南京大學

研 究 生 毕 业 论 文

（申请硕士专业学位）

论 文 题 目 短语翻译系统中的交互翻译研究

作 者 姓 名 程善伯

学科、专业名称 计算机技术

研 究 方 向 自然语言处理

指 导 教 师 陈家骏 教授 黄书剑 助理研究员

2016 年 **5** 月 **20** 日

学　　号：**MF1333006**

论文答辩日期：**2016** 年 **5** 月 **30** 日

指 导 教 师：　　（签字）

Research on Interactive Phrase-based Machine Translation

by

Shanbo Cheng

Supervised by

Professor Jiajun Chen

A DISSERTATION SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY OF NANJING UNIVERSITY IN CANDIDACY FOR THE DEGREE OF MASTER

May 20, 2016

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：短语翻译系统中的交互翻译研究

计算机技术 专业 2013 级硕士生姓名：程善伯

指导教师（姓名、职称）：陈家骏 教授 黄书剑 助理研究员

摘　要

当今社会，不同国家之间的交流越来越频繁。为了满足不同语言之间的交流需求，跨语言的翻译迫在眉睫。利用人工进行翻译效率相对低下，并且非常昂贵，所以自动化翻译技术（机器翻译技术）的研究已经成为一个非常重要的研究方向。在长期的发展后，统计机器翻译逐渐成为主流。

虽然经历了飞速发展，但是由于自然语言的复杂性，统计机器翻译的翻译质量仍不完美，用户还需要通过人机交互来进一步提高翻译质量。交互式机器翻译是一种将用户与机器翻译技术相结合，通过人机交互提升翻译质量、提升用户翻译效率的技术。在机器翻译仍不完美的今天，交互式机器翻译技术的需求非常大。当前主流的交互式翻译系统中，用户只能从左到右地修正翻译错误，导致用户可提供的信息相对少。这就限制了机器翻译系统利用用户提供的信息来提高翻译质量的能力。

在本文中，我们针对目前的交互式翻译系统的弱点做出改进。首先，提出了一种新型交互框架及该框架下的限制解码算法，该框架改变自左向右的修正操作，让用户可以修正任意位置的翻译错误；系统结合人机交互信息，使用限制解码算法进行解码，生成更优翻译。其次，提出了基于新型交互框架的自动建议模型，自动为用户的操作提供建议。再次，扩展了新型交互框架，以减少翻译系统的翻译调序错误。最后，将人机交互过程中产生的信息用于模型自适应，提升翻译系统本身的能力。

实验结果表明，本文提出的交互框架相比传统的交互框架能够更快地提升翻译质量，显著降低用户交互代价；自动建议模型可以进一步降低用户交互代价；基于新型交互框架的扩展交互方法能够根据用户提供的信息一定程度上减少翻译系统的调序错误；模型自适应方法能够根据用户历史信息更新翻译系统，提高翻译系统能力。

关键词： 短语机器翻译；交互框架；解码；自动建议；框架扩展；模型自适应

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Interactive Phrase-based Machine Translation

SPECIALIZATION: Computer Technology

POSTGRADUATE: Shanbo Cheng

MENTOR: Professor Jiajun Chen

Abstract

In current society, the communication between countries is becoming more and more frequent. The demands for translation between different languages become urgent. Due to the fact that human translation is relatively inefficient and expensive, automatic translation technology (machine translation) has become an important research area. After a long period of development, statistical machine translation has been mainstreamed.

Despite the rapid development of statistical machine translation, the translation quality of statistical machine translation is still not perfect due to the complexity of natural languages. Users still need to improve translation quality through human-computer interactions. Interactive machine translation is such a technology that improves the quality of machine translation by human-computer interactions. The demand for interactive machine translation is very large because machine translation is not perfect nowadays. Users can only perform left-to-right correction when using current interactive machine translation systems, so the information provided by users is relatively few. It will limit the ability of interactive machine translation system.

In this paper, we make improvements to the weaknesses of the current interactive translation systems. Firstly, we introduced a new interactive translation framework and the corresponding constrained decoding algorithm. Users can modify translation errors at any position instead of left-to-right correction using our framework. With human-computer interactive information, the translation system can generate better translations with constrained decoding algorithm. Secondly, to further reduce human interaction, we proposed automatic suggestion models that can offer suggestions for users automatically. Thirdly, we extended the interactive translation framework to reduce translation

reordering errors. Finally, we used human-computer interactive information to adapt statistical models and improve the ability of translation system itself.

