

文章编号:1003-0077(2004)06-0053-08

语言工程的软件体系结构研究综述

冯 冲^{1,2}, 陈肇雄¹, 黄河燕¹

(1. 中国科学院 计算机语言信息工程研究中心, 北京 100083;

2. 中国科学技术大学 计算机系, 安徽 合肥 230027)

摘要:语言工程的软件体系结构已经逐渐发展成为语言工程的主要研究领域之一。它面向通用的自然语言应用,为其提供架构层次的参考方案。研究内容涵盖与体系结构相关的计算资源、语言资源、方法和应用等多个方面。在一定意义上,可以把它看作是在语言工程领域内的特定领域软件体系结构(DSSA)。本文概要介绍了该领域的发展历程和研究意义,然后对其基本概念和当前主要研究进展进行了阐述和分析,并展望了进一步的发展趋势。

关键词:人工智能;自然语言处理;综述;语言工程;软件体系结构;自然语言处理

中图分类号:TP391.1 **文献标识码:**A

Research on Software Architecture for Language Engineering: A Survey

FENG Chong^{1,2}, CHEN Zhao-xiong¹, HUANG He-yan¹

(1. Research Center of Computer & Language Information Engineering CAS, Beijing 100083, China;

2. University of Science and Technology of China, Hefei, Anhui 230027, China)

Abstract: Providing reference architectures for general natural language applications, software architecture for language engineering has gradually become one of the main research fields of language engineering in the past several years. This paper makes a short review on this fresh area, introduces its primary concepts, and discusses some representative progresses. Based on the analysis to the current work, we present some promising direction for future research.

Key words: artificial intelligence; natural language processing; overview; language engineering, software architecture, natural language processing

1 引言

1.1 发展概况

人们开发自然语言处理技术的应用系统已有较长的历史了,例如,第一个重要的商用机器翻译系统 Systran 是在 20 世纪 60 年代出现的。从语言处理的产品化系统的角度看,语言工程领域的开发活动已取得相当丰富的经验和教训;但是,作为有明确的实践内容的工程学科,语言工程还是最近的事情。语言工程这一术语最早可以追溯到 Mitkov 在 COLING88 会议上的论文^[1];到了 90 年代,随着欧洲委员会资助的一个同名项目,这一术语在欧洲逐渐被广泛接受。同时,这方面的学术研究的主体也开始出现,研究内容不断得到充实。专注于该领域的学术期

收稿日期:2004-03-09

基金项目:国家自然科学基金资助项目(60272088);国家 863 项目资助项目(2002AA11401)

作者简介:冯冲(1977—),男,博士研究生,主要研究方向为自然语言处理和机器翻译。

刊《Journal of Natural Language Engineering》于 1995 年创刊。

随着研究人员越来越多的从工程角度看待自然语言处理中的健壮性、实用性等方面的问题,语言工程的软件体系结构(Software Architecture for Language Engineering)作为语言工程的主要领域之一逐渐发展起来,并正在受到越来越多的研究人员的关注^[24]。在 2003 年北美计算语言学协会的 HLT-NAACL 会议中,举办了讨论语言工程的软件体系结构的 SEALTS 会议(Workshop on Software Engineering and Architecture of Language Technology Systems),并策划成立语言工程的软件体系结构特别兴趣组(Special Interest Group)。此方面的研究内容和成果也在不断充实,表 1 列出了本领域的一些有代表性的成功研究项目。《Journal of Natural Language Engineering》还在 2003 年出版了研究语言工程的软件体系结构的专刊。

表 1 部分有代表性的语言工程的软件体系结构研究项目

项目名称	研究机构
RACS, Reference Architecture for Generation Systems	Brighton and Edinburgh
ATLAS	LDC, NIST
Galaxy Communicator Software Infrastructure	MIT & MITRE
GATE, a General Architecture for Text Engineering	Sheffield
Protégé	Stanford
TEI, CES, XCES	Oxford, Vassar, etc.

