湖南大學

HUNAN UNIVERSITY

毕业论文

论文题目	面向机器翻译语言模型的优化技术研究
学生姓名	罗雅文
学生学号	201308030320
专业班级	通信1303
学院名称	信息科学与工程学院
指导老师	吴建辉
学院院长	李肯立

2017 年 5 月 31 日

湖 南 大 学

毕业论文原创性声明

本人郑重声明：所呈交的论文是本人在老师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

学生签名：　　日期：2017 年 5 月 31 日

毕业论文版权使用授权书

本毕业论文作者完全了解学校有关保留、使用论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本论文。

本论文属于

1、保密□，在_____年解密后适用本授权书。

2、不保密√。

（请在以上相应方框内打“√”）

学　　生签名：　　日期：2017 年 5 月 31 日

指导教师签名：　　日期：2017 年 5 月 31 日

面向机器翻译语言模型的优化技术研究

摘　要

全球化的推进导致文化间相互交流的需求愈发扩大，而为了解决因语言不同而产生的沟通问题，翻译成为了一项必备的功能。翻译的本质是把一种语言信息，通过在各种人为设定语法规则下，转变为另外一种语言信息的过程。并且随着计算机性能的飞速提升与发展，许多奋斗在科研前线的科研人员们发现，利用计算机来完成翻译的任务是一件划时代、超越时代的壮举。这便是机器翻译思想的雏形与诞生缘由。

本文分析了机器翻译的历史和研究现状，并剖析了机器翻译发展过程中各时期的主流语言模型构造方法，概述了每种方法的优点和局限性。之后论述了整个统计机器翻译流程步骤及其实现原理。列举了机器翻译的多个主流语言模型，详细分析了统计机器翻译中的N元语法模型，并实现了对N元语法模型关键算法的优化。实现了一个基于N语法模型的统计机器翻译系统，并详细介绍了系统架构和各个模块的部署方式。之后还测评了基于深度学习的机器翻译效果和基于神经网络模型的机器翻译效果。

关键词：自然语言处理；机器翻译；语言模型

Research on Optimization Technology of Language Modeling for Machine Translation

Abstract

The advancement of globalization has led to a greater need for intercultural communication, for solving the communication problems caused by different language, translation has become a necessary function. The essence of translation is the process of transforming a language message into another language under various artificial grammar rules. Many researchers at the front line of scientific research found that with the rapid improvement and development of computer performance, the use of computers to complete the task of translation is an epoch-making, beyond the feat of the times. This is the prototype and the birth of the reason of the machine translation thought.

This paper analyzes the history of machine translation and the current situation of research, and analyzes the main language model construction methods in the process of machine translation development, and summarizes the advantages and limitations of each method. And then discusses the whole process of statistical machine translation process and its realization principle. This paper analyzes the multi - mainstream language model of machine translation, analyzes the N - gram grammar model in statistical machine translation in detail, and realizes the optimization of N - element grammar model. A statistical machine translation system based on N grammar model is implemented, and the architecture of the system and the deployment of each module are introduced in detail. The effect of machine translation based on depth learning and the effect of machine translation based on neural network model are also evaluated.

Key words: Natural Language Processing; Machine Translation; Language Model

目 录

1. 绪论

1.1 选题研究背景及研究意义

国与国之间，不同民族之间，不同文化之间，自古以来就有相互交流的传统和需求。古时担任两种甚至多种语言翻译的使员，通常肩负着搭建起两种文化思想激烈碰撞与交流的桥梁的重任，不难得见，翻译正确与高效是十分重要的。全球化的推进导致文化间相互交流的需求愈发扩大，并且随着计算机性能的飞速提升与发展，许多奋斗在科研前线的科研人员们发现，利用计算机来完成翻译的任务是一件划时代、超越时代的壮举。这便是机器翻译思想的雏形与诞生缘由。

机器翻译（英文全称 Machine Translation，英文简称 MT）是一种借助计算机来运算，从而实现不同种自然语言之间的自动翻译，它主要操作和研究的对象，是人类沟通所使用的各种自然语言。它们可以是文本，也可以使语音，但主要形式是文本，其中语音形式还涉及到了语音识别和语音合成。

机器翻译系统（英文全称 Machine Translation System，英文简称 MTS）则是一种翻译过程全自动或半自动的计算机系统。机器翻译系统有双语系统也有多语系统。双语系统在两种指定自然语言下能够进行翻译，可以单向也可以双向[1]；多语系统则是允许多种语言同时翻译。

机器翻译萌生于上世纪四十至五十年代，当时的方法主要是利用词典对照翻译和词频统计来互译，是非常单纯的方法。

六十年代时机器翻译发展开始停滞不前，当时甚至发表了怀疑机器翻译存在必要性的报告。

到了七十年代，计算机性能提升速度加快，人工智能也开始发展起来，悄无声息里，机器翻译系统中，基于规则的方法等语言模型开始兴起。七十年代末，一些可以被称作真正意义上的机器翻译系统横空出世，标识着第二代机器翻译的时代开启了。

八十年代是机器翻译发展的黄金时代，基于规则的方法发展得越发成熟，同时研究人员也发现了基于规则的方法的局限性，开始考虑其基于语料库的方法等。

九十年代则是第三代机器翻译的时代，在理论发展里，基于语料库的方法的研究突飞猛进。IBM 公司率先提出基于统计的方法的五个 IBM 模型，其理论依据是信息论中

的噪声信道模型理论。

进入二十一世纪后，采用基于统计的模型的机器翻译取得了非常大的进步[2]。各式机器翻译系统相继推出，其中佼佼者有 Moses，Joshua 等，机器翻译开始进入繁荣期。

机器翻译主要操作和研究的对象，是人类沟通所使用的各种自然语言。它们可以是文本，也可以使语音，但主要形式是文本，其中语音形式还涉及到了语音识别和语音合成，此属于另外的研究领域，不在本文研究探讨范围内。机器翻译所必须的工具是计算机，并且处理过程自动化。为了处理对自然语言的翻译，机器翻译发展了不少对自然语言的处理技术，换而言之，机器翻译是自然语言处理研究领域的一个细分方向，并且，机器翻译和自然语言理解、计算语言学都有很深的联系。

1.2 国内外研究现状

现主流的研究都认为统计语言模型是利用数学描述来展现内含于语言文字之中的语言学规则。

基于规则方法是早期机器翻译的大潮流，吸收大量语言学规则，并对语言学知识进行人工处理总结，对源语言和目标语言的表达规律进行人工分析，之后对其进行形式化符号描述，形成翻译规则库。在翻译时依据规则库，对词汇，语法，句法进行来替换和调整，生成目标语言的翻译。

基于统计的方法则是如今机器翻译的主流研究方向[3]。基于统计的方法需要构建大规模的语料库。统计的核心思想，是把目标语言视作通过某种信道模型传输之后的信息，源语言是待传输的信息，要得到经过翻译的目标语言译文，其实就是要得到经过信道传输之后的信息。

N 元语法模型是基于统计的方法中的一种很优秀的模型。N 元语法模型是一种基于阶马尔科夫链的概率语言模型，立足于一种特定情景下，即第个项的出现概率仅仅与前项有关，与其他位置的任何项无关，常用于预测这种序列中下一项的概率。这些项通常是从文本或语音语料库中收集得来。

基于神经网络的语言模型也发展迅猛，其中的声名远扬的便是 Word2Vec 模型，Word2Vec 模型是由美国谷歌公司研制开发用于词向量训练的一款工具，特点是高效性和准确性。通常认为 Word2Vec 模型属于神经网络中的深度学习模型。Word2Vec 模型的作用，是可以通过大量的输入语料，讲语料中的词组化为一个个 K 维的词向量。

1.3 主要研究内容及组织结构

本文的主要研究内容是为了探究面向机器翻译语言模型的优化技术的研究，对机器翻译各种方法都进行了详细剖析，并对表现优异的语言系统进行评测和优化，最终实现了一个基于统计模型的机器翻译平台。

本文其他部分内容组织结构如下：

第二章，分析了机器翻译的历史和研究现状，并剖析了机器翻译发展过程中各时期的主流语言模型构造方法，概述了每种方法的优点和局限性。之后论述了整个统计机器翻译流程步骤及其实现原理。

第三章，列举了机器翻译的多个主流语言模型，详细分析了统计机器翻译中的 N 元语法模型，并实现了对 N 元语法模型关键算法的优化。之后详细论述了 Word2Vec 模型里面的 Skip-gram 模型和 CBOW 模型的原理和优缺点。

第四章，实现了一个基于 N 语法模型的统计机器翻译系统，并详细介绍了系统架构和各个模块的部署方式。之后还测评了基于深度学习的机器翻译效果和基于神经网络模型的机器翻译效果。

第五章，总结本论文的工作。

2. 机器翻译

2.1 机器翻译概述

翻译的本质是把一种语言信息，通过在各种人为设定语法规则下，转变为另外一种语言信息的过程。机器翻译（英文全称 Machine Translation，英文简称 MT）是一种借助计算机来运算，从而实现不同种自然语言之间的自动翻译，机器翻译系统（英文全称 Machine Translation System，英文简称 MTS）则是一种翻译过程全自动或半自动的计算机系统。机器翻译主要操作和研究的对象，是人类沟通所使用的各种自然语言。它们可以是文本，也可以使语音，但主要形式是文本，其中语音形式还涉及到了语音识别和语音合成，此属于另外的研究领域，不在本文研究探讨范围内。机器翻译所必须的工具是计算机，并且处理过程自动化[4]。为了处理对自然语言的翻译，机器翻译发展了不少对自然语言的处理技术，换而言之，机器翻译是自然语言处理（英文全称 Natural Language Processing，英文简称 NLP）研究领域的一个细分方向，并且，机器翻译和自然语言理解（英文全称 Natural Language Understanding,英文简称 NLU)、计算语言学（英文全称 Computational Linguistic，英文简称 CL）都有很深的联系。NLU 是 CL 的核心，也是机器翻译的基础。其中，自然语言处理又是人工智能（英文全称 Artificial Intelligence，英文简称 AI）的一个研究细分方向，即是说，机器翻译和人工智能也有紧密的联系。

机器翻译系统有双语系统也有多语系统。双语系统在两种指定自然语言下能够进行翻译，可以单向也可以双向[5]；多语系统则是允许多种语言同时翻译。机器翻译的方法有许多种，包括：

