新时代的计算语言学（代序）

Grishman（1986：4）将“计算语言学”（computational linguistics）定义为“一门研究如何利用计算机来理解和生成自然语言的科学”。这指明了计算语言学的研究目标和研究手段。理解和生成自然语言，是计算语言学的研究目标；利用计算机，是计算语言学的研究手段。更确切地说，是“利用计算机建立传输说话者所表述和听话者所理解的信息的计算模型”（Hausser，2014：xix）。Allen（1995：3）则认为计算语言学的目标应该表述为：“利用计算机科学的算法和数据结构来建立语言的计算理论。”

要实现语言的生成，首先得要实现计算机对自然语言的理解。有人认为，现阶段提出理解目标不切实际，因为目前占主流地位的统计方法和深度学习的方法所达到的目标只是处理，还谈不上理解。更进一步说，并非经过理解才能处理。但是，统计方法只是解决问题的方法之一，它不能处理所有的语言问题；深度学习也不能真正理解语言，计算机所理解的人类语言不过是一种模仿或者复述。只有真正理解了人类语言，才能实现语言的生成。

要使计算机理解自然语言，必须使之具备以下自然语言知识（Allen，1995）：

- 语音和音系学知识：主要关注语音怎样转化为词；
- 形态学知识：主要关注词素怎样构成词；
- 句法知识：主要关注词怎样构成句子；
- 语义知识：主要关注词义怎样构成句义；
- 语用知识：主要关注句子在不同语境中的使用；
- 语篇知识：主要关注上下句之间的关系；
- 世界知识：主要指说话者和听话者所具备的对外部世界的认知。

通常来说，计算机要具备的自然语言知识似乎和传统语言学和现代语言学的内容大致相当。传统语言学着重语言事实的描写，经验性质比较突出。现代语言学，例如乔姆斯基语言学（Chomskyan linguistics），

理论性非常强，已经脱离了经验科学的范畴，我们称之为“理论语言学”。但计算语言学和这两者是有本质区别的。

理论语言学和计算语言学都是研究自然语言的，但服务对象有所不同：前者是面向人的，后者是面向计算机的。计算语言学是一门实验科学，所以它提出的问题既要符合自然语言处理的实际需要，又要用现有的计算机技术解决。超出计算机的能力，就不具有可行性。此外，计算语言学中研究对象的定义必须明确，不能含糊。例如汉语“词”的定义，理论语言学上的定义是：词是最小的、能独立运用的语言单位，但这一定义并不清晰。语言学家也分析了词的一些特征，例如“结合紧密、使用稳定”等，但没有定量标准，这样的定义对计算机来说是无益的。计算语言学中“词”的定义，简言之，能在分词词表中找到的就是词，否则就不是词，或者是未登录词。这样，计算机就在词表中查找，能找到的就是词，找不到的就划归到未登录词里做下一步处理。

理论语言学研究主要不是考虑计算机的应用，因此无法提出自然语言处理的问题和理论。例如，汉语自动分词（Chinese word segmentqtion）问题就是从中文信息处理角度提出来的，汉语理论语言学研究从来没有、也不可能提出这样的问题[1]。此外，理论语言学不一定要形式化，也没有为形式化提供任何手段。形式化是数学表示的问题，包括两个方面：一是问题本身的形式化描述；二是解决问题的方法的形式化描述，后者通常用数学模型来体现。要让计算机掌握和具备以上的语言知识，计算语言学研究者首先得将这些知识形式化，并将其用算法的形式在计算机上加以实现。

从宏观上看，计算语言学的基本方法有两种：基于规则的方法和基于经验的方法。前者的理论基础是语言学上的理性主义（rationalism），以乔姆斯基理论为代表。乔姆斯基（Chomsky，1986：5）认为人的语言知识“通过某种方式表现在我们的心智之中，最终表现在我们的大脑之中，这种知识的结构我们希望能够抽象地描写出来，用具体的原则、根据物质机制描写出来”。语言学研究的目标是人类的这种语言能力，

1 “词式书写”在拼音文本或者拼音和汉字对照文本中早就开始尝试，已经普遍使用，但是在汉字文本的书面语中还在尝试，如中南大学出版社出版的《语言理论》（彭泽润、李葆嘉主编）。彭泽润认为，他坚持词式书写汉语是为了让汉语和世界接轨，并没有太多理论语言学或者语言研究上的考虑。