在本文接下来的部分中,首先概要阐述了本领域研究意义,然后,对语言工程的软件体系结构这一研究领域中的基本概念和研究状况进行了总结和评述,并介绍了部分有代表性的研究项目,最后,我们提出了对未来发展的一些认识。

1.2 研究意义

语言工程的软件体系结构对于自然语言处理研究具有重要意义。

正如文献[2]所指出的,“当所研究的系统变得越来越大时,软件工程和人工智能对于程序设计的考虑会逐渐走向融合”。语言工程的软件体系结构研究正反映了这种趋势。语言工程要想真正成为名副其实的“工程”,就决不可忽视自然语言处理中的体系结构设计、分析和评估问题。众所周知,在一般的软件系统开发中,按部就班地完成开发任务是相当困难的,约有 20 % 以上的项目以失败告终^[3]。对于更为复杂的自然语言处理方面的应用程序,这个比例可能会更高些。语言工程的软件体系结构的研究价值和实践价值主要体现在如下几个方面:

® 灵活性:和其它系统一样,NLP 系统应当是灵活的:一个最初为分析互联网新闻而开发的语法分析器也应当能够被移植到商业电子邮件的处理中。系统应当能够处理不同的数据格式,并且这些表示层和输入输出层上的知识应当与核心的语言学知识分离。

® 健壮性:在工程领域,健壮性一般指设备在不同的工作条件下的工作能力。在语言工程中,健壮性还要更狭义一些,一般指某种语言学方法对不同类型的文本和不同应用领域的适应性。在语言工程中,较低的健壮性通常并不会象在其他工程领域那样可能导致整个系统的失效,而是带来准确率、召回率等性能指标的下降。

® 准确率:与其他传统软件相比,NLP 系统的一个基本差异是它的不完备性——当前的语言处理技术既难以保证 100 % 的准确率,也难以保证 100 % 的召回率,因此,整个系统的设计都要考虑到这些并提供适当的应变手段。

® 生产率:对现有资源缺乏了解,对组织外的构件缺乏信任,都会使研究人员难以提高生产率。同时,由于时间紧迫,研究人员通常只开发原型系统并用它来演示一种方法的可行性,而把效率问题和系统的完整实现留给工程技术人员。但工程技术人员通常不会进行从原型系

统到产品系统的重新开发。语言工程的软件体系结构方面的一些研究结果表明,软件工程的引入能在低投入的情况下带来整体生产率的提高^[4]。

1.3 主要研究方向

语言工程的软件体系结构主要研究以下方面内容:语言技术系统(Language Technology Systems)的体系结构,语言资源的建设和管理,语言技术系统的数据共享,语言技术系统的项目管理,用于不同目的的语言技术系统工程,语言技术系统之间的知识传输和代码共享标准,不同体系结构及其实现的对比试验,等等。在更广泛的意义上,它还讨论知识存储、消息传递、数据模型等软件体系结构研究中的共同问题。

2 语言工程的软件体系结构研究现状

2.1 主要概念

2.1.1 语言工程(LE, Language Engineering)

在语言工程的概念上,我们认同文献[5]中的相关阐述。语言工程的概念与计算语言学、自然语言处理等概念有着不同的侧重点。计算语言学侧重于语言科学,而且使用计算机作为研究工具。如果说“理论语言学家努力研究某种语言或某种语法的特点”,那么“计算语言学家则是努力用计算的方式研究这些特点”^[6]。自然语言处理侧重于计算机科学,而且以处理人类语言的计算机系统为研究对象^[7]。语言工程则是自然语言处理的应用,或者说,“把语言处理作为一种工具来使用,比如作为一个有实用目标的、更大规模的系统的一部分”^[6]。

这一观点也得到了多数研究人员的支持。《Journal of Natural Language Engineering》的创刊号在社论[8]阐述了语言工程与自然语言处理、计算语言学的关键区别:“自然语言工程的基本特征在于它面向的是处理自然语言的工程化的产品,并要满足为达到此目标所需的各种限制条件。这一点在其他的工程领域(如机械工程)可能是显而易见的,但对于软件工程领域的实践者来说则是仍需强调的,而对于自然语言处理领域的实践者来说则近乎是革命性的。”

2.1.2 软件体系结构(Software Architecture)

软件体系结构是软件工程研究中新出现的一个学科分支。尽管它发展迅速并正日益走向成熟,但类似于许多计算机技术,人们对软件体系结构的定义尚未形成统一认识。较为广泛使用的定义有^[9]:

® Bass 等人认为,程序或计算系统的软件体系结构是系统的一个或多个结构(structure),包括软件构件(components)、构件的外部可视属性(properties)和构件之间的关系(relationships)。