（1）基于规则（英文全称 Rule-Based，英文简称 RB）；

（2）基于语料库（英文全称 Corpus-Based，英文简称 CB）；

基于规则方法是早期机器翻译的大潮流，并衍生发展处了许多细化方法，如：

（1）直接翻译（英文全称 Driect Translation，英文简称 DT）；

（2）基于转换（英文全称 Transfer-Based，英文简称 TB）；

（3）基于中间语（英文全称 Interlingua-Based，英文简称 IB）；

（4）基于知识（英文全称 Knowledage-Based，英文简称 KB）。

基于语料库的方法则是如今机器翻译的主流研究方向，主要分类为两种：

（1）基于实例（英文全称 Example-Based，英文简称 EB）；

（2）基于统计（英文全称 Statistics-Based，英文简称 SB）。

2.1.1 基于规则

基于规则的翻译方法，吸收大量语言学规则，并对语言学知识进行人工处理总结，对源语言和目标语言的表达规律进行人工分析，之后对其进行形式化符号描述，形成翻译规则库。在翻译时依据规则库，对词汇，语法，句法进行来替换和调整，生成目标语言的翻译。

直接翻译方法是依据源语言（英文全称 Source Language，英文简称 SL）与目标语言（英文全称 Target Language，英文简称 TL）词汇，短语之间的对应关系，直接将源语言的词汇和短语替换成目标语言的对应词汇和短语，不考虑词汇之间的调序来生成译文[6]。这样的翻译方法生成的译文效果比较差，一般不能满足需求。

基于转换的方法背景是考虑到直接翻译的翻译效果不是很好，且有许多问题，基于转化的机器翻译方发发展起来。其分为四个阶段：对源语言分析——面向源语言抽象表达——面向目标语言抽象表达——生成目标语言。TB 翻译方法对源语言的分析处理，只考虑句法层面，且和目标语言相独立。同时准备好双语对应词典，再使用目标语言词汇或短语替换时，将考虑上下文，并考虑两者的句法结构差异，同时进行结构转换。

基于中间语的方法在多语系统中萌芽并发展起来，这种方法的思想是建立一套与各种自然语言相独立，又能自由转换向各种自然语言的的人工语言作为中间媒介[7]。在该种方法中，目标语言和源语言无直接关联，只有中间语言联系二者的词汇、句法结构及更深层次的分析。这种方法对于多语系统来说，是资源的最大化利用，但是实现难度大。

基于知识的方法则是在翻译时不仅用到语言学规律，还参考更多的，更广阔领域的知识，这种翻译方法主要应用在人工智能方向上。

基于规则的方法曾经在机器翻译发展史上的地位上举足轻重，到现在也有着深远的影响。基于规则的方法发展至今，有了许多变化。以前的自然语言规律，大都靠语言学家人工总结，但现在更倾向于使用机器学习，自动从语料库中习得规律；以前是建立适用于整个某种自然语言的规则，现在则更加细化，对于每个细分领域都建立单独的小规则库。

基于规则的方法历经多年发展，积累了许多语言规则，建立了庞大的规则库。但库

越繁笼，越不利于系统的灵活性，而且新的词汇、语言习惯、句法语法又在新生，越来越不适合发展基于规则的方法。而在大数据时代，如今计算机的计算性能的大幅增强，相关领域的研究人员开始研究起基于语料库的方法。

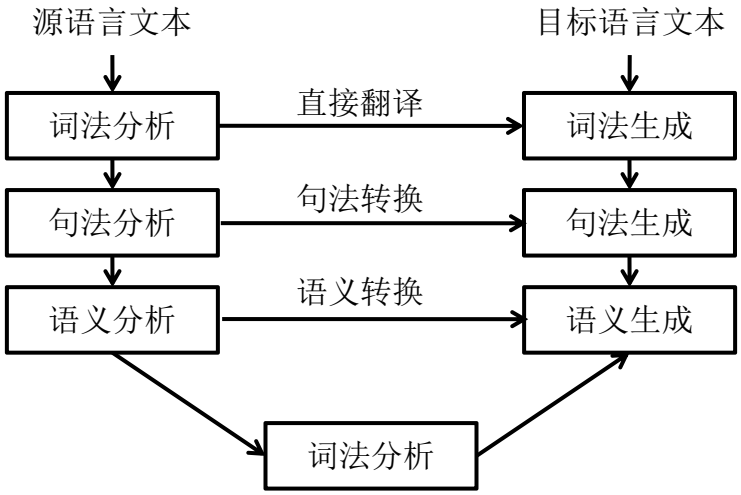

图 2.1 基于规则的机器翻译方法

2.1.2 基于语料库

基于语料库的方法依凭于大量语料构成的语料库，就是对语料库数据大规模的收集、整理、清洗，并有专门的研究领域，即语料库语言学（英文全称 Corpus Linguistic，英文简称 CL）。

基于实例的方法的基本方法是建立双语语料实例库，选择恰当的匹配算法来生成译文。这样做的理论依据是，在自然语言的实际运用中，许多语言实例都使用相近甚至一样的句式。该方法的基础就是双语语料实例库，库越大，翻译系统的鲁棒性和准确性就越高。

基于统计的方法与基于实例的方法有相似点，SB 方法也需要构建大规模的语料库[8]。统计的核心思想，是把目标语言视作通过某种信道模型传输之后的信息 T，源语言是待传输的信息 S，要得到经过翻译的目标语言译文，其实就是要得到经过信道传输之后的信息 S。这与统计有何关系呢？基于统计的方法的目的是想得到该信道模型，信道模型的建立，主要依据贝叶斯（Bayes）公式推导：

$$P(T \mid S) = \frac{P(T) \times P(S \mid T)}{P(S)} \tag{2.1}$$

推导后可得：

$$T = \arg\max P(T \mid S) = \arg\max_{T} P(T)P(S \mid T) \tag{2.2}$$

上述公式是统计机器翻译最基础的公式。其中 $P(T)$ 是指代经过信道模型之后的信息出现的概率，即目标语言中的词汇、短语出现概率。$P(S \mid T)$ 是指代原传输信息 S 在已知传输结果是 T 的情况下的出现概率，即在源语言的某词汇、短语在已知被翻译成目标语言的某种词汇、短语的概率，同理类比 $P(T \mid S)$。整个公式就是翻译了信道模型，即翻译模型，目的是要通过 S 还原出 T。由此可知，基于统计的方法最关键的三个点分别是：

（1）计算评估目标语言的语言模型 $P(T)$；

（2）计算评估翻译模型 $P(S \mid T)$；

（3）构建用于求解 $\arg\max_{T} P(T)P(S \mid T)$ 的快速准确的搜索算法。

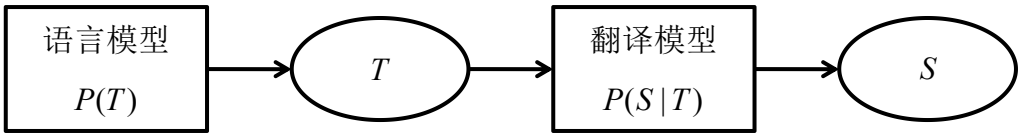

图 2.2 基于统计机器翻译中的信源信道模型

令目标语言构成的长句 T 由词汇 $W_1W_2...W_n$，那么，语言模型

$$\begin{aligned} P(T) &= P(W_1W_2...W_n) \\ &= P(W_1) \times P(W_2 \mid W_1) \times ...P(W_n \mid W_{n-2}W_{n-1}) \end{aligned} \tag{2.3}$$

参数 $P(W_i \mid W_{i-2}W_{i-1})$ 的计算将在之后的章节详细解说；而参数 $P(S \mid T)$ 与目标语言和源语言的平行语料有关，涉及到词汇对齐。

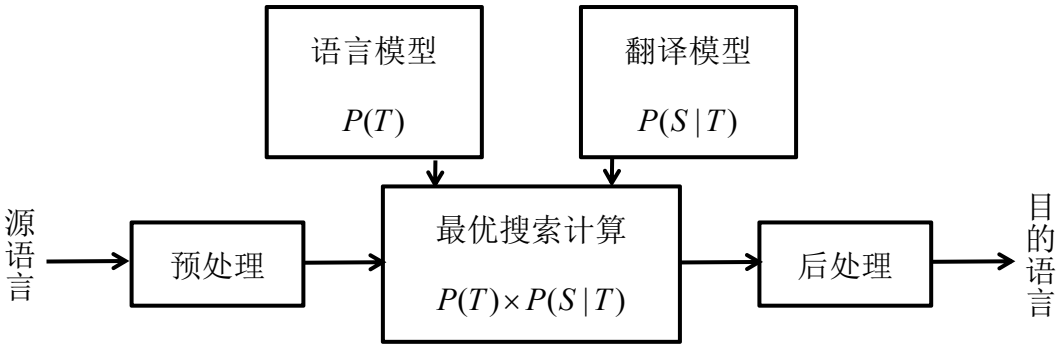

图 2.3 统计机器翻译模型

基于实例的方法是一种希望不深究内部原理，只依靠语料库的相似句子实例来类比来翻译。基于实例的方法要求大规模的至少长句级对齐的语料库。如果对齐的单位越小，显而易见，则翻译的准确性就越高，但这也意味着对数据的加工程度也越高。更大问题是，语料库的体量越大，就越需要设计一种高效的查询匹配算法。但 EB 方法的翻译系统维护成本低，也容易得到高质量的译文。

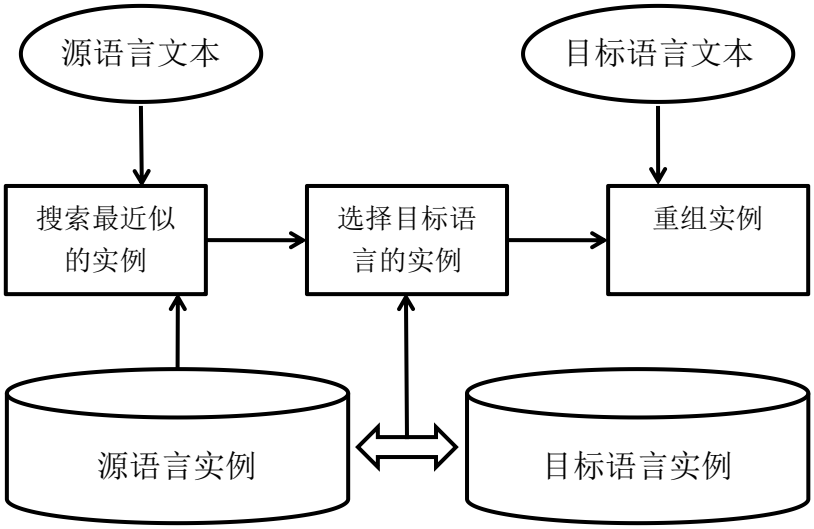

图 2.4 基于实例机器翻译体系结构

EB 方法和 SB 的方法都基于大规模语料库的支持，但它们之间的本质不同是，前者在翻译过程中，需要一直查询实例语料库，而后者则在翻译之前对全部词汇、短语进行了概率统计，在翻译过程中则不再查询。在实际现实生产生活中，SB 方法的机器翻译的运用和发展，比 EB 方法的机器翻译要更好更长远。