The experimental results show that our proposed framework can improve translation quality much faster than traditional framework. The human interactions can be significantly reduced by our framework, too. With the help of our automatic suggestion models, the human interactions can be further reduced. The extension of our framework can reduce the translation reordering errors based on user provided information. With the help of the historical information of human-computer interaction, our model adaptation methods can update the translation system and improve the ability of the translation system.

keywords: Phrase-based Machine Translation, Interaction Framework, Decoding, Automatic Suggestion, Framework Extension, Model Adaptation

目　　录

目　　录 ····· **v**

插图清单 ····· **ix**

附表清单 ····· **xi**

1　绪论 ····· **1**

1.1　研究背景 ····· 1

1.2　交互式机器翻译研究现状 ····· 1

1.3　论文的主要工作 ····· 3

1.4　论文的组织 ····· 4

2　背景知识 ····· **7**

2.1　引言 ····· 7

2.2　建模方法 ····· 7

2.2.1　词对齐 ····· 8

2.2.2　翻译模型 ····· 8

2.2.3　词汇化调序模型 ····· 10

2.2.4　语言模型 ····· 11

2.3　参数训练 ····· 11

2.4　解码 ····· 12

2.4.1　解码流程 ····· 12

2.4.2　短语翻译中的柱搜索 ····· 13

2.5　自动评价指标 ····· 17

2.6　本章小结 ····· 18

3　交互翻译框架 ····· **19**

3.1　引言 ····· 19

3.2　选择 -修正交互框架 ····· 20

3.2.1 选择 -修正交互翻译系统 ····· 20
3.2.2 选择 ····· 21
3.2.3 修正 ····· 22
3.2.4 解码器和模型自适应 ····· 22
3.3 实验及结果分析 ····· 23
3.3.1 实验配置 ····· 23
3.3.2 实验方法 ····· 24
3.3.3 理想环境下翻译质量提升 ····· 24
3.3.4 通用环境下翻译质量提升 ····· 26
3.4 本章小结 ····· 27

4 自动建议模型 ····· 29
4.1 引言 ····· 29
4.2 自动建议模型 ····· 29
4.2.1 选择建议模型 ····· 30
4.2.2 修正建议模型 ····· 32
4.3 实验及结果分析 ····· 33
4.3.1 实验配置 ····· 33
4.3.2 实验方法 ····· 34
4.3.3 自动建议模型的分类表现 ····· 34
4.3.4 自动建议模型的翻译表现 ····· 35
4.4 本章小结 ····· 36

5 交互框架的扩展 ····· 37
5.1 引言 ····· 37
5.2 选择 -修正框架的不足 ····· 37
5.3 限制翻译片段交互方法 ····· 39
5.3.1 限制片段 ····· 39
5.3.2 限制片段交互流程 ····· 39
5.3.3 选取限制片段 ····· 40
5.3.4 解码器和模型自适应 ····· 41
5.4 实验及结果分析 ····· 43
5.4.1 实验配置及实验方法 ····· 43

5.4.2 限制片段交互方法对翻译质量的影响 …… 44
5.5 本章小结 …… 45

6 模型自适应 …… 47
6.1 引言 …… 47
6.2 典型方法及其不足 …… 47
6.3 自适应流程及方法 …… 48
6.3.1 补充翻译模型 …… 49
6.3.2 翻译模型插值 …… 50
6.3.3 翻译模型修正 …… 51
6.3.4 模型插值与模型修正相结合 …… 52
6.4 实验及结果分析 …… 53
6.4.1 实验配置及实验方法 …… 53
6.4.2 模型自适应对翻译质量的影响 …… 53
6.4.3 模型自适应方法对比 …… 57
6.5 本章小结 …… 58

7 总结与展望 …… 59
7.1 工作总结 …… 59
7.2 未来工作 …… 60

参考文献 …… 61

致　谢 …… 67

附录 …… 69

学位论文出版授权书 …… 71

插图清单

2-1 词对齐示例 …… 8
2-2 短语翻译过程示例 …… 13
3-1 PRIMT 框架流程图 …… 21
5-1 限制翻译片段流程图 …… 40
6-1 模型自适应流程图 …… 49

附表清单

3-1 使用 L2R 框架和 PR 框架修正中英翻译的例子 …… 20
3-2 理想环境下的交互翻译结果…… 25
3-3 通用环境下的交互翻译结果…… 26

4-1 选择建议模型使用的特征 …… 31
4-2 修正建议模型的特征…… 33
4-3 自动建议模型的分类器表现…… 35
4-4 随机选择和使用自动建议模型下的翻译质量变化情况 …… 36