® Shaw 等人认为软件体系结构由构件、连接件和约束构成。其中,构件(component)可以是一组代码,也可以是一个独立的程序。构件是相关对象的集合,运行后实现某计算逻辑。它们相对独立,或是结构相关或是逻辑相关,仅通过接口与外部相互作用。

® David Garlan 和 Dewne Perry 采用如下定义:软件体系结构是一个程序/系统各构件的结构、它们的相互关系以及进行设计的原则和指导方针,这些原则和方针随时间演化。

在语言工程的软件体系结构领域,人们通常采用 Shaw 和 Clements 在文献[10]中给出的定义:软件体系结构解决系统结构方面的问题——软件的组织,构件间责任的分配,并确保构件的交互满足系统需求。我们认为这一定义较好的总结了软件体系结构研究者们的共识,并能反映其在语言工程中的作用。

2.1.3 语言工程的软件体系结构(SALE, Software Architecture for Language Engineering)

可以把 SALE 看作是在语言工程领域的特定领域的软件体系结构(DSSA, Domain Specific

Software Architecture)。根据我们对语言工程的软件体系结构的理解,语言工程的软件体系结构就是研究语言工程领域中的标准软件体系结构,为各种自然语言应用提供组织结构的参考方案。所谓 DSSA,即“专用于一个特定类型的任务(领域)的、在整个领域中能有效使用的、为成功构造应用系统限定了标准的组合结构的软件构件的集合”^[11]。DSSA 的另一种常用定义是:“DSSA 就是在一个特定的问题领域中支持一组应用的领域模型、参考需求、参考体系结构等组成的开发基础,其目标就是支持在一个特定领域中多个应用的生成”^[12]。DSSA 最早是由美国 DARPA 倡导的,该部门发起的 DARPA - DSSA 计划是 DSSA 研究的重要发展阶段,由美国军方、工业界、学术界共同参与,开发了许多软件开发领域的参考体系结构和设计分析工具,如:航空电子、指挥监控、CNC(引导、导航和控制)、适应性智能系统等,极大的推动了 DSSA 研究。我们认为语言工程的软件体系结构和其他 DSSA 一样,具有如下特征:

- ® 是对整个语言工程领域适度的抽象;
- ® 具有严格定义的问题域和解决方案域;
- ® 具备语言工程领域固有的、典型的在开发过程中可重用元素;
- ® 具有普遍性,即可用于语言工程领域中某个特定应用的开发。

2.2 当前研究状况

在文献调研的基础上,我们把语言工程的软件体系结构方面的研究工作按照处理对象不同大致归纳为如下 3 类:

1) 计算资源

语言处理系统中的计算资源通常由几个离散的步骤构成。例如,一个机器翻译系统必须首先分析源语言文本,得到其内部表示,然后才能确定目标语言文本的结构。而一个典型的分析过程又包括多个步骤:文本结构分析,词法分析,语法分析,语义分析。其中的每一个步骤都可以描述为一个处理文本的构件。换言之,分析步骤可以实现为一组计算资源的集合。例如,一些信息抽取系统,如文献[13,14]等,采用的就是这种 pipeline 风格的体系结构。还有很多研究也强调了语言工程构件和主控执行程序的分离,如文献[15~17]等。计算资源方面的研究主要处理如下问题:

- (a) 从本地和非本地机器上定位,载入和初始化构件;
- (b) 串行或并行的运行完成处理任务的构件;
- (c) 描述和构件有关的信息;
- (d) 抽取出构件之间的共性。

2) 语言资源,语料库和标注

语言资源主要是指诸如词典、语料库和语言等方面的数据构件。它们构成了语言工程的基本素材。LT XML, TEI, CES, XCES 等是近年来的一些较有代表性的研究项目。语言资源、语料库和标注方面的研究主要处理如下问题:

- (a) 访问数据构件;
- (b) 管理文档集合及其格式;
- (c) 描述文本和语音信息;
- (d) 描述语言有关的信息;
- (e) 索引和检索信息。

3) 方法和应用

为了解决特定的语言工程任务,人们已经开发了大量工具包,它们可以起到基础框架的作

用。例如,CMU-Cambridge 统计模型工具包^[18]就是一个能够构建 n-gram 语言模型的命令行工具集合。类似的工具包还有用来构建语音识别系统的 HTK^[19]等。同时,在运用语言工程技术和语言工程构件来完成实用系统时,还常遇到一些应用问题,例如怎样部署应用程序,怎样嵌入构件,等等。方法和应用方面的问题主要包括:

- (a)方法支持;
- (b)应用问题;
- (c)开发问题。

2.3 主要研究项目

在这一领域,人们已经完成了大量的研究和开发。下面我们介绍几个典型的 SALE 研究项目。

1) RACS (Reference Architecture for Generation Systems)

RACS 项目开始于 1998 年 3 月,受英国工程与物理科学研究委员会资助,是由英国 Brighton 大学信息技术研究中心和 Edinburgh 大学信息学分部联合承担的。该项目研究自然语言生成系统的参考体系结构。

尽管自然语言生成 (Natural Language Generation) 的一般问题仍然远未被人类理解,但这一技术已经开始被成功运用到工程实践中。这时,所面临的一个重要问题是:缺乏描述整体生成过程的标准视图,因为有了这一视图,才能更好的研究各种专门技术。RACS 项目的目的就是要提供这样一个视图:一个自然语言生成系统的参考体系结构。这一参考体系结构建立在研究者们的广泛共识的基础上,并把这些共识进一步提炼为明确的参考体系结构规范 (reference architecture specification)。它还确认了 NLG 中的基本构件和数据表述。尽管这一体系结构可能并不完全适合于所有的 NLG 应用程序,但在通常情况下该体系结构将有效的为资源共享和方法测评提供帮助。

RACS 体系结构定义在一个层次集合中,研究人员可以利用这个模型中自己需要的部分而不必考虑它们的全部。这是因为体系结构被分为 3 个独立的层次:

- ® 抽象数据模型:定义 NLG 系统操作的数据的类型,怎样用固定的术语表示它们,以及能够对其进行哪些操作;
- ® 实例化抽象数据模型中的类型:这样就把能够表示和特定的理论或理论集合联系起来;
- ® 一个可能的体系结构的集合:这些体系结构是通过向数据模型中的操作添加进一步的约束构建起来的。

2) ATLAS (Architecture and Tools for Linguistic Analysis Systems)

ATLAS 项目开始于 1999 年。它来自于美国 NIST, LDC (Linguistic Data Consortium) 和 MITRE 的提议。ATLAS 框架提供了一个面向语言学标注应用开发的体系结构,目标是解决一系列的应用程序需求,包括语料库的创建,评估框架,多模式的可视化等。ATLAS 框架在 LREC2000 会议上正式发布^[20]。

标注语料库已经成为当今自然语言技术研究的核心组成部分。但是,语料库的语种、所面向的学科、所采用的技术都不尽相同,缺乏公共的交换和存储格式已成为一个严重问题。一种可行的解决方案是引入一种通用的标注模型,通过它作为中间层来操纵标注数据。这种间接方式的最大优点在于把物理存储和应用程序逻辑相分离。ATLAS 项目的基本目标就是为语言学标注的多样性提供这样的一种抽象机制。这种抽象是对 Bird 和 Liberman 的标注图 (Annotation Graphs) 的扩展,能够在任意维度上的表示复杂标注。ATLAS 由 4 个主要构件组成:

- ® 标注本体(annotation ontology)构件;
- ® 应用程序编程接口(API)构件;
- ® 语言学数据交换格式构件;
- ® 类型定义基础框架构件;

其中,标注本体(annotation ontology)构件是 ATLAS 框架的核心构件,因为 ATLAS 的其余部分正是以它提供的抽象为基础而建立起来的。这些抽象可以使用多种程序设计语言实现。NIST 创建了数据模型的 Java 实例,应用程序通过调用 API 构件可以用简单的操作来访问核心对象。ATLAS 交换格式构件(AIF,ATLAS Interchange Format)用于支持语言学数据的交换和重用,它能够把标注数据串行化输出为 XML。元标记框架构件(MAIA ,Meta - Annotation Infrastructure)定义了类型基础框架,用来对 ATLAS 的通用结构做出某些限制以支持特定的应用。

3) GCSI(Galaxy Communicator Software Infrastructure)