2.2 统计机器翻译系统流程

上一小节提到过，SB 方法依赖于大量的双语语料，并且来自源语言和来自目标语言的词汇要能互相翻译，即为双语平行语料，这些双语平行语料通常是被切割成很小的颗粒（片段）。翻译的过程总的来说，就是把目标语言的小颗粒去和源语言的小颗粒匹配，然后依据目标语言的句法语序，把这些颗粒拼接成完整的长句。

我们希望从双语平行语料中得到的双语片段，不管在源语言处还是在目标语言处，它们的语义都能保持一致。这意味这，如果在源语言里包含了某个词时，显然目标语言处也要包含这个词。对已经做好句对齐的双语语料进行词汇级的对齐，这种处理被称为词对齐（英文全称 Word Alignment，英文简称 WA），是如今绝大多数机器翻译所必须的流程之一。

做完了词对齐处理之后，我们得到了翻译流程中的最小单位。词对齐的的颗粒可大可小，颗粒越大，歧义越小，但匹配的灵活度也随之减小；反之，颗粒越小，歧义越大，但是更容易匹配更多句式和语法。如果我们把对齐的颗粒划分至最小，即词汇级的别的对齐，那么训练出来的翻译模型，即是基于词汇的翻译模型；如果将颗粒划分扩大一些，使得任意长度的片段也能对齐，那么将得到基于短语的翻译模型；如果不对双语语料做任何切分，则为基于长句的翻译模型。

我们对这些抽取出来的大量的对齐语料和翻译规则计算出概率模型，建立模型的方法有很多，基础的有最大似然估计（英文全称 Maximum Likelihood Estimation，英文简称 MLE）[9]。

大部分翻译系统的翻译流程，除了整理大量的翻译规则之外，还有两个非常重要部分：语言模型（英文全称 Language Modeling，英文简称 LM）和调序模型（英文全称 Reordering Model，英文简称 RM）。语言模型简而言之，就是预测一句话出现的概率的模型，它的作用是评价几个候选译文的得分从而能够挑选出最优秀的译文。翻译模型的存在理由则是因为源语言和目标语言句法结构的差异性，翻译过程中相对于源语言，目标语言的端的语序需要依据其句法规则进行调整。

简而言之，整个流程，就是处理好双语平行数据，生成基于双语语料的翻译模型，输入源语言下的带翻译长句，通过翻译模型寻找出与之匹配的目的语言的的译文。依据信道模型理论，其中翻译模型即为解码器，整个过程就是一个信息解码过程。通过训练

语料，生成翻译模型、语言模型，再通过测试语料进行调优。最后对模型做出评价。

2.2.1 词汇对齐

词汇对齐是整个机器翻译流程的基础，对后续各种处理操作来说必不可少。词汇对齐即是标注出双语句对中达到词汇级别的相互对应关系。目前应用较为广泛的是 IBM 模型，最常用的工具是 GIZA++。IBM 模型认为，词对齐其实是源语言到目标语言的翻译过程中的一个隐变量[10]：

$$p(\mathbf{f}\,|\,\mathbf{e}) = \sum_{\mathbf{a}} p(\mathbf{f},\mathbf{a}\,|\,\mathbf{e}) \tag{2.4}$$

$$p(\mathbf{f},\mathbf{a}\,|\,\mathbf{e}) = p(m\,|\,\mathbf{e})\prod_{i=1}^{m} p(ai\,|\,a_1^{i-1},f_1^{i-1},m,\mathbf{e})\,p(fi\,|\,a_1^{i-1},f_1^{i-1},m,\mathbf{e}) \tag{2.5}$$

IBM 翻译模型其实就是五个统计翻译模型，不过它们的复杂度依次递增。模型 l 是五个模型之中最简单的，也是另外四个模型进行计算的基础。

模型 1 只包含词汇间的互译概率；模型 2 将考虑上下文信息，记录词汇位置变化；模型 3 开始考虑歧义。这五个模型是如今基于词的 SMT 翻译模型的根底，更加是如今 SMT 里主流技术中的重要环节。

IBM 模型身为概率模型，有着严密的自身推演。一句话来说，模型 1 和 2 的于同一个公式的模型扩展，模型 3、4 和 5 属于另一个公式的模型扩展。但如果要从它们的复杂度来看，它们之间的关系是$1<2<3<4<5$，如果要从参数的从属关系来看，则是$1\subset 2\subset 3\subset 4\subset 5$，从计算顺序来看，是$1\to 2\to 3\to 4\to 5$。之后要用到的 GIZA++的源码中，就提供可为使用者自由定制迭代次数的功能。

2.2.2 选择模型

翻译模型根据处理的最小单元的量级和建模规则，分为了以下三类：

（1）基于词汇的翻译（英文全称 Word-based Translation，英文简称 WT）；

（2）基于短语的翻译（英文全称 Phrase-based Translation，英文简称 PT）；

（3）基于句法的翻译（英文全称 Syntax-based Translation，英文简称 ST）。

基于词汇的翻译是最早在统计机器翻译里发展并应用起来的模型。这种翻译模型的最小翻译单位就是词汇，颗粒度是三个模型中最小的。虽然该模型灵活度大，但因为没有保留上下文信息，翻译效果并不是非常理想，最大的问题就是消除歧义。如今基于词

汇的翻译已经应用范围没有原来那么广泛了。

基于短语的翻译是在 WT 翻译模型之后，紧接着发展起来的模型。这种翻译模型的最小翻译单位是短语，颗粒度是在三个模型中是中等级别。相比 WT 模型，包含了一定体量的上下文信息，是目前最主流的翻译模型。

基于句法的翻译也称为基于形式语法（英文全称 Formal Grammer）的翻译比较复杂，目前主流的两种：同步上下文无关语法（英文全称 Synchronous Context-Free Grammer，英文简称 SCFG）和基于反向转录语法（英文全称 Inverted Transduction Grammer，英文简称 ITG）。ST 模型相对于 WT 模型和 PT 模型，更强调了语言本身所蕴含的的逻辑信息，从翻译效果上来看，显然胜出；但是处理数据和运行效率比较低，这是阻碍目前 ST 模型应用的主要原因。

2.2.3 训练模型参数

众所周知，模型中间层参数的训练，对生成的译文，其本身的品质没有很大影响。这样需要使用最小错误率训练（英文全称 Minimum Error Rate Training，英文简称 MERT），利用优化集（英文全称 Tuning Date，英文简称 TD）来训练这些模型中间层参数。引入最小错误率训练的目的，是为了能在译文生成后，有自动打分评测监督训练过程，更一边能在自动评测上表现不错，另一边在人工评价上表现也不错。

MERT 通过在优化集上训练参数权重，使得在给定的情景下能达到最优化。一般常见的优化准则包括信息熵，双语评价替补（英文全称 Bilingual Evaluation Understudy，英文简称 BLEU），TER 等。这个步骤需要使用求解模型算法对优化集多次求解模型，每次求解模型可以得到多个表现最优的结果，并利用这些模型调整特征权重。当权重被更改后，它们的排序也会变化，它们之中的最优者，将被用于计算 BLEU 得分或 TER。如此这般便能得到一系列新的特征权重，将整个优化集的得分提升后，将进行下一轮调整。如此往复直至不能得到更多提升[12]。

2.2.4 自动测评技术

整个机器翻译流程完成后，通常需要对其进行检测。检测手段有人工测评和自动测评。人工测评准确度高，但耗费人力物力大，而且时间较长，不适合体量较大的机器翻译系统，在生产生活中应用较少，与此同时，自动测评技术发展了起来。相较于人工测

评，自动测评具有处理速度快，处理数量多，但准度精度不及人工。提升准确度仍然是目测自动测评技术的研究热点。目前较热门的自动测评技术有 BLEU 测评方法和 NIST（英文全称 National Institute of Standards and Technology，英文简称 NIST）测评方法。

BLEU 测评方法是由 IBM 公司首次提出的一种针对基于 n 元语法模型的机器翻译的自动测评方法[11]。它的原理是将机器翻译系统翻译出的译文，与人工翻译的候选译文做对比。详细来说，一般是选取机器翻译得到某个 N 元语法片段，查看它在人工翻译的候选译文中出现的次数比例（如若超出则倒减）。这种做法只考虑到了精确率而没有考虑召回率，所以 BLEU 针对比候选译文还要短的待测评译文有特殊的惩罚机制。

NIST 测评方法是由美国国防高级研究计划局（英文全称 Defense Advanced Research Projects Agency，英文简称 DARPA）带头研发的项目。NIST 测评方法是在 BLEU 方法的一种改进。它并不是单纯的将已配对的 N 元语法片段数目求和，而是求出每一个 N 元语法片段的信息量，然后求和再除以全部 N 元语法片段数目。计算信息量之后，就可以对每一个共现 N 元语法片段乘以它的信息量权重，之后求平均得到最终的评分[13]。

2.4 小结

本章分析了机器翻译的历史和研究现状，并剖析了机器翻译发展过程中各时期的主流语言模型构造方法，概述了每种方法的优点和局限性。之后论述了整个统计机器翻译流程步骤及其实现原理。

3. 常用语言模型分析

3.1 N-gram 语言模型

3.1.1 模型概述

N 元语法模型，又称为 n-gram 模型，n-gram 是指在给定的文本或语音序列中连续出现的 N 个项（item）。N 元语法模型是一种基于阶马尔科夫链（英文全称 Markov Model，英文简称 MM）的概率语言模型，立足于一种特定情景下，即第个项的出现概率仅仅与前项有关，与其他位置的任何项无关，常用于预测这种序列中下一项的概率。这些项通常是从文本或语音语料库中收集得来，在不同的应用领域中，它们可以代指为音素（phonemes），音节（syllables），字母（letters），词语（words）甚至碱基对（base pairs）。