言语是语言能力的具体表现，不是语言学应该关注的重点。理性主义方法的特点是演绎法，从原则和参数演绎出规则，从规则推导出具体的句子。乔姆斯基语言学虽然不属于计算语言学，但对于计算语言学的形成和发展有重大影响。基于规则的计算语言学研究方法中的理性主义体现在两个方面：第一，目标定位于“自然语言理解”，希望在理解的基础上来处理自然语言；第二，方法的核心是“基于规则”，希望根据通过内省和演绎得到的一整套规则来处理自然语言。

而基于经验的方法的理论基础是经验主义（empiricism），来源于香农的信息论。信息论认为语言事件（语言表现）是有概率的，可以通过统计得到这些概率，从而对自然语言处理（natural language processing，NLP）的各种具体问题进行决策。经验主义方法的特点是归纳法，集中体现为语料库语言学。与理性主义相对立，经验主义认为，完成自然语言处理任务不一定要经过理解的阶段，通过内省和演绎得到的规则往往是颗粒度较大的语言知识，只有通过运用统计方法，才能自动获得大量的、带概率的小颗粒度语言知识，从而处理大规模真实文本。

冯志伟（2005）将计算机对语言的研究和处理划分为以下四个阶段：

（1）把需要研究的问题在语言学上加以形式化，建立语言的形式化模型，使之能以一定的数学形式，严密而规整地表示出来；

（2）把这种严密而规整的数学形式表示为算法，使之在计算上形式化；

（3）根据算法编写计算机程序，使之在计算机上加以实现，建立各种实用的自然语言处理系统；

（4）对于建立的自然语言处理系统进行评测，使之不断地改进质量和性能，以满足用户的要求。

自然语言处理的这四个阶段可以简单概括为：数学模型→算法表示→程序实现→质量评测。计算语言学会涉及上述四个阶段的哪个阶段或者哪些阶段，目前学界和业界对此认识并不明晰，计算语言学和自然语言处理的学科分界尚不明确。

一般情况下，学界对于计算语言学和自然语言处理这两个术语是不加区分的。因为两者的本质是基本相同的，区别可能仅仅在于自然语言处理更注重实践，而计算语言学较重视理论。在《牛津计算语言学手册》

（*The Oxford Handbook of Computational Linguistics*）第一版（2003）中尚能看到两者的明显区别：全书分为三部分——第一部分“基础篇”、第二部分“方法与资源篇”、第三部分“应用篇”，自然语言处理的内容大都被放在第三部分“应用篇”；然而在其第二版（2014）中已然很难发现两者的界限了：全书分为四部分——第一部分“语言学基础篇”、第二部分“计算基础篇”、第三部分“语言处理任务”和第四部分“自然语言处理应用”，其中第三部分和第四部分占据全书的大部分体量，但对计算语言学和自然语言处理并未做区分。然而，Roland Hausser 在其全三版《计算语言学基础》（*Foundations of Computational Linguistics*）教材中坚持语言的可计算性和形式语言学，从未涉及任何具体的语言处理任务，以示计算语言学“坚壁清野”，不与自然语言处理发生任何学科交叉。

我们相信，在学科交叉和学科融合的大背景下，坚持一个学科的纯洁性既无必要也不现实，但一个学科有一个学科本身的发展规律和学科特点，丢掉特点去迎合热点是一件极其危险的事情。尽管两者的本质是基本相同的，但总体而言自然语言处理更注重实践，而计算语言学较重视理论。也可以说，计算语言学是建构自然语言处理系统的理论基础（刘海涛，2001），两者还是应该各自有所侧重的。Manaris（1998：1）认为自然语言处理可以定义为“研究在人与人交际中以及在人与计算机交际中的语言问题的一门学科，即研究表示语言能力和语言应用的模型，建立计算框架来实现这样的语言模型，提出相应的方法不断地加以完善，根据模型设计各种实用系统，并探讨这些实用系统的评测技术”。

也有学者认为，自然语言处理就是计算语言学的应用领域。随着计算机速度的加快和存储量的增加，计算语言学在语音合成、语音识别、文字识别、拼写检查、语法检查应用领域进行了商品化开发。除了早期开始的机器翻译和信息检索等应用研究进一步得到发展之外，计算语言学在信息抽取、问答系统、自动文摘、术语的自动抽取和标引、文本数据挖掘、自然语言接口，计算机辅助语言教学（computer-assisted language learning）等新兴的应用研究中，都有了长足的进展。此外，计算语言学的技术在多媒体系统（multimedia system）和多模态系统

