的语言技术应用，只会是昙花一现。当然，我们也应该看到，在语言学的所有分支学科中，计算语言学是最具实践性的。作为文科（语言学）背景下的计算语言学，我们更感兴趣的是第一、三种方向。综合以上看法，我们给计算语言学下的工作定义是：**从多种角度研究如何通过计算机来模仿人类语言处理能力<sup>3</sup>的学科，它的终极目标是构造一个能懂人语、会说人话、可用自然语言进行交流的机器。**这个定义突出了计算语言学的两个特点：理论性和实践性，前者体现在为了模仿人的语言处理能力，我们必须对人的这种能力有所认识，而且要把这种认识上升到一定的理论层面，如果这种认识不能用一种数学的手段表述出来，那么这样的认识是无助于最终目标的实现的，这一点也体现了计算语言学与其他

<sup>3</sup> 这里的“能力”指的不仅仅是乔姆斯基所说的语言能力（competence），也包括语言应用（performance）。这个定义可视为[7]和[8]中提到的有关定义的一种简化版本。

语言学分支的不同。计算语言学中不但含有构建成分，而且应该能够解决问题，因此也可以将它称为“应用驱动”的语言学研究。计算语言学具有的建构性也导致技术现实对理论框架有一种反作用、一种限制，说起来近乎完美的理论，如果现有的技术无法实现，那么也是解决不好实际问题的。这样，我们有理由说计算语言学是一种真正的交叉学科，即它是有一套自己的理论和方法的，绝对不是简单的“计算机”+“语言学”。“这些理论和方法不是学而能的，而是需要通过学习之后才能掌握的，如果不认真学习是不容易掌握的”<sup>4</sup>。这样我们就找到了计算语言学课程的基本出发点，即：计算语言学课程的重点应该放在交叉的这一部分内容上。

## 2. 数学理论基础

我们曾经借用软件工程中的思想把计算语言学系统的构建过程分为三个模块：理论语言学（基础研究层）、计算语言学（形式化层）和计算机科学（实现层）<sup>[9]</sup>：

1. 基础研究层关心的是语言本身的特性，它的主要任务是为后面的模块提供语言学基础，它与计算机实现没有直接关系，这一阶段的主角是(理论)语言学家。
2. 与计算机关系最密切的是实现层，它的关心焦点是如何控制计算机的各种执行过程，它也负责向其他层次的人员提供高效的开发工具与环境，理论上它与具体的语言无关。实现层的主角是计算机科学家和程序员。
3. 形式化层可以说是基础研究层与实现层之间的接口，它的主要任务是将基础研究提供的普通语法模型改写为更易于计算机处理的形式化语法模型。普通语法与形式化语法的主要区别在于前者是静态的，是陈述性的，而后者是动态的，是过程性的。将语法形式化是计算语言学家的任务。

其中计算语言学家的主要任务就是语言的形式化。显然，形式化的基础离不开数学和其他面向计算的理论。也就是说，为了对语言进行形式化的描述，计算语言学家应该掌握一定的数学及其它工具。具体而言，相关的知识有：

- 集合论、命题逻辑、谓词逻辑、类型（高阶）逻辑、Lambda 演算、模态逻辑等；
- 形式语言、有限状态理论、上下文自由及相关文法，算法的可计算性和确定性等；
- 图和树、特征结构及其一般运算等；
- 概率统计的一般知识；
- 文本标注技术、HTML、XML 等。

为了学生便于理解这些数学工具在语言学中的应用，我们选用了冯志伟的《数理语言学》作为教材开设了“数理语言学”课程。而有关文本标注技术的内容则置于《信息技术》和《语料库语言学》两门课中。

## 3. 计算语言学理论基础

为了让学生了解计算语言学及中文信息处理的一般问题，我们设置了《计算语言学基础》课程，采用的教材为侯敏的《计算语言学与汉语自动分析》<sup>[10]</sup>。通过这门课程，学生基本掌握了计算语言学的历史和概况，对于中文信息处理中的汉字输入和分词等也有了一定的认识。在这门课的教学过程中，教师采用自编的小软件让学生做语言形式化分析的练习，还用已有的一些较成熟的分词和词性标注软件指导学生进行文本语料的处理，这不但培养了学生的动手能力，也让学生初步感受到了计算语言学作为一门交叉性学科的魅力，增加了进一步

<sup>4</sup> 见冯志伟为侯敏《计算语言学与汉语自动分析》一书所作的序。P. 1.