当 N 为 1 时的 n-gram 模型被称为“unigram”；为 2 时被称为“bigram”；为 3 时被称为“trigram”；当更大时，通常直接由的值来表示，例如“four-gram”，“five-gram”等等。

我们通常会考虑一个由互相不独立的随机变量组成的序列，序列中的每个变量的值依存于它前面的项。在实际的生产运用中，很多系统如果要预测将来的状态，就是依据现在状态，因为在很多情景下，我们并不需要用到所有过去的状态。N 元语法模型便是建立在这一共识上。

表 3.1 不同领域中的 n 元语法示例

领域	单位	示例	一元语法	二元语法	三元语法
马尔可夫链阶数	\	\	0	1	2
蛋白质测序	氨基酸	C-G-L-S-T	C, G, L, S, T	C-G, G-L, L-S, S-T	C-G-L, G-L-S, L-S-T
DNA 测序	碱基对	ATTCGA	A, T, T, C, G	AT, TT, TC, CG	ATT, TTC, TCG
计算语言学	单词	to be or not to be	to, be, or, not, to, be	to be, be or, or not, not to, to be	to be or, be or not, or not to, not to be

3.1.2 模型的形式描述

假设 $\mathbf{X}=\{X_1, X_2, X_3, \cdots, X_T\}$ 为一个基于马尔科夫链的随机序列，$\mathbf{S}=\{S_1, S_2, S_3, \cdots, S_N\}$ 为

包含 *xi* 所有取值的状态空间。下面将分别讨论基于一阶马尔科夫链、基于二阶马尔科夫链和三阶马尔科夫链的 N 元语法模型的数学描述。

基于一阶马尔科夫链的 N 元语法模型，其第 $(t+1)$ 项出现与否只取决于第 t 项。则 $S_1S_2S_3\cdots S_{t+1}$ 出现的概率为：

$$\begin{aligned} & P(X_1=S_1,X_2=S_2,X_3=S_3,\cdots,X_{t+1}=S_{t+1}) \\ = \ & P(X_1=S_1)\prod_{i=1}^{t}P(X_{i+1}=S_{i+1}\mid X_1=S_1,\cdots,X_i=S_i) \\ = \ & P(X_1=S_1)\prod_{i=1}^{t}P(X_{i+1}=S_{i+1}\mid X_i=S_i) \end{aligned} \tag{3.1}$$

基于二阶马尔科夫链的 N 元语法模型，其第 $(t+2)$ 项出现与否只取决于第 $(t+1)$ 项和第 t 项。则 $S_1S_2S_3\cdots S_{t+2}$ 出现的概率为：

$$\begin{aligned} & P(X_1=S_1,X_2=S_2,X_3=S_3,\cdots,X_{t+2}=S_{t+2}) \\ = \ & P(X_1=S_1)\prod_{i=1}^{t}P(X_{i+2}=S_{i+2}\mid X_i=S_i,X_{i+1}=S_{i+1}) \end{aligned} \tag{3.2}$$

同理，在基于三阶马尔科夫链的 N 元语法模型里，$S_1S_2S_3\cdots S_{t+3}$ 出现的概率为：

$$\begin{aligned} & P(X_1=S_1,X_2=S_2,X_3=S_3,\cdots,X_{t+3}=S_{t+3}) \\ = \ & P(X_1=S_1)\prod_{i=1}^{t}P(X_{i+3}=S_{i+3}\mid X_i=S_i,X_{i+1}=S_{i+1},X_{i+2}=S_{i+2}) \end{aligned} \tag{3.3}$$

从理论上来说，当 n 越大，则模型刻画得越准确，更贴合真实数据，预测的正确率也越高。但随着 n 的增加，计算量将成指数级增长。实际生产运用中，考虑到计算效率和准确度之间的平衡，较多使用的是 bigram 和 trigram。

3.1.3 模型的性质

举例 n 为 1 时的 N 元语法模型属性：

1）有限视野性：

$$P(X_{t+1}=S_k\mid X_1,\cdots,X_t)=P(X_{t+1}=S_k\mid X_t) \tag{3.4}$$

2）时间不变性：

$$P(X_{t+1}=S_k\mid X_t)=P(X_2=S_k\mid X_1) \tag{3.5}$$

$X_1X_2X_3\cdots X_n$ 为一个一阶马尔科夫链，随机转移矩阵 **A** 可以描述该马尔科夫链：

$$a_{ij}=P(X_{t+1}=S_j\mid X_t=S_i) \tag{3.6}$$

其中，$a_{ij} \geq 0, \forall i, j$，并且$\sum_{j=1}^{N} a_{ij} = 1, \forall i$。令$\pi_i = P(X_1 = S_i)$，则有$\sum_{j=1}^{N} \pi_i = 1$。为了方便计算，通常指定模型以某个特定的额外状态$S_0$作为起始状态，然后利用记录在$\pi$中的概率，用矩阵 **A** 存储转移状态。

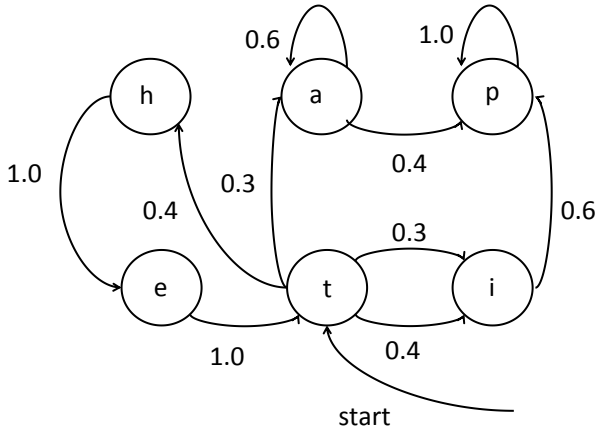

图 3.1 一个马尔科夫模型的例子

3.1.4 核心算法及优化思路

在 MM 中，状态可以看做是一串被观察的序列，也可以看做是关于时间的随机过程，即为可视马尔可夫模型（英文全称 Visible Markov Model, 英文简称 VMM）。隐马尔科夫模型（英文全称 Hidden Markov Model, 英文简称 HMM）中的状态无法被观察，可以被观察的是观察值、不可视状态的概率函数。在 HMM 中，观察值是关于状态的随机过程，而状态是关于时间的随机过程，因此 HMM 是一个双重随机过程。

以天气问题举例，在某个天气状态观测记录里面，会记录每天的空气状态是干燥(Dry)，略干燥(Dryish)，微湿(Damp)还是湿润(Soggy)，而每天的天气，则有可能是晴天(Sunny)，多云(Cloudy)或雨天(Rainy)。三种不同的天气都有可能导致空气状态是干燥，略干燥，微湿或湿润，但概率不尽相同。

如表 3.2，表明在每种天气下，分别有多大的概率会得到这四种空气状态：

表 3.2 天气-空气观测状态概率放射矩阵

	Dry	Dryish	Damp	Soggy
Sunny	0.6	0.2	0.15	0.05
Cloudy	0.25	0.3	0.2	0.25
Rainy	0.05	0.10	0.35	0.50

同时我们也知道相对前一天天气而言，得到第二天的天气的概率转移矩阵，如表 3.3：

表 3.3 天气-天气隐含状态概率转移矩阵

Yesterday \ Today	Sunny	Cloudy	Rainy
Sunny	0.5	0.375	0.125
Cloudy	0.25	0.125	0.625
Rainy	0.25	0.375	0.375

求解 N 元模型的基础是隐马尔科夫模型。

令天气状态（隐含状态）为**Q**，空气状态（观测状态）为**O**，模型中的状态个数 N，状态的可以被观察到的的不同观测值 M（叶片的状态数目），转移矩阵 $\mathbf{A}=\{a_{ij}\}$，放射矩阵 $\mathbf{B}=\{b_j(k)\}$，初始状态概率分布 $\boldsymbol{\pi}=\{\pi_j\}$ (第 j 种天气的概率），整体即是五元组 $\boldsymbol{\mu}=\{N,M,\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$，简记为三元组 $\boldsymbol{\mu}=\{\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$。

依托于隐马尔科夫模型的 N 元模型，有三个基础问题：

1）评估问题：给定一个观察序列 $\mathbf{O}=O_1O_2...O_T$ 和模型 $\boldsymbol{\mu}=\{\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$，快速计算模型 μ 的条件下，观察序列 $\mathbf{O}=O_1O_2...O_T$ 的概率，即 $P(\mathbf{O}|\boldsymbol{\mu})$？

2）解码问题：给定一个观察序列 $\mathbf{O}=O_1O_2...O_T$ 和模型 $\boldsymbol{\mu}=\{\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$，快速计算在模型 μ 的条件下，最优的状态序列 $\mathbf{Q}=Q_1Q_2...Q_T$，是该状态序列最好的解释观察序列？

3）学习问题：给定一个观察序列 $\mathbf{O}=O_1O_2...O_T$，如何调整参数 $\boldsymbol{\mu}=\{\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$，使得 $P(\mathbf{O}|M)$ 最大？

针对这三大问题，均有前人设计了相对的算法来求解。评估问题的解决算法是动态时间归准算法（英文全称 Dynamic Time Warping, 英文简称 DTW），解码问题的解决算法是维特比算法（Viterbi）；学习问题的解决算法是前向后向算法（BAUM-WELCH）。

N 元语法模型实际上就是 $(n-1)$ 马尔科夫模型。模型中 $P(X_1X_2X_3\cdots X_n)$ 可完全按马尔科夫模型模型的状态转移法计算。N 元语法模型在自然语言处理中的主要用途有两

个，一个是预测下一项的状态，还有一个是则是通过观测状态序列和概率矩阵求解隐含状态序列。前者同时是直接预判下一个词语，后者通常应用于语音识别、合成等。

在一元语法模型里面，如果要预判下一个状态，只需要找到一个 S_k，使得 $P(X_{i+1}=S_k \mid X_i)$ 最大即可。如若要通过观测状态序列求解隐含状态序列，常用的算法有维特比算法。