（multimodal system）中也得到了应用。

汉字识别的核心技术是字形特征的抽取和模式识别，识别结果是否能组织为有意义的文本，取决于自然语言理解。语音识别和语音合成则需要用到文语转换技术，即从文本到标音符号的相互转换，其中多音字的处理是关键。自动校对可大大减轻人工校对工作量，使这一环节跟出版业的其他环节的自动化相适应。计算机辅助语言教学属于现代教育技术，如果没有自然语言处理技术的支持，电子教案可以说是纸质教案的翻版。好的教学软件应该包括更多的人机交互活动，例如习题的自动生成、作业的自动批改。机器翻译的意义毋庸赘言，这是一种综合性最强的应用。仅就文本形式的翻译而言，就需要用到知识表示方法、机译词典构造、源语言的分析、目标语言的生成等技术。如果是口语现场翻译，还需要有语音识别、语音合成以及人机接口技术的配合。智能检索，包括信息检索、信息抽取、文本挖掘、话题跟踪、文本分类、文本过滤、问答系统等，是当前最热门的应用。文本分类是智能检索的一个重要方面，对于网站新闻频道的自动更新具有特殊意义。例如，中国搜索在线报告，他们的新闻频道就是使用文本分类技术而自动更新的，其他网站的最新消息可在两分钟内在他们的频道得到反映。自动文摘可帮助人们快速、准确、全面地获取信息，特别是因特网上的信息。简单的原文浓缩，就能起到一定的作用。哪些句子最能代表原文内容，需要根据其出现位置、所含词语进行计算。如果要用不同于原文的句子来表示，还需要用到语句分析和语句生成技术。

但计算语言学的研究内容和其主要应用不是一一对应的，后者应符合市场需要。有些基础研究本来就不是瞄准直接应用的，例如句法分析技术可在多种应用系统中起作用，但不可能独立成为一种社会大众需要的应用。也很难讲在上述应用场景中，计算语言学到底在自然语言处理任务的哪个（哪些）环节作出了贡献。

一般认为计算语言学是语言学的分支，自然语言处理是计算机科学的子学科。但是现在由于计算语言学和自然语言处理之间的界限越来越模糊，甚至两个领域的学者常常去参加同样的会议，交流各自的研究工作也完全没有障碍，于是就有了一个说法：计算语言学和自然语言处理都是跨语言学和计算机科学的交叉学科。然而，Ryan Cotterell 博士和

Emily M. Bender 教授几年前在推特上发起的一场有关“自然语言处理是不是交叉学科”的争论[1]却将此引入纵深，同时也引发了对计算语言学学科属性的深层次讨论。

Ryan Cotterell 认为自然语言处理的研究成果并没有吸收语言学方面最新的进展，因此不被公认为是跨学科的。他更加坦率地认为语言学和自然语言处理已经分离开了，甚至表示自然语言处理在过去 10 到 20 年的发展与近期语言学的研究无关，理由是他认为交叉学科必须建立在两个学科共同的工作基础上，而目前自然语言处理的工作大部分不符合这个定义。Emily Bender 则认为如果问题要求多个领域的专业知识有效地接近，一个研究领域原则上就是跨学科的。据此定义，自然语言处理原则上就是跨学科的。但她同时又同意 Ryan 的观点，说自然语言处理在实践中大多不是学科交叉的，同样也不认为语言学的所有子领域都和自然语言处理相关。因此她的观点是：学习语言如何工作以及（或者）与有相关经验的人合作，会让自然语言处理发展得更好。无独有偶，现代语音识别和自然语言处理研究的先驱 Frederick Jelinek 曾经说过：“每当我开除一个语言学家，语音识别系统就更准确了。”后来他又改口说“我的一些最好的朋友是语言学家”（曾江，2020）。由此看来，为了从事计算语言学和自然语言处理的研究，语言学家很有必要更新知识，很有必要学习数学和计算机科学的知识。

如果把计算语言学或自然语言处理领域分为两大派别，即计算机主义者和语言学主义者，随着人工智能、机器学习在自然语言处理领域影响力不断增大，计算机主义者逐渐演化成人工智能 / 机器学习主义者，而 Bender 教授则是坚定的语言学主义者。她认为自然语言处理领域越来越看重神经模型、人工智能算法而忽视传统、忽视语言本体，而且对模型和算法的过分赞誉和夸大宣传主要是由于对语言形式和语言意义的误解造成的，一个直接的理由就是“语言模型无法学习语义，因为语言模型仅仅使用语言形式作为训练数据，并没有碰触到语言意义本身”（Bender & Koller，2020：5185）。