学习的兴趣。

在这些课程后，我们发现还有必要进一步从学科的交叉性入手，再开设一些能够体现计算语言学交叉性特点的课程。然而，面对如此复杂、涉及多种学科领域，该如何选择呢？我们在前面说过，计算语言学是一种面向应用、含有构建成分的语言学分支学科，它的最终目标是构造一个能够和人用自然语言进行交流的人工智能体。既然如此，就少不了要和计算机打交道，换言之，我们需要通过计算机来模仿人类的语言处理能力。而要让计算机做任何事情，我们必须要让计算机理解我们的意图，这种人机之间的交流需要用语言来完成，目前这种人机交流的语言一般是面向机器的程序设计语言。人类的梦想是有朝一日用自然语言和机器进行交流，事实上，这也正是计算语言学家的主要任务。由此，我们会很自然地会遇到这样的问题：有没有一个应该成为培养所有自然语言处理和计算语言学方向学生的得到共识的核心知识和技能呢？语言学家是否也应该懂得一些算法知识？如果回答是肯定的，那么，文科背景的语言学系的计算语言学专业所学的程序设计与计算机科学系有何不同？能不能找到一种能够为文科背景学生接受的、比较容易掌握的、又适合自然语言处理的程序设计语言呢？

我们还一直认为，计算语言学是所有语言学分支中，最需要和国际接轨的。为了培养具有国际视野的学生，我们在课程设置时应该参考国外大学有关课程的设置情况。为此，我们通过各种形式考察了国外近百所大学的计算语言学课程的内容，特别对那些设有计算语言学专业大学的相关课程进行了详细的分析。我们也参考了一个有关欧洲 60 所开设了计算语言学相关课程的大学的调查<sup>5</sup>。在 60 所大学中，其中有 33 所的计算语言学课程属文科。该调查显示，在计算语言学教学的程序设计语言方面，其中有 40 所采用了 Prolog 作为主要的教学语言。关于这一点大家很容易通过互联网来验证，一是访问有关大学的计算语言学或自然语言处理课程的网页，看看课程采用什么程序设计语言；二是登陆全球最大的书店 Amazon 检索有关计算语言学所用程序设计语言的相关书籍，其中使用较为普遍的面向自然语言处理的 Prolog 教科书有 Covington<sup>[11]</sup>，Pereira/Shieber<sup>[12]</sup>等。看来，Prolog 语言是该专业学生学习程序设计语言的首选。这个调查也显示，有 49 所大学的计算语言学课程核心内容是句法剖析算法<sup>6</sup> (Parsing algorithms)，另外一个教学的重点是形式语法 (Formal grammars)，有 45 所大学开设。这样做当然是有一定道理的，因为无论要让机器做什么，首先都得对输入的自然语言语句进行分析。这样我们又为学生开设了《Prolog 程序设计》和《自然语言的计算机处理》两门课程。选择 Prolog 的理由主要有：

- ◆ 它是最常用的 NLP/CL 教学语言之一，几乎就是计算语言学界的 Lingua Franca；
- ◆ 国际接轨和阅读国外专业文献的需要；
- ◆ 对于没有其他程序语言编程知识的人而言，Prolog 更易入门；
- ◆ Prolog 是一种“用户友好”的语言；
- ◆ 可以满足构造软件原型的需要；
- ◆ 非常适合于自然语言剖析算法的实现。