维特比算法是一种动态规划（英文全称 Dynamic Programming，英文简称 DP）算法，它可以通过递推求得隐含的状态序列，其中该序列最有可能预测观察到的事件。这个序列叫做维特比路径(英文全称 Viterbi Path，英文简称 VP)。例如，在语音转换为文本的过程中（即语音识别），那么声音就是观察序列，而文本就是声音的隐含的状态序列。维特比算法可以从给定的声音信号序列找出最有可能的文本序列。

以天气-空气问题为例，给出 n 天的空气状态记录，推测最大可能的天气状态序列。利用维特比算法解决该问题的主要流程如下：

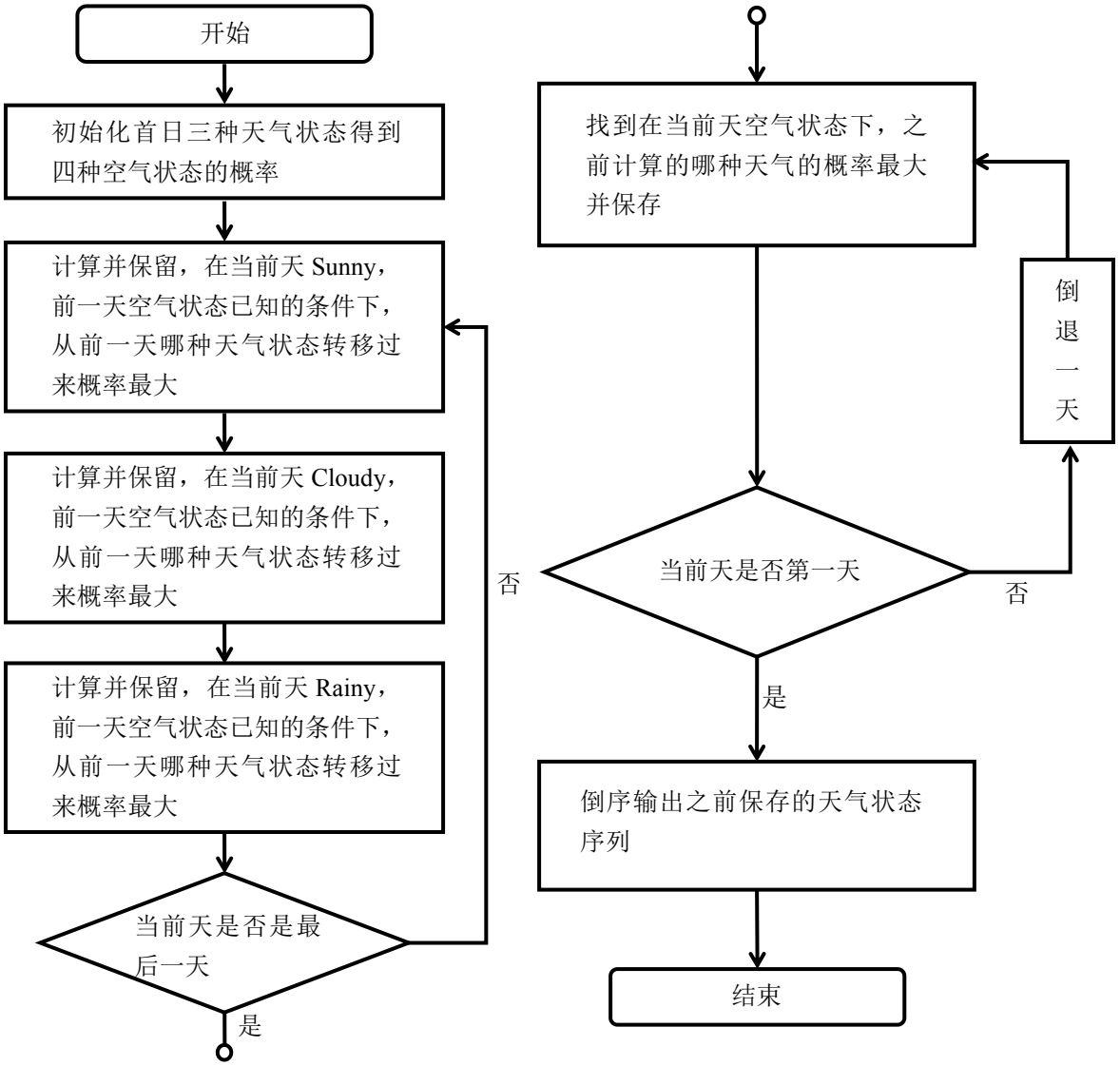

图 3.2 维特比算法求解隐含状态序列流程

给定观察值空间 $\mathbf{O}=\{O_1,O_2,O_3,\cdots,O_N\}$；状态空间 $\mathbf{S}=\{S_1,S_2,S_3,\cdots,S_K\}$，观察值序列 $\mathbf{Y}=\{Y_1,Y_2,Y_3,\cdots,Y_T\}$；转移矩阵 $\mathbf{A}$（大小为 $K\cdot K$），$\mathbf{A_{ij}}$ 存储状态 s_i 转移状态 s_j 的概率；生成矩阵 $\mathbf{B}$（大小为 $K\cdot K$），其中 $\mathbf{B_{ij}}$ 存储从状态 s_i 生成观察值 o_j 的概率；大小为 K 的初始概率数组 π，其中 π_i 存储 $x_1=s_i$ 的概率。设 $\mathbf{X}=\{X_1,X_2,X_3,\cdots,X_T\}$ 是生成观察值 $\mathbf{Y}=\{Y_1,Y_2,Y_3,\cdots,Y_T\}$ 的状态序列。

构造两个二维表 $\mathbf{T_1}$，$\mathbf{T_2}$ (大小均为 $K\cdot T$)。$\mathbf{T_1}$ 的元素存概率，$\mathbf{T_2}$ 的元素存路径。基

于上述定义的伪代码如下：

```
Input: Y = {Y_1, Y_2, ··· , Y_T}, π, A, B
Output: X = {X_1, X_2, ··· , X_T}
begin
    O ← {O_1, O_2, ··· , O_N};
    S ← {S_1, S_2, ··· , S_K};
    for each state i ∈ {1, 2, ··· K} do
        T_1[i, 1] ← π · B_{iy_1};
        T_2[i, 1] ← 0
    end
    for each observation i ∈ {2, 3, ··· T} do
        for each state j ∈ {1, 2, ··· K} do
            T_1[j, i] ← B_{jy_i} · max{T_1[k, i − 1] · A_{kj}};
            T_2[j, i] ← argmax{T_1[k, i − 1] · A_{kj}}
        end
    end
    Z_T ← argmax{T_1[k, T]};
    X_T ← S_{Z_T};
    for i ← T, T − 1, ··· , 2 do
        Z_{i−1} ← T_2[Z_i, i];
        X_{i−1} ← S_{Z_{i−1}}
    end
end
```

通过伪代码我们可以看出，算法的复杂度是$O(K \cdot T)$的。

在流程图和伪代码中，可以看到，保存在$\mathbf{T}_1$中的概率，都是通过不断连乘得到的。在这种的情况下，经过多层连续相乘，得到的概率将变得非常小。在实际利用程序计算时，基础变量类型无法存储下如此高的精度的数据。

为了解决这一问题，有三种思路：

1）使用 Java 语言的 BigDecimal 类存储，缺点：运行速度十分缓慢，编码复杂；

2）使用 Python 语言的 decimal 模块，缺点：运行速度十分缓慢；

3）对计算公式进行变形，避开要计算高精度的部分，用更精炼的语言进行计算。

笔者决定从第三种办法入手。观察式子：

$$\begin{aligned} &P(X_1 = S_1, X_2 = S_2, X_3 = S_3, \cdots, X_{t+1} = S_{i+1}) \\ &= P(X_1 = S_1) \prod_{i=1}^{t} P(X_{i+1} = S_{i+1} \mid X_i = S_i) \end{aligned} \tag{3.7}$$

可以发现，全式仅有连乘，而无加减。这让我们考虑到可以用到线性转对数的方法，以e为底，这么上式可转变为：

$$\ln(P(X_1 = S_1)) + \sum_{i=1}^{t} \ln(P(X_{i+1} = S_{i+1} \mid X_i = S_i)) \tag{3.8}$$

这样就变成了一个纯累加的计算式。并且，最重要的是，经过了线性-对数变换之后，精度大大降低，使得基础数据类型也可以存储计算了几十次的连乘结果。

如下图 3.3，在天气-空气观测状态概率放射矩阵和天气-天气隐含状态概率转移矩阵都固定不变的情况下，随着天数（连乘次数）的增加，用 C++语言编写的该算法运行效率和准确率的折线图（10000 次试验取均值）：

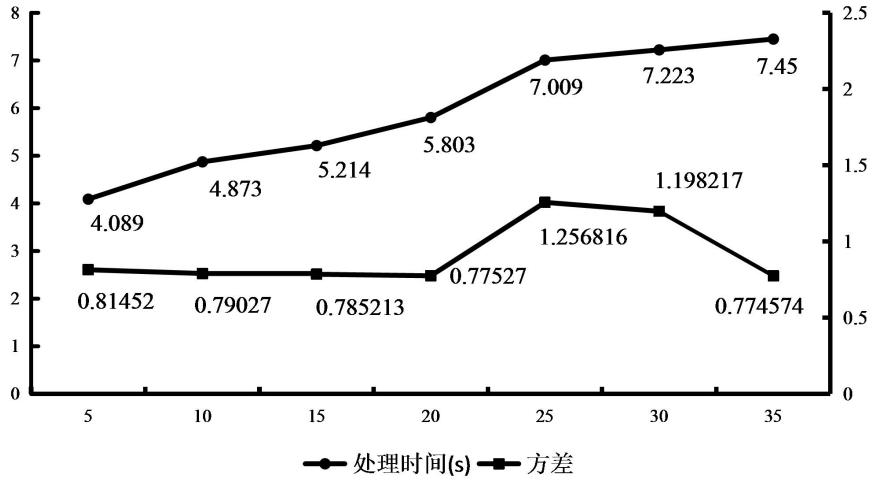

图 3.3 优化后处理时间与链长度和状态数的关系图

运行实验代码得到以上实验结果的计算机配置如下：

（1）操作系统：Windows7 64 位

（2）处理器：Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz 2.40GHz

（3）内存：4.00GB

观察图表可以发现，在状态数确定的情况下，随着天数增长（增加），计算量加大，程序的运行时间也呈线性增加，符合于之前推论的算法复杂度。并且，虽然天数增加，但程序预测结果与原数据的误差较为稳定，且比较都比较小。这些特点都表明，经过了线性-对数优化之后的维特比算法，在处理较小状态数时（较小），是一个非常优秀的算法。