GCSI 是一个用于构建对话系统的开放源代码的体系结构,是从 MIT 的 Galaxy 对话系统发展而来的。Galaxy Communicator 项目启动于 1998 年,由 MITRE 进行开发和维护,并受美国 DARPA 资助。Communicator 项目的目的是为创建各种具有口语能力的界面提供支持,既包括仅有语音的界面,也包括集成了图像、地图、手语等的复杂界面。尽管 DARPA Communicator 项目已经结束,但其代码和文档仍然是开放的。目前的最新版本是 2002 年发布的 4.0 版。

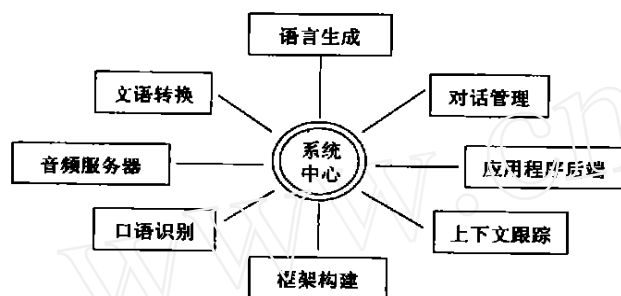


图 1 GCSI 系统结构

GCSI 是一个分布式的,基于消息的,用于创建口语对话系统的基础设施。它采用中央辐射式的结构,如图 1 所示。在 GCSI 基础上构建的应用程序都包括一个被称为系统中心的构件和一组被称为服务器的构件集合。系统中心和服务器之间的通讯通过一种被称为框架的命名属性值结构实现。这些框架构成了 GCSI 中

所有结构化通讯的基础。系统中心负责维护到各种服务器(语法分析器,语音识别器,等等)之间的连接和通讯,它还负责与一个内部服务器共同实现用户可视的管理任务。可以通过脚本语言编程对系统中心的消息流进行控制。GCSI 的服务器库对创建服务器提供支持,它为管理数据和通讯提供了很多方便的功能调用。

4) GATE(a General Architecture for Text Engineering)

GATE 是由英国 Sheffield 大学开发,最早的版本于 1996 年发布。GATE 是一个用于开发和部署语言工程构件和资源的框架和图形化的开发环境。GATE 体系结构使我们不仅能够为不同的语言处理任务(例如信息抽取)开发多种应用,而且能够为建设、标注语料库提供帮助,并能为所生成的应用程序提供测评。

GATE 框架由一个核心库(类似于主板)和一组可重用的语言工程构件组成。这些可重用的语言工程构件能够完成很多基本的语言处理任务,诸如词性标注,语义标注,等等。如果有必要,用户可以替换或扩展这些构件。GATE 框架实现了 GATE 体系结构,并为处理各种资源(包括数据的表示、导入和导出)提供了条件。此外,通过调用 GATE API,用户还可以脱离 GATE 环境的图形用户界面使用 GATE 提供的支持。

作为一个体系结构,GATE 定义了语言工程系统的组织结构,并分配了不同构件的任务,

确保了构件交互能够满足系统需求;作为一个框架,它为语言工程系统提供了可重用的设计和一组可重用的软件构件;作为一个开发环境,它能帮助用户在尽可能短的时间内完成新的语言工程系统的构建。由于 GATE 采用了组件化的模型,因此处理单元之间的耦合程度非常小,这非常有利于比较系统的不同配置或同一构件的不同实现(例如不同的语法分析器)之间的性能差异。

3 展望

作为一个富有交叉色彩的新研究领域,语言工程的软件体系结构还处在不断发展中。进一步的研究工作主要包括如下两个方面:

1) 基于网格计算等下一代网络技术的语言工程软件体系结构

我们认为一个值得注意的趋势是基于网格计算等下一代网络技术的语言工程软件体系结构研究。计算网格是一种软件和硬件结合的基础底层结构,能够可靠、一致、有普遍性、以及代价较低的使用高层计算能力。通过基于网格计算的虚拟平台,可以根据需要重新分配计算机资源。网格计算为数据密集或计算密集型的应用提供了大规模的分布式资源和处理能力。

考虑到语言工程所必需的大量数据以及分析这些数据所需的高强度的计算处理,结合网格计算等下一代网络技术的软件体系结构研究将在语言工程领域发挥日益重要的作用。已有研究人员开始探索基于网格结构的自然语言处理系统。例如,文献[21]提出了一个基于网格计算的组件化、可扩展的自然语言处理应用体系结构,对资源发现机制、构件识别、多构件应用设计、网格服务接口、网格应用规范语言等等方面的问题进行了讨论。在该体系结构中,语言资源的发现机制和标准接口建立在网格方法的基础之上。文献[22]讨论了 NLP 基础设施的编程接口,包括脚本语言、GUI 界面,以及用于分布式 NLP 系统开发的 Web 服务。

