基于软件构件的机器翻译研究方法*

刘 群　　张 祥

（中国科学院计算技术研究所二室　北京 100080）

摘 要：机器翻译系统的研究开发工作是十分艰巨的。为了减轻开发的工作量，提高代码的可重用性，我们采用面向对象的方法，设计并实现了一个通用机器翻译开发平台。该平台提供了一系列的软件构件，这些软件构件实现了机器翻译中很多常用的数据结构和算法。该平台不依赖于具体的机器翻译系统和语法理论体系，可以用于构造不同类型的机器翻译系统。

关键词：机器翻译　软件构件　面向对象方法

A Software Component Approach to Machine Translation

Liu Qun　　Zhang Xiang

(Institute of Computing Technology, Chinese Academy of Science)

Abstract: It is arduous to develop a machine translation system. In order to reduce the work and make the code more reusable, we develop a General Development Platform for MT System based on the object-oriented method. The platform provides a series of software components, which implement many common data structures and algorithms in the MT field. The platform is independent of specific MT systems and grammar theories. Thus it can be used to build different kinds of MT systems.

Keyword: Machine Translation, Software Component, Object-Oriented Approach

一、前言

机器翻译系统的研究开发工作是十分艰巨的，研究工作者往往要在系统的具体编程实现上花费大量的时间精力，而很难将主要精力集中到所研究的内容上。

专用的知识描述语言的出现，是机器翻译研究中的一大进步[1]，使得语言工作者在算法相对固定的情况下，可以直接描述机器翻译所用到的各种知识，而不必考虑程序实现上

* 本项研究受国家八六三项目基金资助

的各种细节。语言工作者可以专注于他们所熟悉的语言问题，而不再需要了解过多的程序知识。

然而，开发一个机器翻译系统仍然是一件非常艰巨的工作。由于机器翻译所固有的复杂性，软件开发者需要对机器翻译的每一方面的所有细节都有较为深入的了解。另外，由于工作量巨大，开发一个机器翻译系统往往需要大量的人力、物力的投入，以及财力的支持。这也使得很多研究工作者对机器翻译的研究望而却步。

我们认为，存在以上问题的一个重要原因，就在于机器翻译研究中使用的软件代码的可重用性太低。每个机器翻译研究小组都需要编制自己的一套程序，而这套程序不仅不能为其他小组的研究人员所使用，甚至同一小组中，也只有少量编程人员能掌握这套程序，其他人即使有什么好的想法，也很难编制出相应的程序来进行实验。当已有的系统不能满足实际工作的要求，需要对系统的能力进行扩充时，当我们需要尝试某种新的语法体系，或改变翻译的源语言或目标语言时，我们仍然面临着大量修改程序的艰巨任务。所有这一切，都严重地影响了人们对机器翻译的研究热情，阻碍了机器翻译研究工作的发展。

其实，现代程序技术的进步，面向对象的理论与技术的成熟，已经为解决代码重用的问题提供了有效的手段[2,3,4]。例如 Microsoft 公司的 MFC 类库和 Borland 公司的 OWL 类库，使得任何一个初级的编程人员都可以在 Windows 环境下轻松地编写出漂亮的用户界面程序[5]。类似地，我们希望构造一个通用的机器翻译开发平台，使机器翻译研究工作者不再为繁重的编程工作所困扰，而能够专注于他们所研究的问题。

二、软件构件方法

软件构件方法的出现，得益于面向对象理论的发展和技术的成熟。各种面向对象的编程语言的推广使用，使得提供软件构件的做法成为可能。

我们这里所说的软件构件，就是指将一组数据结构与算法封装在一起，以类库的形式提供给用户使用，用户只需通过给定的接口来访问该构件，而无须了解构件内部的具体实现方式。通过继承和重载等手段，用户还可以改变该构件的行为，或增加新的功能。

依据这一设想，我们设计并实现了一个基于软件构件的“通用机器翻译系统开发平台”，该平台具有以下特征：

(1) 提供的软件构件足够丰富，能够用于构造一个机器翻译系统；
(2) 各构件之间相对独立，相互之间的耦合程度最小，因而很多基本的构件也可以单独使用，或用于自然语言处理中除了机器翻译以外的其他研究领域；
(3) 实现了机器翻译研究中一些常用的数据结构，以及比较成熟的算法；
(4) 具有可扩充性，用户可以重新定义某个功能或增加新的功能；
(5) 不依赖于某个特定的机器翻译系统；
(6) 不局限于某一具体的语法体系，通过重新定义某些功能，可以支持不同的机器翻译方法（目前只支持规则方法，暂不支持语料库与统计方法）；
(7) 独立于具体的自然语言，同时又为各种自然语言的具体实现提供了相应的接口，可用于开发各种语言对的机器翻译系统；