3.1.5 模型评估

N 元语法模型的性能评价标准主要是混乱度 $P_e = 2^H$，其中：

$$H = -\frac{1}{N-n+1}\sum_{i=n}^{N}\log^2 P(X_i \mid X_{i-n+1}^{i-1}) \tag{3.9}$$

公式中各项符号含义分别是：n 是只模型元数，$X_1X_2X_3\cdots X_N$ 是测试用词序列，N 为测试样本大小。

如果 $P(X_i \mid X_{i-n+1}^{i-1})$ 越大，则第 n 个词和前面 $n-1$ 个词的关系性越强，$\sum_{i=n}^{N}\log^2 P(X_i \mid X_{i-n+1}^{i-1})$ 的值也就会越大，意味着 H 越大，即 P_e 越小，训练出来的模型的质量也越高。

3.2 Word2Vec 模型

Word2Vec 模型是由美国谷歌（Google）公司研制开发用于词向量训练的一款工具，特点是高效性和准确性。通常认为 Word2Vec 模型属于神经网络中的深度学习模型。Word2Vec 模型的作用，是可以通过大量的输入语料（可以是单语种的），讲语料中的词组化为一个个 K 维的词向量[14]。一个很形象的例子就是：

King – Queen ≈ Man – Woman

将短语转化为词向量，可以得到许多隐藏的信息。更值得一试 Word2Vec 的另外一个原因，便是其训练模型的高效性。在服务器性能足够的情况下，Word2Vec 一天可以训练的词向量达千亿量级。目前主流的 Word2Vec 有两种，跳元模型（英文全称 Skip-gram Model）和词袋模型（英文全称 Continuous Bag-of-Words Model，英文简称 CBOW），应用的主要基础结构是 Huffman 树。

3.2.1 CBOW 模型

词袋模型不同于 N 元模型的做法，去预测最后一个将要出现的词，而是预测中间出现的词，即 $P(W_i \mid W_{i-c},\cdots,W_{i-1},W_{i+1},\cdots,W_{i+c})$，其中 c 是上下文窗口大小。

先首先随机初始化窗口内词的 K 维向量坐标，利用 Huffman 编码将所有词做成一棵 Huffman 树，则每个词在树上都有一个独一无二的二进制编码，假令左子树节点为编号 1，右子树节点编号为 0。将这棵 Huffman 树与隐层的每个节点连边，边权随机初始化。想要使被预测词概率越大，则使其在 Huffman 树上的编码概率最大即可，概率计算办法即是从树根节点到词代表的叶节点的路径上的所有概率乘积。令上下文词为正样本，被

预测词为负样本，利用正负样本不断训练，利用梯度下降法来调整神经网络里的边权参数，得以求解出最佳预测中间词。

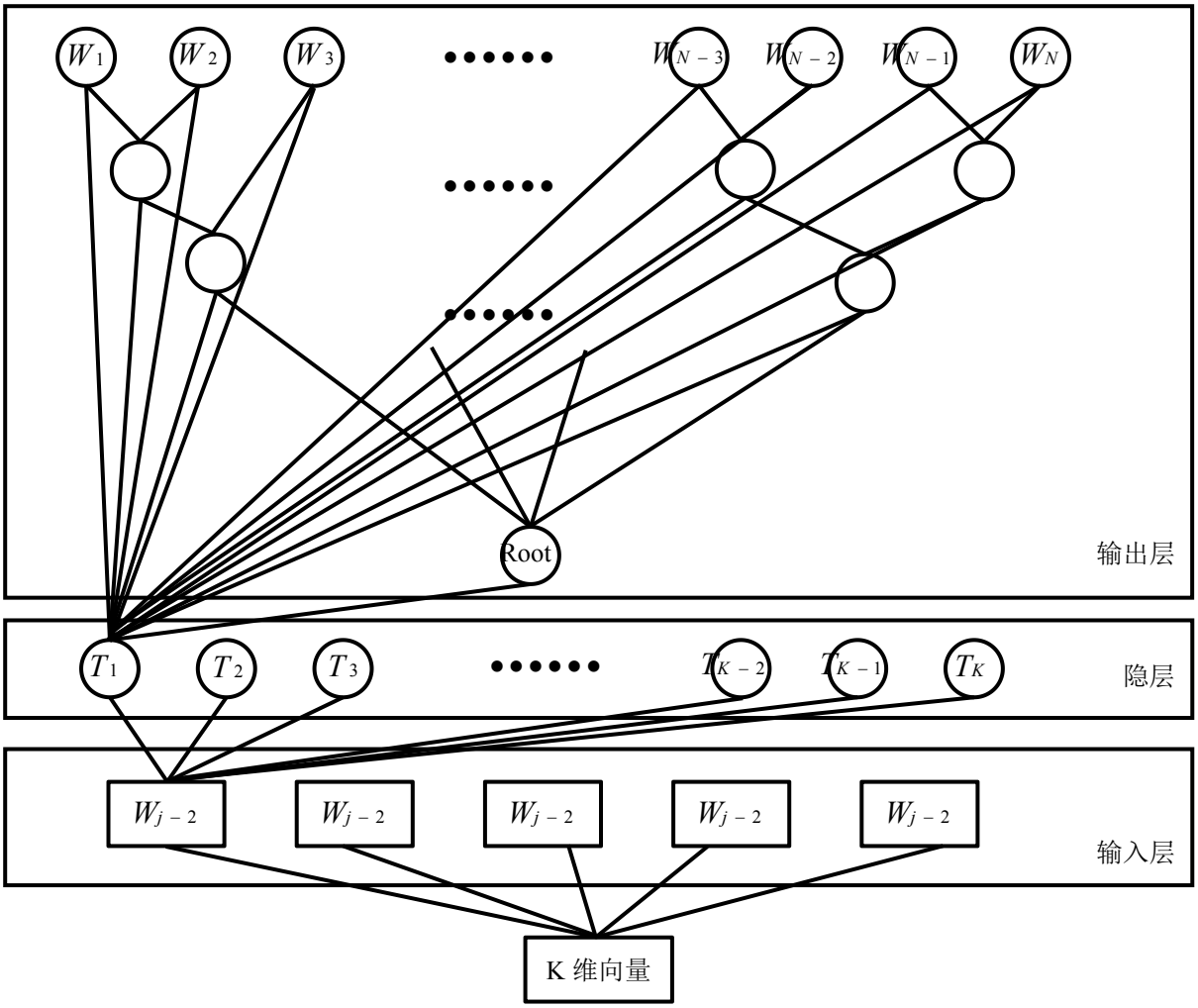

图 3.4 利用 Huffman 树构建训练 CBOW 模型的神经网络

3.2.2 Skip-gram 模型

Skip-gram 模型与 N-gram 模型有相似，但和 CBOW 模型一样，预测的是中间出现的词，但该模型的主要任务是预测 $P(W_i|W_t)$，其中 $t-c \le i \le t+c$，且 $i \ne c$，c 是上下文窗口大小[13]，这意味着与 N 元模型相比，它不仅会考虑前 c 个词，也会考虑之后 c 个词[15]。显然，当 c 越大的时候，需要预测的 $P(W_i|W_t)$ 越多，计算量越大，训练时间越长，但也越精确。

3.5 小结

本章列举了机器翻译的多个主流语言模型，详细分析了统计机器翻译中的 N 元语法模型，并实现了对 N 元语法模型关键算法的优化。之后详细论述了 Word2Vec 模型里面的 Skip-gram 模型和 CBOW 模型的原理和优缺点。

4. 面向机器翻译的语言模型分析

4.1 基于 N-gram 模型的 SMT 系统

统计机器翻译（英文全称 Static Machine Translation，英文简称 SMT）是如今机器翻译发展的基石。当今较优秀的机器翻译（MT）系统不少，到现在也仍然广泛应用中，下面简单列举其中的一些精品。

Joshua 系统：最初设计以基于层次短语的模型为主，发展至今开始支持句法模型。Joshua 系统首次实现了层次短语模型，该系统的性能也是十分稳定；

CEDC 系统：卡耐基梅隆大学（CMU）研究开发的通用解码平台，能做许多任务，目前的主要用途还是用在 MT 中，整个系统的中心方法是把问题转化为构建一个超图，然后用图论方法在超图上进行解码；

SAMT 系统：卡耐基梅隆大学开发的另一套 MT 系统。SAMT 系统是首个开源的基于句法的模型的 MT 系统，而且更令人叹服的是，他们在早期的版本就引入了 hadoop，思想非常超前；

Jane 系统：由德国莱茵-威斯特法伦工业大学（RWTH Aachen）开发，最初设计只保有基于层次短语的模型，发展至今也开始支持基于短语的系统。特点是性能较好，技术成熟；

NiuTrans 系统：由中国东北大学开发并维护，翻译模型的选择非常多，性能稳定，并且系统一直在升级换代，保持提供最新的技术支持。

除了上面例举的几个系统之外，名气最大的是 Moses 系统。它是由英国爱丁堡大学（The University of Edinburgh）、德国莱茵-威斯特法伦工业大学等多家大学联合研发的一个基于短语的 SMT 系统。Moses 系统由 C++编写编译，遵循 GNU 许可并开放源代码，支持双平台运行。Moses 系统支持市面上主流 SMT 系统模型和技术，总的来说，其具有技术全面，性能出色，历史悠久等优点。

4.2 搭建基于 N-gram 模型的机器翻译平台

在本文的实验当中，将使用 Moses 系统搭建 MT 平台。实验的主要目的为了实现汉译英系统，熟悉并掌握机器翻译流程细节，熟悉不同工具训练语言模型的方法。整个流

程可分为四个阶段：

（1）配置系统环境；

（2）处理语料数据；

（3）训练模型及调优；

（4）测试系统。

配置系统环境：本实验的系统环境包括，操作系统为 Ubuntu14.04，处理器为 Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz 2.40GHz，内存为 4.00GB。Moses 系统能适应多个 Linux 操作系统版本，不仅限于 Ubuntu。配置系统环境阶段的框架图如下图 4.1：