5-1 使用选择 -修正交互框架进行交互的例子 …… 38
5-2 限制翻译片段交互的例子 …… 39
5-3 限制翻译片段交互翻译结果…… 44

6-1 补充翻译模型自适应方法对翻译质量的影响…… 53
6-2 模型插值自适应对翻译质量的影响 …… 54
6-3 模型修正自适应对翻译质量的影响 …… 55
6-4 模型插值与模型修正结合自适应对翻译质量的影响…… 56

第一章　绪论

1.1　研究背景

在全球交流日益频繁的今天，人们之间的交流不可避免地需要对各种语言进行互相翻译。人工翻译成本高，效率较低，难以满足大规模的翻译需求，使得人们对自动翻译技术的需求与日俱增，机器翻译技术的发展是大势所趋。

机器翻译（Machine Translation）是自然语言处理领域，以及人工智能领域中非常重要的研究方向。经过长期的发展，目前主流的机器翻译方法是统计机器翻译（Statistical Machine Translation, SMT），其中，基于短语的统计机器翻译，或简称为短语机器翻译（Phrase-based SMT, PBMT），的翻译效率高，在对时效性要求较高的翻译环境下扮演着重要的角色。

虽然机器翻译技术在不断发展与成熟，但是由于自然语言内在的复杂性，机器翻译的翻译质量仍然不完美，难以在对翻译质量要求高的情况下被用户直接使用。针对这样的问题，研究人员一方面致力于提升机器翻译模型的自身能力，从而提高机器翻译的质量；另一方面，通过将用户与机器翻译结合起来，利用人机交互来提高翻译质量，降低人工翻译成本。

交互式机器翻译（Interactive Machine Translation, IMT），是一种利用机器翻译与人机交互方法，提高自然语言之间的翻译效率的技术。虽然目前的交互式机器翻译取得了一定的成功，但仍然存在一些弱点。本文主要对基于短语翻译系统的交互式机器翻译系统中的交互框架、解码算法、交互框架的扩展、模型自适应等问题进行了研究，以克服当前的交互式机器翻译中存在的弱点，从而提高交互式翻译系统的能力，降低用户翻译的代价。

1.2　交互式机器翻译研究现状

由于交互式机器翻译系统对时效性要求高，所以目前主流的交互式机器翻译系统大多建立在基于短语的统计机器翻译系统的基础上，本文着重讨论基于短语的交互式机器翻译。

译后编辑系统（Post-editing, PE）[1] 是交互式机器翻译出现之前机器翻译

在实际应用中的主要形式。在 PE 系统中，对每个源语言句子，翻译系统首先给出原始的翻译，用户直接对原始的翻译进行字符串编辑，从而得到正确翻译结果。交互式机器翻译不是简单地给出翻译结果让用户直接进行编辑，而是让用户在人机交互的过程中不断更新翻译结果，以此提高翻译效率。

Foster 等人[2] 最先提出了交互式机器翻译的概念。他们首先提供了一种交互式机器翻译系统原型[2,3]。在此基础上，完成了 TransType 项目[4] 和 TransType 2 项目[5]。这些项目提供了完整的交互翻译系统，系统的核心部分是一个目标语言的生成引擎，该引擎可以在给定源语言句子和目标语言正确前缀的条件下，搜索目标语言端的正确补全形式。具体而言，在使用 TransType 工具进行交互式翻译时，给定一个待翻译源语言句子，首先需要用户输入正确翻译的首字符，利用机器翻译系统根据输入字符提供单词翻译的补全建议，用户在观察到该建议后，既可以接受，也可以继续输入下一个字符。在接下来的过程中，每当用户输入一个字符或接受一个词翻译后，系统认为当前翻译是正确的翻译前缀（Prefix），并提供接下来的翻译补全建议，持续该过程直到翻译完成。在此后的发展中，虽然建模方法、系统设计在不断发展，但自左向右进行句子补全的翻译框架作为交互式机器翻译的主流框架一直沿用至今。

Barrachina 等人[6] 提出了交互式翻译系统的评价指标，Key-stroke and Mouse-action Ratio (KSMR)。KSMR 建立在自左向右的交互框架基础上，其计算包括两部分，第一部分是 Key-stroke Ratio，其计算方法是：用户进行键盘输入的次数 c_k 与翻译标准答案（或参考译文, reference）的字符数 c_r 的比值 c_k/c_r；第二部分是 Mouse-action Ratio，其计算方法是：用户的鼠标操作的次数 c_m 与参考译文的字符数的比值 c_m/c_r。KSMR 越小，代表用户需要交互的代价越小，交互翻译系统的能力越强。