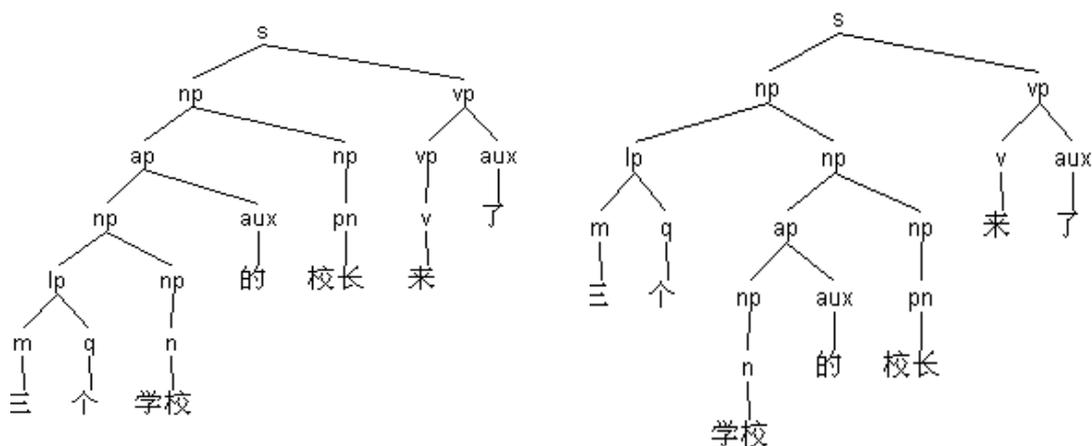
我们的 Prolog 课程教学采用由 Blackburn、Bos、Striegnitz 编写的《Learn Prolog Now!》(2001, Saarland University)，此书已被国外多所大学作为教材。通过这门课的学习，学生可以掌握 Prolog 程序设计语言的一般知识，读懂用这种语言写成的程序并编写一些简单的程序，可以利用 DCG (定子句文法) 构建简单的自然语言句法分析程序。《自然语言的计算机处理》课程采用了冯志伟的《自然语言的计算机处理》<sup>[6]</sup>一书和自编教材，

<sup>5</sup> <http://www.hit.uib.no/AcoHum/cl/cl-questresult.html>

<sup>6</sup> 关于句法剖析(parsing)的重要性，德国计算语言学家 Hans Uszkoreit 有如下名言 “If you can walk, you can dance. If you can talk, you can sing. If you can parse, you can understand.” (转引自 Martin Volk 的语料库语言学讲义)

主要内容有：有限状态自动机，有限状态剖析器和转换器，递归转移网络(RTN)，RTN 转换器和 ATN, 自底向上的句法剖析，自顶向下的剖析，左角剖析算法，良构子串表和 CYK 算法，自底向上活性线图剖析，自顶向下活性线图剖析，特征结构与合一运算，基于特征结构的剖析，基于依存语法的剖析等。这门课的特点是理论和实践相结合，在说清楚每一种计算语言学算法的同时，也用 Prolog 进行了实现。学生可以通过对这些程序的调试和运行，更深入地比较各种算法的优劣。课程的目的不仅如此，我们希望学生用自己写的汉语 CFG 语法调试学习这些算法，这培养了学生的自然语言形式化能力，而这种能力正是一个语言学背景的计算语言学人员最应该掌握的。为了方便学生调试自己的汉语形式语法，我们也向学生介绍了由丹麦哥本哈根商学院计算语言学系的 Matthias Trautner Kromann 开发的 Copenhagen Tree Tracer (以下简称 CTT)。CTT 是一个用 Prolog 开发的图形接口和一个基于线图的剖析程序，它可以方便地跟踪句子分析结果并修改剖析所用的 CFG 和 PATR 形式的自然语言语法。

事实证明，Prolog 语言确实具有方便易学、适用性强的特点，以下例句及句子树结构选自学生的课程作业<sup>7</sup>：三个学校的校长来了。



为了弥补 CFG 作为形式化体系的不足，我们又开设了《句法引论》的选修课。课程采用了 Sag、Wasow<sup>[13]</sup>的“句法”教科书，另外也介绍了其它一些面向实用的语言形式化理论，如依存和配价语法等。选修《句法引论》课程的学生，在课程结束后，可以读懂 HPSG 的有关文献并且具有使用这种理论来分析汉语的初步能力，课程也要求学生掌握依存语法的一般原理，可以用依存语法模型分析和标注真实语料。

#### 4. 结语

计算语言学是一门主要由语言学和计算机科学交叉形成的学科。随着计算机在人类生活中的作用越来越重要，计算语言学家肩上的任务也越来越重。为了使这一对人类极有意义的学科得以发展，在大学设立相应的专业是非常必要的。计算语言学经过 50 多年的历史已经形成了自己的一些理论和方法，在课程设置和教学时，我们应该突出计算语言学的这种交叉性，只有这样学科才能得到继续前进的动力。本文根据中国传媒大学应用语言学专业的语言信息处理方向课程的设置和安排，提出了一些自己的看法，希望对其他兄弟院校相关专业的课程设置有些用处。除了一般的语言学课程外，我们开设的有关课程计有：《信息技术与应用》、《数理语言学》、《计算语言学基础》、《软件工程 (Prolog)》、《自然语言的计算机处理》、《实验语音学》、《语料库语言学》、《计算语言学方法》、《机器翻译引论》、《句法引论》等。