(8) 提供知识库的开发管理工具，为语言工作者提供方便。

三、类库基本结构

所有的软件构件都以C++类库的形式提供。
本平台主要包括以下几个类库：

(1) 知识库类库

有关知识库的类库主要包括：

1. 基本知识库类：所有知识库类都是基本知识库类的派生类，定义了知识库的基本数据结构和操作；
2. 语言模型类：用于定义语言模型；
3. 词典：用于实现机器翻译用的词典；
4. 各类规则库：用于实现机器翻译各阶段所使用的规则库；
5. 词典模板库：词典模板库用于存放词典模板，词典模板用于定义新词
6. 实例库：实例库用于存储翻译过的句子的各种信息；
7. 各类知识库的界面库：用于实现基于窗口的各类知识库管理界面，这些类库不是基本知识库类的派生类（见图1）。

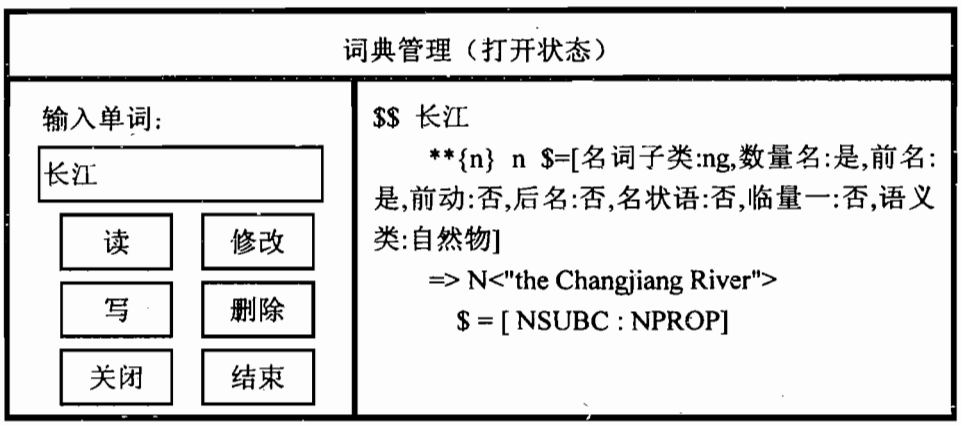

图1、词典管理界面

以上这些类实现了机器翻译中常见的知识库的创建、管理及查询操作，还可以支持一个小组中多个人同时通过网络对知识库进行操作。

在以上类库中，使用严格的语言模型管理。语言模型本身也是一个知识库，用户可以通过改变语言模型的内容来改变自己所使用的语言学基础。语言模型是其他所有知识库的基础。其他所有知识库中所使用的语言知识描述符号，包括描述词法、句法、语义知识的分类、属性（名称及取值范围）等等，都必须符合语言模型中的定义。

通过重载知识库类中的格式处理函数，用户可以创建符合自定义格式的数据库。这些

类库所实现的知识库也可用于机器翻译以外的其他目的。

(2) 特征网络类库

特征网络是我们定义的一种数据结构，用于表达语言成分的各种句法语义信息，它包含了一般意义上的特征结构和语义网络的表达能力。特征网络类库主要包括以下一些类：

1. 基本原子类：定义原子的一些基本操作，是各种具体原子类的基类；
2. 各种原子类：定义各种具体的原子类，我们定义了层次型、符号型、数值型和布尔型四种类型的原子，可以根据需要进行扩充其他类型；在原子这一级，可以进行与、或、非等逻辑运算；
3. 特征结构类：定义基本特征结构，特征结构是组成特征网络的“结点”；
4. 合一现场类：定义合一现场，合一现场是每一次合一运算后的数据现场。

特征结构与合一运算已成为自然语言处理和机器翻译中知识表示的一种最基本形式，以上这些类实现了对特征结构和合一运算的支持。尤其是合一现场类，使我们可以实现真正意义上的合一运算，而不是“伪合一”运算。

这些类也可以用于自然语言处理中除机器翻译以外的其他场合。

(3) 句法成分类库

句法成分用于保存句法分析产生的所有树结构。具体包括以下类：

1. 原文词串类：表达原文词串；
2. 原文词段类：表达原文词串中的一个片段；
3. 原文结点类：表达原文分析产生的结点；
4. 译文结点类：表达转换生成产生的译文结点；
5. 译文词段类：表达一个译文词；
6. 译文词串类：表达译文词串；
7. 原文结点表：用于存放一系列原文结点指针；
8. 译文结点表：用于存放一系列译文结点指针。