Koehn 等人[7] 提出使用短语翻译系统中的短语表进行翻译补全的方法。这项工作仍然基于自左向右的翻译补全的框架，并且正式把交互式机器翻译建立在基于短语的统计机器翻译系统上。Koehn 等人开发了基于 Web 的交互翻译系统原型 Caitra。与 TransType 不同，Caitra 每次提供的补全建议都是短语表中的短语翻译，这样的设置更大程度上利用了短语翻译系统的短语表。用户按照自左向右的顺序选择正确的翻译选项，系统根据输入前缀限制翻译路径的生成，生成补全建议。在此基础上，Koehn 等人[8] 进一步提升了交互式机器翻译的性能。提升的方法包括根据交互式机器翻译和短语机器翻译的特点来改进前缀匹配标准、后缀预测方法等，。

Spence 等人[9] 开发了 Predictive Translation Memory (PTM) 系统。该系统提供了更友好的交互界面，更丰富的交互功能，使得系统的实用性更强。PTM 系统仍然建立在短语翻译系统上，与 TransType 类似，系统根据用户输入提供补全建议。在此基础上，系统提供了词翻译查找功能、源端覆盖范围展示功能、整句补全建议等。

在自左向右进行翻译补全的框架中，很多工作中提供了各种不同的人机交互接口，提供了多种方法将预测结果呈现给用户。例如呈现整句翻译的最佳预测结果、仅提供部分词的最佳预测结果、提供部分词的多个预测结果供用户选择等。用户可以使用鼠标、键盘、触控等进行人机交互[4,6,7,9]。另外一些工作[10–12] 也不同程度地优化了翻译系统的搜索算法以提高搜索效率，并且提升了翻译引擎的能力。这些工作都不同程度地发展了交互式机器翻译。

ORTIZ 等人[13] 也提出了交互式机器翻译的在线学习方法。该方法同样基于自作向右的交互式翻译框架，并且在用户交互的过程中利用用户已经翻译过的句子来动态更新翻译系统的统计模型，从而达到提升翻译系统本身能力的目标。

虽然目前而言，交互式机器翻译主要采用自左向右进行翻译补全（或错误更正）的框架，并取得了成功，但是这种框架存在着一定的问题。在句子的翻译中，可能存在若干翻译错误，不同的翻译错误对整体翻译质量的影响程度不同。我们称对其他词或短语的翻译有较大影响的翻译错误为关键错误。修正关键错误可能对其他部分的翻译也会带来积极的影响。自左向右进行翻译补全或错误更正的框架无法直接针对关键错误进行修正，针对这一方面的改进有可能极大地提升翻译质量。

1.3 论文的主要工作

本文主要研究对象是基于短语翻译系统的交互式翻译系统。针对当前交互式机器翻译框架中的无法直接针对关键错误进行修正等问题，提出新的交互框架，并针对短语翻译系统和新的交互框架进行了扩展。具体地，我们从四个主要方面展开工作。

其一，我们提出了一种新的交互翻译框架，选择 -修正交互机器翻译框架（Pick-Revise Interactive Machine Translation, PRIMT）以及该框架下的限制解码算法。在该框架下，用户可以直接选取翻译关键错误，并根据系统提供的翻译

建议，结合自身知识对翻译错误进行修正。系统根据该信息利用限制解码算法进行解码得到新的翻译结果。

其二，在上述交互翻译框架的基础上，为了进一步提高用户交互效率，我们提出了自动建议模型（Automatic Suggestion Model）。自动建议模型可以基于用户的历史数据为用户的选择 -修正操作提供建议信息。

其三，我们在选择 -修正交互框架的基础上，针对机器翻译中存在的短语调序问题，引入了限制翻译片段的交互操作。用户结合自身知识，对机器翻译中应该作为整体被翻译的源语言片段进行限制。这一交互方式扩展了选择 -修正交互框架，减少了短语机器翻译中的调序错误。我们同时扩展了 PR 框架下的限制解码算法，使得用户提供的限制翻译片段相关信息可以被机器翻译系统所利用，生成更优翻译结果。

最后，我们提出对机器翻译模型进行动态更新的方法，引入了机器翻译模型的自适应技术。在人机交互不断进行的过程中，系统可以捕获人机交互过程产生的信息，并将这些信息利用起来，从而增强机器翻译系统自身模型的能力。