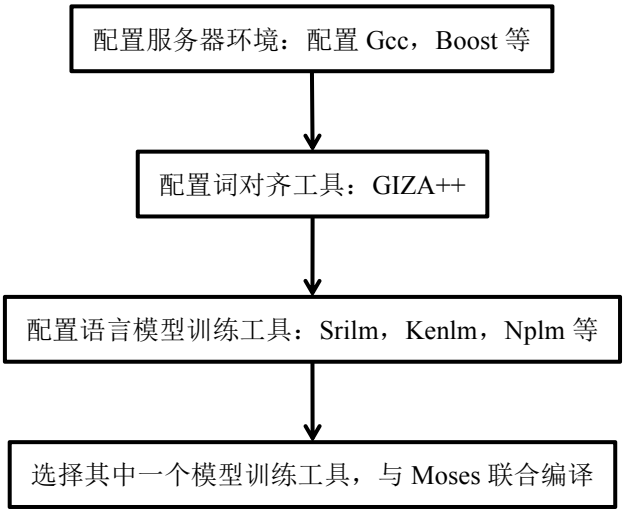

图 4.1 配置系统环境框架图

配置系统环境是必不可少的一步，Moses 系统由 C++编写编译，引用了 Boost 库的文件，没有正确配置好 Gcc 和 Boost 将导致后续操作无法进行。

GIZA++是一个 SMT 工具，它的主要用途是用来训练词对齐模型和 IBM 的五个模型，是目前被使用最广泛的统计机器翻译软件。它主要功能是用于从句子对齐的双语语料库中训练词语对齐，简而言之，就是从单纯的原文和译文句对中得到对齐的双语词对。

我们需要选用一种估值方法来构建语言模型，可以选用的工具很多，包括 Srilm，Kenlm，Nplm 等。

Srilm 可用来构建和应用统计语言模型，算是最早期的模型训练工具，历史悠久，并支持运行在双平台上。它的用途是构建和评测语言模型。构建这一步是从训练集中训练得到语言模型，将用到最大似然估计算法和平滑训练集数据的算法；而评测这一步则需要从测试集中计算上一步训练出来的语言模型的困惑度。其中 n-gram 模块是 Srilm 最早

期的、最基础的和最核心的部分，n-gram 模块包含两个细分模块：ngram 和 ngram-count，它们分别被用于计算语言模型的困惑度和统计语言模型片段。

Kenlm 同 Srilm 一样也是语言模型训练工具，它可以评估，过滤和查询语言模型，因使用的是流算法（streaming algorithms），评估速度快且可扩展。

Nplm 既可以指代神经概率语言模型（英文全称 Neural Probabilistic Language Model, 英文简称 NPLM)，同时也是一个训练 NPLM 模型的工具名称，NPLM 是一个的基于神经网络的语言模型，其中心方法和 N 元语法模型还是一样，认为第 i 项由其前面的 $N-1$ 个项决定，即依靠于 $N-1$ 阶马尔可夫链，。其网络结构如下图 4.2 所示：

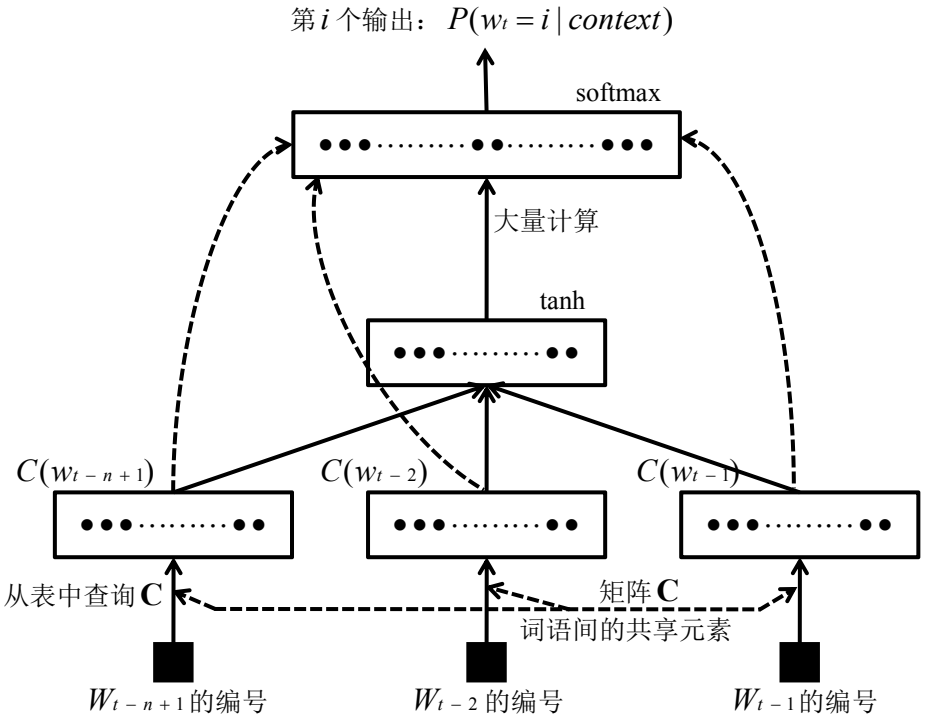

图 4.2 NPLM 模型中的神经网络结构

NPLM 这种基于神经网络的语言模型有许多好处：第一，因为 N 元语法模型需要记录每个 N 元语法片段的内容和概率，于是模型体积会随着语料的扩充而膨胀，但 NPLM 不会有这个问题；第二，N 元语法模型为了解决数据集里面的 0 概率问题，需要用到许多数据平滑方法。但 NPLM 毫不担忧，即使是语料中没出现的某个 N 元语法片段，NPLM 能对这个片段标识为非零的概率；第三，NPLM 能训练模型得到一个可以表示每个词汇的 K 维向量，并且每个维度的值都非离散的二进制值而是连续的实值，即较早形式的 Word Embedding[16]。

本次实验将用 Kenlm 工具来训练语言模型。

处理语料数据也是一个十分重要的阶段。首先收集到已经做好互译的双语句对，格式如表 4.1：

发展机器翻译，共兴人工智能
Developing Machine Translation, thriving artificial intelligence together.

表 4.1 初始双语平行语料格式

数据处理第一步是分词，即是在单词之间，或者单词和标点之间插入空格，方便后续词汇对齐等工作。编写这一步操作算法的代码是 GIZA++中的 tokenisation。有分词需求的语言大多是中文，日文等，分词后的语料格式如下表 4.2：

发展 机器 翻译 ，共 兴 人工 智能
Developing Machine Translation , thriving artificial intelligence together.

表 4.2 经过分词后的语料格式

接下来是统一大小写，虽然是简简单的一个功能，却有非常大的作用，同一大小写之后，对于英语等语种，可以大大降低数据的稀疏性。编写这一步操作算法的代码是 GIZA++中的 truecasing。再经过大小写后的语料格式如下表 4.3：

发展 机器 翻译 ，共 兴 人工 智能
developing Machine Translation , thriving artificial intelligence together.

表 4.3 经过统一大小写后的语料格式

最后一步是除去较长句、空句和明显没有对齐的句子，这一步即是清洗数据，编写这一步操作算法的代码是 GIZA++中的 cleaning。

处理语料数据阶段的框架图如下图 4.3：

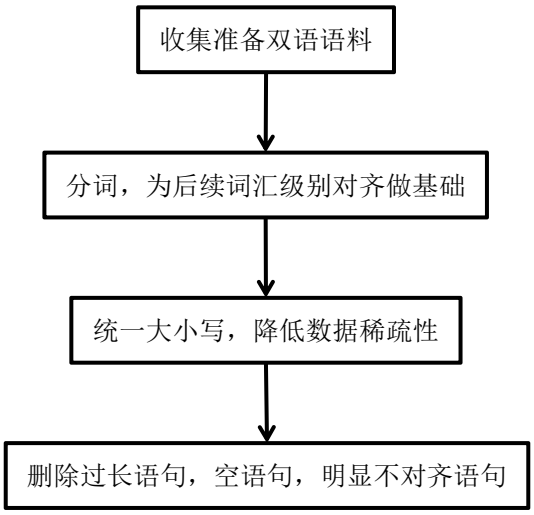

图 4.3 处理语料数据框架图

训练模型及调优阶段是搭建整个统计机器翻译平台的重头戏。为了确保最后翻译能流畅输出，本次实验训练生成的语言模型是关于目标语言（即英文）的模型。模型采用 N 元语法模型中的 trigram 模型，数据平滑算法用的是改进的 Kneser-Ney 方法。

首先需要运行 GIZA++得到对齐的词汇表。第一步是得到词汇统计表，这一步过后后会有两个统计了双语语料库中标点、词汇和词组的个数的词汇文件生成，生成的数据格式如下表 4.4：

==> corpus/en.vcb <==		
1	UNK	0
2	the	1085527
3	.	714984
4	,	659491
5	of	488315
6	to	481484
7	and	352900
8	in	330156
9	is	278405
10	that	262619

表 4.4 GIZA++生成的英文词汇统计表

同样也会的得到一个中文的词汇统计表。然后依次运行 IBM 的 5 个模型，得到词汇对齐表。 词对齐的过程由初始两个对齐的词开始，然后再一个一个加入新的匹配好的词对，直至完成全部的词汇对齐，该方法称之为生长-诊断-结束法（英文全称 Grow-Diag-Final，英文简称 GDF）[17]。

通过之前的一步之后，我们能够生成一个基于最大似然估计的词汇翻译表。表 4.5 为“有利于”这个中文单词翻译成英文的最佳翻译结果：

> grep '有利于' lex.e2c | sort -nrk 3 | head
有利于 multipolarized 0.3333333
有利于 prompting 0.2500000
有利于 preserve 0.2500000
有利于 conducive 0.2241379
有利于 physically 0.1666667
有利于 helps 0.1666667
有利于 benefit 0.1463415
有利于 beneficial 0.1428571
有利于 supervising 0.0833333
有利于 importing 0.0833333

表 4.5 中文单词“有利于”的最佳翻译结果表

从这些记录的短语翻译对中，我们可以得到一张翻译表。因为当翻译模型的体量非常大时，如果直接用翻译表而不是短语翻译表，那么在内存中是存不下来的，所以要进行这个步骤，这样就可以把短语翻译表存储在磁盘上。