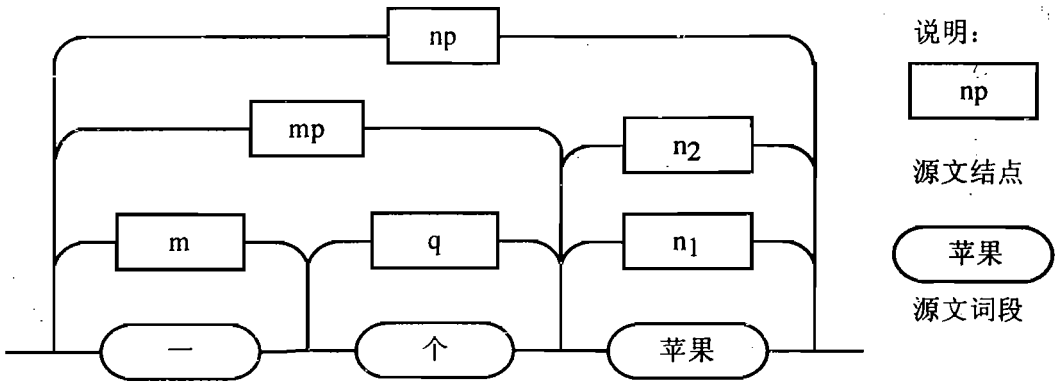

图 2 、 Chart 结构示意图

以上这些类实现了 Chart Parsing 中所使用的 Chart 结构（见图 2），因而使用这些类很容易实现 Chart Parsing 算法。在这种算法中，通过调整原文结点表的排列顺序及新结点的激活策略，可以模拟句法分析中所使用的很大一类算法。

(4) 翻译系统类库

翻译类库中实现了一个基本翻译系统类，这是一个独立于具体语言的翻译系统类，它实现了翻译的主要数据结构，包括源文词串、源文结构、译文结构和译文词串，以及结构分析、转换和结构生成算法，对于词法分析和词法生成算法，只提供了接口，但没有具体实现（见图 3）。

通过基本翻译系统类可以派生出各种语言对的翻译系统类，只要在派生类中实现源文词法分析和译文词法生成算法即可。也可以重载基类中的结构分析、转换和生成算法，以实现不同类型的翻译系统。

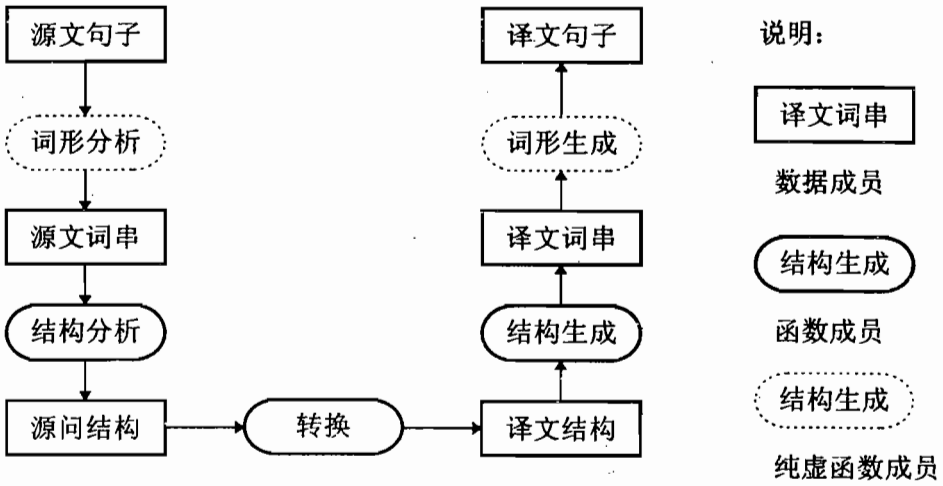

图 3 、翻译的流程 及 基本翻译系统类的成员

四、结论

机器翻译开发平台的建设，将使后来的研究者站在前人的肩膀上进行工作，不必事事都从头做起，从而可以节省大量的时间和精力，避免低水平的重复和财力的浪费。

目前，我们已经在 IBM/PC 计算机上使用 Microsoft Visual C++语言实现了该开发平台，并在此基础上开发了一个汉英机器翻译模型系统，取得了良好的效果。今后，我们希望能对这套平台进一步扩充，使之更为合理和完善。近期内，我们将提供对统计学研究方法，即语料库的方法的支持。

参考文献：

[1] 冯志伟，自然语言机器翻译新论，语文出版社， 1995

[2] Pascoe G.A., Elements of Object-Oriented Programming, Byte, Aug 1986

[3] Peter Coad, Edward Yourdon, Object-Oriented Analysis, Yourdon Press,1990

[4] Peter Coad, Edward Yourdon, Object-oriented Design, Yourdon Press, 1991

[5] David J.Kruglinski, Inside Visual C++, Mircosoft Press, 1993