在国内,中科院计算机语言信息工程研究中心已经开始研究开发与 Web 服务相结合的多策略机助翻译系统^[25]。该系统有机融合多种机器翻译技术,具有良好的可扩展性和可维护性,而且为语言处理应用的第三方软件开发和集成提供了新的服务模式。文献[26]认为以 XML 为基础的 Web 服务是分布式环境中文信息处理技术的发展方向,并提出了一个中文信息处理服务体系框架的设想方案。

2) 软件构件化的语言资源研究和建设

我们认为语言工程的软件体系结构研究中的另一明显趋势是计算资源、数据资源和相关接口正在朝着软件构件化、开放化、标准化的方向发展。软件构件化是整个软件工程领域正在发生的变革。复杂软件系统的集成,就是要使体系中的各个层次能够有效配合而形成一个有机的整体。而复杂系统集成的关键,是基于体系结构的集成。因此,必须按照体系结构来定制构件,并将其安装到合适的层次位置上,才能使系统有效运作和集成。

文献[23]介绍了以 XSLT 为基础集成多种自然语言处理软件构件的项目。它成功的把词态分析、词性标注、词典、命名实体识别、短语 chunk 等深层和浅层自然语言处理构件完全自动的集成到基于 XML 的德语在线新闻句子分析系统之中,极大地改善了整体性能和健壮性。欧盟在 DG XIII 项目中创建的语言工程标准专家顾问组(EAGLES)提出了以 SGML 为基础的语料库编码标准 CES(Corpus Encoding Standard)。在该标准的指导下,人们已进一步提出了以 XML 为基础的语料库编码标准 XCES(Corpus Encoding Standard for XML)。其他基于 XML/SGML 的研究也已经受到研究人员的广泛关注。

参 考 文 献:

- [1] R. Mitkov, Language Engineering on the Highway: New Perspectives for the Multilingual Society[A], In Proceedings of NLPRS[C], Korea, 1995, 226 - 231.
- [2] H. Abelson, G. Sussman, and J. Sussman, The Structure and Interpretation of Computer Programs[M], MIT Press, Cambridge, Mass, 1985.
- [3] Ian Sommerville, Software Engineering[M], 6th edition, Addison - Wesley, Reading, MA, 2001.
- [4] Jochen Leidner, Current Issues in Software Engineering for natural language processing[A], Jon Patrick co - chairs, Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS) [C], Edmonton, Canada, 2003, 45 - 50.
- [5] H. Cunningham, A Definition and Short History of Language Engineering[J], Journal of Natural Language Engineering, 1999, 5(1): 1 - 16.
- [6] H. Thompson, Natural language processing: a critical analysis of the structure of the field with some implications for parsing[A], In K. Sparck - Jones and Y. Wilks, editors, Automatic Natural Language Parsing[C], Ellis Horwood, 1985, 76 - 81.
- [7] D. Crystal, A Dictionary of Linguistics and Phonetics[M], 3rd edition, Blackwell Publishers, Oxford, 1991.
- [8] B. Boguraev, R. Carigiano, and J. Tait, Editorial[J], Journal of Natural Language Engineering, 1995, 1 - 2.
- [9] 冯冲, 江贺, 冯静芳. 软件体系结构理论与实践[M]. 北京: 人民邮电出版社, 2004. 1.
- [10] M. Shaw, P. Clements, A Field Guide to Boxology: Preliminary Classification of Architectural Styles for Software Systems[A], In: Proceedings COMPSAC97, 21st Int'l Computer Software and Applications Conference[C], R. Soley Editor, 6 - 13, 1997.
- [11] Hayes - Roth, F. Architecture - based acquisition and development of software: Guidelines and recommendations from the ARPA Domain - Specific Software Architecture (DSSA) Program[M], Technical Report, Teknowledge, Inc., 1994.
- [12] Will Tracz, DSSA (Domain - Specific Software Architecture): pedagogical example[J], ACM SIGSOFT Software Engineering Notes (July 1995), 20(3) 17 - 26.
- [13] R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks, Description of the LaSIE system as used for MUC - 6[A], In: Proceedings of the Sixth Message Understanding Conference (MUC - 6) [C], Morgan Kaufmann, California, 1995, 118 - 126.
- [14] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham and Y. Wilks, Description of the LaSIE system as used for MUC - 7[A], In Proceedings of the Seventh Message Understanding Conference (MUC - 7) [A], 1998, 213 - 220.
- [15] F. Wolinski, F. Vichot, and O. Gremont, Producing NLP - based On - line Contentware[A], In Natural Language and Industrial Applications[C], Moncton, Canada, 1998, 58 - 63.
- [16] H. Poirier, The XeLDA Framework[A], Baslow workshop on Distributing and Accessing Linguistic Resources[C], Sheffield, 1999, 33 - 38.
- [17] R. Zajac, Reuse and Integration of NLP Components in the Calypso Architecture[A], In Workshop on Distributing and Accessing Linguistic Resources[C], 34 - 40, Granada, Spain, 1998.
- [18] P. Clarkson and R. Rosenfeld, Statistical Language Modeling using the SMU - Cambridge Toolkit[A], In Proceedings of ESCA Eurospeech[C], Greece, 1997, 110 - 115.
- [19] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, The HTK Book (Version 2.2) [M], Entropic Ltd, Cambridge, 1999.