在计算语料库中某一个词的相关信息的时候，对其所有的翻译候选项记录并求和，然后计算正向和反向短语翻译概率。

计算出了双向短语翻译概率表之后，还需要对其他的短语翻译结果考虑项打分，例如词汇权重、短语惩罚等。比如可以这样设置，令正向和反向翻译概率的有向求和为词汇权重，再加上十分之一分的短语惩罚。

上述过程完成后，将会得到一个 Moses 的配置文件。配置文件包含用来权衡不同的模型之间重要程度的权重信息，但都是刚初始化的，并非最优。要寻找更好的权重，需要用测试数据来对模型进行调优(tuning)[18]。

整个训练模型阶段的流程框图如下图 4.4：

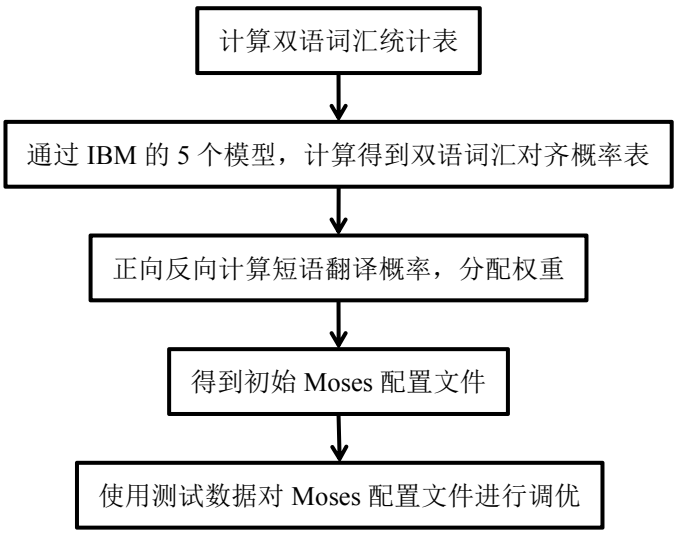

图 4.4 训练模型及调优框架图

最后一步即是测试系统。将调整参数过后的模型用 Moses 主程序运行，键入已分好词的中文待译句对，可得到翻译好的英文。下表 4.6 是中文“我爱你”的翻译测试结果：

```
Reading model/phrase-table.gz
----5---10---15---20---25---30---35---40---45---50---55---60---65---70---75---80---85---90---95--100
********************************************************************************
Created input-output object : [18.864] seconds
我 爱 你
Translating: 我 爱 你
Line 0: Initialize search took 0.000 seconds total
Line 0: Collecting options took 0.000 seconds at moses/Manager.cpp Line 141
Line 0: Search took 0.008 seconds
i love you
BEST TRANSLATION: i love you [111]    [total=-2.891] core=(0.000,-3.000,3.000,-3.646,-2.909,
-3.553,-2.383,-2.052,0.000,0.000,-1.230,0.000,0.000,0.000,-34.852)
Line 0: Decision rule took 0.000 seconds total
Line 0: Additional reporting took 0.000 seconds total
Line 0: Translation took 0.000 seconds total
```

表 4.6 Moses 机器翻译平台的测试结果

4.3 训练 Word2Vec 模型

Word2Vec 模型训练词向量速度非常快，而且在实际效果中有非常好的效果。Word2Vec 模型的作用，是可以通过大量的输入语料（可以是单语种的），讲语料中的词组化为一个个 K 维的词向量。将短语转化为词向量，可以得到许多隐藏的信息。更值

得一试 Word2Vec 的另外一个原因，便是其训练模型的高效性。在服务器性能足够的情况下，Word2Vec 一天可以训练的词向量达千亿量级[19]。本次实验利用中文数据单独训练，下表 4.7 是训练好后查看与短语“奔驰”最近的几个词和它们之间的距离：

> ./distance vectors.bin	
Enter word or sentence (EXIT to break): 奔驰	
Word: 奔驰 Position in vocabulary: 218	
Word	Cosine distance
A6	0.638032
路虎	0.631117
Q5	0.621943
宝马	0.615681
奥迪	0.599094
X6	0.59724
保时捷	0.590797
宾利	0.585854
A8	0.534328
迈巴赫	0.5299
Q7	0.526772
红旗	0.525445
A4L	0.499243
梅赛德斯	0.499034
法拉利	0.486509

表 4.7 与“奔驰”最近的几个词向量和它们之间的距离

4.4 训练 NMT 语言模型

本次实验采用了 EUREKA-MangoNMT 模型来训练模型并翻译。它是一款用于 CPU 的神经机器翻译的 C ++工具包。该工具包扩展了仅单向、无注意、无馈入输入的简单的 LSTM 编码器-解码器，其中可以用最大似然损失和噪声对比估计，实现了双向 LSTM 对输入序列进行编码，将注意输出连接到下一个解码器 LSTM 节点的馈送输入机制[20]，并支持不同培训时期的训练数据洗牌。

下表 4.8 是由 EUREKA-MangoNMT 模型训练得到的测试结果：

Source	:请 带 我 到 医院
Hyp0	:please show me to the hotel Sequence probability: 2.84421e-05

Hyp1	:please give me to the hotel Sequence probability: 2.78683e-05
Hyp2	:please give me the way to japan Sequence probability: 2.31874e-05
Hyp3	:please show me the way to japan Sequence probability: 2.11663e-05
Hyp4	:please tell me how to get to japan Sequence probability: 3.76646e-06
Hyp5	:please give me the way to the hotel Sequence probability: 2.6129e-06
Hyp6	:please show me the way to the hotel Sequence probability: 2.17746e-06
Hyp7	:please tell me how to get to the hotel Sequence probability: 9.1961e-07
Hyp8	:please tell me how to get to the seat Sequence probability: 6.92794e-07
Hyp9	:please tell me how to get to the party Sequence probability: 4.72063e-07
Hyp10	:please tell me how to get to the room Sequence probability: 4.25806e-07

表 4.8 使用 EUREKA-MangoNMT 模型训练测试结果

4.5 小结

本章实现了一个基于 N 语法模型的统计机器翻译系统，并详细介绍了系统架构和各个模块的部署方式。之后还测评了基于深度学习的机器翻译效果和基于神经网络模型的机器翻译效果。

结论

面向机器翻译语言模型是当今自然语言处理的重点和难点，目前在统计机器翻译中，N 元语法模型是一个综合评价非常高的语言模型，但它有自身所固有的局限性，比如难以维护所有的上下文信息，计算量大等。

本文针对N元语法模型的应用场景，化繁为简概述了该模型的原理和核心算法步骤，并对算法本身实现了自己的优化。随后探讨并测评了其他几款市面主流的语言模型，比如基于深度学习的模型和基于卷积神经网络的模型。

最后本文构建了一个基于统计机器翻译的翻译平台，通过较小的语料库就能训练出翻译简单对话的语言模型，并对搭建翻译平台的每一个步骤都做了详细的论述，最后测评了基于深度学习的语言模型和基于卷积神经网络的语言模型的翻译效果。

参考文献

[1] 郭永辉. 英汉机器翻译系统关键技术研究[D]. 解放军信息工程大学, 2006.

[2] 翟飞飞. 基于语言结构知识的统计机器翻译方法研究[J]. 2014.

[3] 戴新宇, 尹存燕, 陈家骏,等. 机器翻译研究现状与展望[J]. 计算机科学, 2004, 31(11):176-179.

[4] 林栋彬. 名词性短语的英汉机器互译研究[D]. 延边大学, 2014.

[5] 苏艳霞. 英汉双向未登录词翻译方法研究[D]. 复旦大学, 2012.

[6] 王长胜. 基于实例的汉英机器翻译研究与实现[D]. 中国科学技术大学, 2002.

[7] 郑逢斌. 关于计算机理解自然查询语言的研究[D]. 西南交通大学, 2004.

[8] 周磊. 基于实例英汉翻译系统研究与实现[D]. 电子科技大学, 2008.

[9] 梁华参, 赵铁军. 统计机器翻译中双语语料的过滤及词对齐的改进[J]. 智能计算机与应用, 2013, 3(4):10-13.

[10] 杨南. 基于神经网络学习的统计机器翻译研究[D]. 中国科学技术大学, 2014.

[11] 付航. 面向维汉机器翻译的平行语料库的研究与实现[D]. 西安理工大学, 2012.

[12] Och F J. Minimum error rate training in statistical machine translation[C]// Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2003:160-167.

[13] 张丽云. 英汉机器翻译系统自动评测方法的研究与实现[D]. 北京工业大学, 2006.

[14] Goldberg Y, Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method[J]. Eprint Arxiv, 2014.

[15] N. Shazeer, J. Pelemans, and C. Chelba. Sparse non-negative matrix language modeling for skip-grams[c]// in Proc. INTERSPEECH, 2015.

[16] Okita T. Joint Space Neural Probabilistic Language Model for Statistical Machine Translation[J]. Computer Science, 2013.

[17] Denero J, Macherey K. Model-based aligner combination using dual decomposition[C]// Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2011:420-429.

[18] Ludeña V L, Sansegundo R. Sentence selection for improving the tuning process of a statistical machine translation system[J]. Procesamiento De Lenguaje Natural, 2012,

48(5):327-333.

[19] Goldberg Y, Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method[J]. Eprint Arxiv, 2014.

[20] Greff K, Srivastava R K, Koutnik J, et al. LSTM: A Search Space Odyssey[J]. IEEE Transactions on Neural Networks & Learning Systems, 2016, PP(99):1-11.

致谢

在算法设计和论文撰写的过程中，我一直得到我的导师吴建辉老师的耐心指点。他在繁忙的教学、科研工作之余给了我和同组的同学们许许多多细致而耐心的指导、宝贵且实用的建议，以及许许多多的关心和帮助。这使我得以顺利完成了本课题。吴老师严谨的治学态度、忘我的工作热情、平易近人的学者风范和正直的为人，使我受益匪浅，使我学到了许多书本上无法学到的知识。因此我首先要诚挚地感谢我们的吴老师！

其次，我要感谢和我同一组的同学们，与他们进行的交流，从他们那里学习到的知识和经验，使我获益多多。在他们真诚无私的帮助下，我的设计才得以顺利进行，在此也向他们表示衷心的感谢！

再次，我要感谢我的父母。没有他们的奉献和教导，就没有我的大学生涯。是他们让我顺利地完成了本科学业！

最后，感谢所有关心我、支持我的老师们、同学们、亲人们、朋友们！

罗雅文

2017 年 5 月 31 日