实验结果表明，本文提出的交互框架相比传统的交互框架能够更快地提升翻译质量，显著降低用户交互代价；自动建议模型可以简化用户操作，进一步降低用户交互代价；新型交互框架的扩展方法能够让用户提供更丰富的翻译信息，系统根据用户提供的信息进行限制解码，可以缓解翻译系统的调序错误；模型自适应方法能够根据用户历史信息更新翻译系统，提高翻译系统能力。

1.4 论文的组织

本文内容的组织如下：

第一章主要介绍本文研究内容的背景。主要包括机器翻译、交互式机器翻译的必要性和发展前景。其次介绍了交互式机器翻译的研究发展现状，包括若干主要交互翻译系统的发展现状。

第二章主要介绍本文主要工作中所涉及的背景知识。主要包括短语机器翻译系统中的相关内容，如翻译系统中的短语表、翻译系统的建模方法、翻译系统的解码算法、训练模型参数的方法、系统的评价指标等方面。

第三章主要介绍本文提出的选择 -修正交互翻译框架的相关内容。主要包括交互翻译框架的流程、限制解码算法等，并通过实验证明交互翻译框架和自

动建议模型能够快速提高翻译质量，降低用户交互代价。

第四章主要介绍本文提出的基于选择 -修正交互翻译框架的自动建议模型相关内容。主要包括自动建议模型的定义，建模方法等，并通过实验证明自动建议模型能够指导用户交互操作，进一步降低用户交互代价。

第五章主要介绍限制翻译片段的交互操作相关内容。主要包括交互操作的流程、扩展的限制解码算法等，并通过实验证明限制片段交互方法能够缓解翻译调序错误，提升翻译质量。

第六章主要介绍模型自适应相关的内容。主要包括模型更新的现有方法分析、本文中使用的方法，并通过实验证明模型自适应方法的能力及结果分析。

第七章主要是对本文工作的总结以及对未来工作的展望。

第二章　背景知识

2.1　引言

本章是本文主要工作展开的基础，主要介绍短语机器翻译相关的关键内容。

机器翻译的目标是要从大规模双语平行语料中学习出翻译知识用于自动化翻译。双语平行语料是源端语言与目标端语言的句对集合，每个句对的两个句子互为翻译。我们首先介绍短语机器翻译的整体流程：搭建一个短语翻译系统，首先需要利用双语平行语料训练词对齐（Word Alignment），进而在词对齐的基础上抽取翻译模型（Translation Model, TM）、词汇化调序模型（Lexical Reordering Model, LRM）等；另外，还需要在目标语言的单语语料上训练语言模型（Language Model, LM）；在模型训练结束后，使用这些模型进行参数调节（Parameter Tuning），获得最优参数后进行解码测试（Decoding）和自动评价。需要注意的是，由于参数调节需要进行解码，并进行自动评价，所以这两个步骤实际上也包含在训练过程中。

我们将在下文中从建模方法、参数训练方法、解码算法、自动评价指标等方面详细介绍短语机器翻译中的重要内容。通过对以上内容的分析介绍，我们可以更清晰地理解基于短语的统计机器翻译的工作原理，为我们的工作打下基础。

2.2　建模方法

给定源语言句子 s，统计机器翻译的目标是找到一个目标语言句子 t，使得条件概率 $p(t|s)$（即给定源语言句子 s，目标语言为 t 的概率）最大。短语机器翻译的建模方法是使用对数线性模型进行建模[14]。在对数线性模型中，条件概率 $p(t|s)$ 可用公式2-1来表示。

$$p(t|s) = \frac{exp[\sum_{m=1}^{M} \lambda_m h_m(s,t)]}{\sum_{t'} exp[\sum_{m=1}^{M} \lambda_m h_m(s,t')]} \tag{2-1}$$

其中，M 为特征维度，λ 和 h 分别为特征值（模型得分）和其对应的权重。在基于短语的统计机器翻译系统中，主要使用了翻译模型特征、语言模型特征、词汇化调序模型特征、计数特征、翻译扭曲特征等。

在此基础上，机器翻译求解最佳翻译的目标可以写成公式2-2，其中 t' 即要求解的最佳译文。

$$t' = \arg\max_t p(t|s) = \arg\max_t exp[\sum_{m=1}^{M} \lambda_m h_m(s,t)] \tag{2-2}$$