(下转第 72 页)

需要指出的是本实验只是单纯研究单念时的元音在上海普通话与普通话中的差异,如果将元音系统放到自然语流中考虑,情况可能会更复杂,这点在以后的研究中将做进一步的分析探究。

参 考 文 献:

- [1] 周殿福,吴宗济.普通话发音图谱[M].北京:商务印书馆,1963.
- [2] 罗常培,王均.普通话语音学纲要[M].北京:科学出版社,1957.
- [3] 徐云扬. The phonetic value of the vowels, diphthongs and triphthongs in Beijing madarin[A]. 第五届全国现代语音学学术会议论文集[C]. 2001, 54 - 60.
- [4] 李思敬. 论现代汉语普通话中儿系列字的音值和儿音缀的形态音位[J]. 中国语言学报, 1988, (3), 301 - 316.
- [5] 许宝华,陶环. 上海方言词典[M]. 江苏教育出版社, 1997 年 12 月第 1 版.
- [6] 赵元论. 语言问题[M]. 北京:商务印书馆,1980.
- [7] 赵元任(倪大白译). 吴语的对比情况[J]. 国外语言学, 1980, 第 5 期.
- [8] 李荣. 上海方言词典[M]. 江苏:江苏出版社, 2000.
- [9] 钱乃荣. 跟我学上海话[M]. 上海:上海教育出版社, 2002.
- [10] Janet Fletcher and Andrew Butcher, 2003, Local and Global Influences on Vowel Formants in Three Australian Languages[A]. 15th ICPHS Barcelona[C].
- [11] <http://www.praat.org>.
- [12] <http://www.speechcon.com>.

(上接第 60 页)

- [20] Bird, S., Day, D., Garofolo, J., Henderson, J., Laprun C. and Liberman, M., 2000. ATLAS: A flexible and extensible architecture for linguistic annotation[A], in Proceedings of LREC 2000[C], Athens, Greece, May 2000, 1699 - 1706.
- [21] Baden Hughes, Steven Bird, Grid - Enabling Natural Language Engineering[A], Jon Patrick co - chairs, Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS) [C], Edmonton, Canada, 2003, 31 - 38.
- [22] James R. Curran, Blueprint for a High Performance NLP Infrastructure[A], Jon Patrick co - chairs, Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS) [C], Edmonton, Canada, 2003, 39 - 44.
- [23] Ulrich Schäfer, WHAT: An XSLT - based Infrastructure for the Integration of Natural Language Processing Components[A], Jon Patrick co - chairs, Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS) [C], Edmonton, Canada, 2003, 9 - 16.
- [24] H. Cunningham, Software Architecture for Language Engineering [D]. Unpublished PhD thesis, University of Sheffield, 2000.
- [25] 冯冲,陈肇雄,黄河燕. 基于 Web 服务的辅助翻译系统体系结构研究[A]. 第二届全国学生计算语言学研讨会(SWCL2004) [C], 北京, 2004.
- [26] 李宁. XML ——中文信息处理的变革之路[J]. 中文信息学报, 2003, 17(2): 54 - 59.