相对于自然语言处理工程问题，计算语言学主要致力于用计算的方

1 详见“AI 科技评论”的原创微信推文“一条 Twitter 引发的学术争论：NLP 是交叉学科吗？”（2017–11–11）。

法来回答语言学的科学问题。语言学的核心问题包括语言表征和语言知识的性质，如何在语言的产生、理解中获得和运用语言学知识。对这类问题的回答，有助于描述人类的语言能力，也有助于解释实际记录的语言数据和行为的分布。在计算语言学中，我们用更形式化的答案来回答这些问题。语言学家关心人类计算了什么以及是如何计算的，所以我们将语言表征和语法通过数学的形式来定义，研究它们的数学属性，并设计有效的算法来学习、生成和理解。只要这些算法可以实际运行，就可以测试我们的模型，看它们是否能作出合理的预测。

语言学也考虑一些“非核心”的语言问题，例如社会语言学、历史语言学、生理语言学或者神经语言学等。这些学科问题本质上和计算语言学是平等的，都是在用一套模型和算法让语言数据看起来合理。从这个角度来说，计算语言学并不试图去对日常用语进行建模，而是将语言学家所作的推论自动化。这潜在地就使我们能够处理更大的数据集（甚至新的数据）并得出更准确的结论。同样的，计算语言学家可能会设计软件工具来帮助记录濒危语言。那么，很明显计算语言学具有跨学科的性质。

以机器翻译为例，计算语言学致力于机器翻译的主要目标是解释和探究翻译的本质以及翻译活动的过程，但自然语言处理工程师则不会考虑机器翻译有没有解释翻译的本质是什么或者翻译人员是如何工作的，他们在意的是机器翻译系统能否产生一个合理、精确、流畅的翻译结果。机器翻译也有自己的衡量方法用以评价和提高这些机器翻译质量，而不是理解翻译的本质。因此套用人工翻译的评价标准和体系（如“信、达、雅”）去衡量机器翻译的译文质量的做法本身就不可取，也不可信。

从学科属性上来说，计算语言学到目前为止，理论体系尚未建立，还不能算是一门理论科学。一方面，其主流方法（统计方法和神经网络方法）是经验主义的，这充分表明计算语言学还是一门经验科学。另一方面，计算语言学又的确是一门实验科学，其理论和方法的正确性都需要通过在计算机上做实验来得到证明。而理论语言学则不是一门实验科学，有些问题本质上无法通过实验来研究，例如语言的发展规律。

近年来，随着人工智能的崛起，自然语言处理也走向了智能化，出现了计算语言学的另外三种主义——符号主义（symbolicism）、连接主义（connectionism）和行为主义（actionism）。符号主义又称为逻辑主义、心理学派或计算机学派，原理主要为物理符号系统（即符号操作系统）假设和有限合理性原理。连接主义又称为仿生学派或生理学派，主要原理为神经网络及神经网络间的连接机制与学习算法。行为主义又称为进化主义或控制论学派，原理为控制论及感知－动作型控制系统。

符号主义认为人工智能源于数理逻辑。数理逻辑从 19 世纪末起得以迅速发展，到 20 世纪 30 年代开始用于描述智能行为。计算机出现后，又在计算机上实现了逻辑演绎系统。其有代表性的成果为启发式程序 LT 逻辑理论家，它证明了 38 条数学定理，表明了可以应用计算机研究人的思维过程，模拟人类智能活动。正是这些符号主义者，早在 1956 年首先采用“人工智能”这个术语，后来又发展了启发式算法、专家系统、知识工程理论与技术等，并在 20 世纪 80 年代取得很大发展。符号主义曾长期一枝独秀，为人工智能的发展作出重要贡献，尤其是专家系统的成功开发与应用，对人工智能走向工程应用和实现理论联系实际具有特别重要的意义。在人工智能的其他学派出现之后，符号主义仍然是人工智能的主流派别。这个学派的代表人物有 Newell、Simon、Nilsson 等。

连接主义认为人工智能源于仿生学，特别是对人脑模型的研究。它的代表性成果是 1943 年由生理学家 McCulloch 和数理逻辑学家 Pitts 创立的脑模型，即 MP 模型，开创了用电子装置模仿人脑结构和功能的新途径。它从神经元开始研究神经网络模型和脑模型，开辟了人工智能的又一发展道路。20 世纪 60~70 年代，连接主义，尤其是对以感知机为代表的脑模型的研究出现过热潮，由于受到当时的理论模型、生物原型和技术条件的限制，脑模型研究在 20 世纪 70 年代后期至 80 年代初期落入低潮。直到 Hopfield 教授在 1982 年和 1984 年发表两篇重要论文，提出用硬件模拟神经网络以后，连接主义才又重新抬头。1986 年，Rumelhart et al.（1986）提出多层网络中的反向传播（back propogation，BP）算法。此后，连接主义势头大振，从模型到算法，