<sup>7</sup> 选自本系 2001 级汉语言专业本科生赵怿怡的《自然语言的计算机处理》课程作业。作业要求学生写出汉语 CFG 语法，然后放入 CTT 调试，树图是 CTT 根据语法自动做出的。

应该指出的是,本文主要针对的是文科背景语言学相关专业计算语言学方向设立的,这样我们所讨论的是一种面向理论的计算语言学教学,所谓面向理论不是说不实践,而是强调我们所说的实践是一种在语言学理论指导下的实践。在本科阶段,我们没有开设专门的基于统计的自然语言处理课程,主要原因在于我们的学生文科背景的占多数,开设此类课程的效果不好。当然,随着越来越多的基于语料库的计算语言学工具的出现,我们也会考虑增加这一方面的课程。无论如何,基于语言学理论的计算语言学教学应该成为语言学系计算语言学方向的一个教学和科研重点。统计方法发展到今天,许多学者都已意识到统计和规则的结合是非常必要的,近年来连续召开的“树库和语言学理论”(Treebanks and Linguistic Theories)<sup>8</sup>研讨会充分说明了这一点。限于篇幅我们所讨论的课程主要是本科层面的,硕士和博士方面的课程设置本文没有涉及。我们也没有提及开设的有关语言学和计算机方面的一般课程,在语言学方面参照中文专业的课程设置,在计算机方面开设了数据结构、程序设计语言等。至于计算机等理工科背景的计算语言学方向的课程设置,可能会有不同,但不管怎么样,均不应忽视计算语言学的这种交叉性,因为这是学科的生命线。

### 参考文献

- [1] John Hutchins (2000, ed.) *Early Years in Machine Translation*. Amsterdam: John Benjamins Publishing Company.
- [2] Robert Dale, Hermann Moisl, Harold Somers (eds. 2000) *Handbook of Natural Language Processing*. Dekker.
- [3] Ruslan Mitkov (ed. 2003) *The Oxford Handbook of Computational Linguistics*. Oxford University Press.
- [4] Carstensen, Kai-Uwe; Ebert, Christian; Endriss, Cornelia; Jekat, Susanne; Klabunde, Ralf; Langer, Hagen (2004, eds.) *Computerlinguistik und Sprachtechnologie. Eine Einführung*. 第二版, Heidelberg/Berlin: Spektrum Akademischer Verlag.
- [5] Daniel Jurafsky, James H. Martin (2000) *Speech and Language Processing. An introduction to natural language processing, computational linguistics, and speech recognition*. New Jersey: Prentice Hall.
- [6] 冯志伟 (1996) 自然语言的计算机处理, 上海外语教育出版社.
- [7] Roland Hausser (2001) *Foundations of Computational Linguistics: Man-Computer Communication in Natural Language*. The second edition. Berlin-New York: Springer-Verlag.
- [8] Bill Manaris (1998)“Natural Language Processing: A Human-Computer Interaction Perspective,” In *Advances in Computers* (Marvin V. Zelkowitz, ed.), vol. 47, pp. 1-66, Academic Press, New York, 1998.
- [9] 刘海涛 (1995) 计算语言学应用中的模块化概念,《语言文字应用》,第4期。也载易绵竹、南振兴编《计算语言学》,上海外语教育出版社,2005。 p. 132-142.
- [10] 侯敏 (1999) 计算语言学和汉语自动分析, 北京广播学院出版社.
- [11] Michael A Covington. *Natural Language Processing for Prolog Programmers*. Prentice Hall, Englewood Cliffs, NJ, 1994.
- [12] Fernando C. N. Pereira and Stuart M. Shieber (2002) *Prolog and Natural-Language Analysis*. Digital edition. Brookline/Massachusetts: Microtome Publishing.
- [13] Ivan A. Sag, Thomas Wasow (1999) *Syntactic Theory: A Formal Introduction*. Stanford, CA: CSLI Publications.

---

<sup>8</sup> 首次会议于2002年9月在保加利亚 Sozopol 举行,第二次会议于2003年11月在瑞典的 Växjö 举行,第三次于2004年12月在德国的 Tübingen 举行。