从理论分析到工程实现，为神经网络计算机走向市场打下基础。现在，对人工神经网络（Artificial Neural Network，ANN）的研究热情仍然较高，但研究成果未达预期。

行为主义认为人工智能源于控制论。控制论思想早在 20 世纪 40～50 年代就成为时代思潮的重要部分，影响了早期的人工智能工作者。Wiener et al.（1948）提出的控制论和自组织系统以及钱学森等人提出的工程控制论和生物控制论，影响了许多领域。控制论把神经系统的工作原理与信息理论、控制理论、逻辑以及计算机联系起来。早期的研究工作重点是模拟人在控制过程中的智能行为和作用，如对自寻优、自适应、自镇定、自组织和自学习等控制论系统的研究，并进行“控制论动物”的研制。到 20 世纪 60 年代，上述控制论系统的研究取得一定进展，播下了智能控制和智能机器人的种子，并在 20 世纪 80 年代诞生了智能控制和智能机器人系统。行为主义是 20 世纪末才以人工智能新学派的面孔出现的，引起许多人的兴趣。这一学派的代表作者首推 Brooks 的六足行走机器人，它被看作新一代的“控制论动物”，是基于感知 – 动作模式模拟昆虫行为的控制系统。

近来学界对自然语言处理领域发展的反思和态度转变可以总结为两种理论构建视角，即自底向上（bottom-up）和自顶向下（top-down）的理论构建。在自底向上的视角下，学术界研究是通过发现和解决具体的研究挑战驱动的，如果科学研究能完全解决一个具体挑战，或者部分解决，那就可以被视作一项学术成果，只要这些让人满意的成果是频繁出现且不断攀升的，就会带来一种持续进步的总体氛围。与之相对的自顶向下视角则聚焦远期终极目标，为整个领域提供一套完整统一的理论体系。自顶向下的视角会带来焦虑感，因为我们还不能完全解释所有现象，还会出现更加棘手的问题，那就是自底向上的进步到底有没有把我们领向正确的方向。同样的任务从自底向上的视角看是自然语言处理问题，而从自顶向下的视角看就成了计算语言学的问题。毫无疑问，自然语言处理正以飞速攀登的速度进步，每年各领域自然语言处理任务的解决办法都通过更好预训练的语言模型得到显著改进，都能达到目前最好的水平（state-of-the-art，SOTA）。但是，如果从自顶向下的角度看，我们如此飞速攀登的山峰，究竟是不是“正确的”山呢？不知道当

今飞速进步会把我们带向什么样的最终目标，是普遍语言智能（general linguistic intelligence），还是一个可以通过图灵测试（Turing test）的系统？

但计算语言学与自然语言处理的学科属性的争论和各自研究重点的区别仍然悬而未决，或许无法解决。不管争论的结果是什么，都是有益的，因为讨论会促使人们反复思考自己的观点。因此，大部分学者对“计算语言学”和“自然语言处理”这两个术语的使用只是遵循各自的使用习惯而未作细致区分，甚至有时两者是混用的。如果非要给两者加以界定，那可能计算语言学更“理论”，而自然语言处理更偏向“应用”。

纵观计算语言学发展史，计算语言学家经历了多次主流变革。基于语法规则和专家知识的方法让位于统计方法，如今大部分研究又吸收了神经网络和深度学习方法。每一代研究者都觉得他们解决了相关问题并且不断进步，但是当每种范式出现不可解决的致命缺陷，该范式随即就会被抛弃。那么，应该如何尽量让计算语言学的科研攀登是在一座正确的山上呢？ Bender & Koller 在论文中提出了五种“爬山攻略”（hillclimbing diagnostics）：

第一，对语言问题保持谦卑与敬畏，多问一些自顶向下的问题。神经网络并不是自然语言处理领域第一个取得成功的方法，应该也不会是最后一个。

第二，了解自然语言处理下游任务的局限性。比如 CAMRP 这样的人工赛道任务（见第 6 章）可以帮助某一个领域的研究尽早取得突破，但是不要妄想测试数据的语言分布能完全模拟现实语言世界的整体分布。

第三，重视和支持新赛道，但要慎重选择和创建新任务。比如，在第十三届语言资源与评测国际会议（LREC 2020）上举行的第一届古代汉语分词和词性标注国际评测（EvaHan）就大力推动了古汉语信息处理和古籍数字人文研究（见第 2 章）。

第四，要通过多任务来评价语义模型。比如，面向通用目标的自然语言理解系统评价 SuperGLUE（Wang et al.，2019）就是通过多个任务来评价一个系统的语义理解模型，而不是让系统只完成某个单项任务来进行评价。