2.2.1 词对齐

训练词对齐是构建统计机器翻译系统的第一步，翻译模型、词汇化调序模型的抽取都需要依赖词对齐的结果。给定双语平行语料，词对齐的任务是通过无监督学习算法学得源语言词和目标语言词之间的翻译概率[15]。如果源端词与目标端词之间的翻译概率大于一定阈值，我们认为这两个词之间有对齐关系。

图2-1给出了一个词对齐的示例，在该示例中，源语言词（汉语）与目标语言词（英语）的对齐关系由连线表示，例如“他”与”he” 有对齐关系，“一家”与”a” 有对齐关系等。

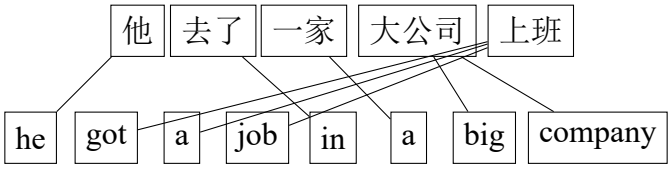

图 2-1: 词对齐示例

2.2.2 翻译模型

翻译模型，又叫短语表。短语表中存储的是双语短语对以及各个短语对之间的翻译概率[16]。在统计机器翻译中，短语指源语言句子或目标语言句子中连续的词序列，该词序列并不一定是语法意义上的短语。短语表是一个集合，该集合中每一个元素包括：源语言、目标语言短语对 (s,t)、短语对之间的短语翻译概率、词汇化翻译概率。

Koehn 等人[17,18] 提出了抽取短语翻译系统中短语表的方法。该方法由于简单易用，实验效果佳等优势逐渐成为主流方法。使用该方法抽取短语表时，要求短语对与词对齐一致（Consistent with Word Alignment）。与词对齐一致的含

义是：如果源端短语 s 中的所有词与目标端短语 t 都有对齐，并且在逆方向也满足此规则，那么称 (s,t) 与词对齐一致，可用公式2-3 表示。

$$\begin{aligned} &\forall t_i \in t : (t_i, s_j) \in Align \Rightarrow s_j \in s \\ &AND\ \forall s_j \in s : (t_i, s_j) \in Align \Rightarrow t_i \in t \\ &AND\ \exists t_i \in t, s_j \in s : (t_i, s_j) \in Align \end{aligned} \tag{2-3}$$

在抽取短语表时，需要抽取出所有满足以下原则的短语对：

1. 抽取的短语对满足短语的定义，即应是连续的词序列。
2. 抽取的短语对中不能出现没有词对齐的现象。
3. 短语对内部的任意词对齐都不能超过任意一端短语。

结合图2-1对上述原则作解释：短语对（”大公司”, ”big company”），（”去了一家”, ”in a”）构成合法的短语对，因为源端短语和目标端短语都是原始句子中的连续词序列，包含两条词对齐，且短语对内部的词对齐都不超过任意一端短语；而 (”上班”, ”got a”) 不构成合法的短语对，因为与源端词”上班”对齐的目标端词除了”got a”之外还包括”job”，超过了目标端短语，不满足第三个原则。

在抽取短语对的同时，更重要的是估计短语对之间的概率表。翻译概率表中主要包括短语翻译概率、词汇化翻译概率、词汇化调序概率等。

短语翻译概率，包括正向（源端到目标端）短语翻译概率、逆向（目标端到源端）短语翻译概率。常用的翻译概率的估计方法是极大似然估计。在计算短语对 (s,t) 的正向短语翻译概率时，首先计算出该短语对是从多少个句对中抽取的，记为 $count(s,t)$。然后统计所有合法的短语对总数，$count$；最后使用 $count$ 对 $count(s,t)$ 作归一化操作，得到正向短语翻译概率（见公式2-4）。类似地，逆向短语翻译概率 $p(s|t)$ 的计算也使用极大似然估计，只是计算方向是从目标端到源端。

$$p(t|s) = \frac{count(s,t)}{\sum_{t_i} count(s,t_i)} \tag{2-4}$$

词汇化翻译概率，包括正向、逆向词汇化翻译概率。词汇化翻译概率描述的是短语内部词语之间相互翻译的概率，也属于一种基本的平滑方法。作为翻译模型中的一部分，词汇化翻译概率对机器翻译系统的性能同样有着重要影响。给定短语对 (s,t)，在估计正向词汇化翻译概率时，同样需要基于词对齐进

行极大似然估计，具体计算方法如公式2-5所示。

$$p(t|s,a) = \prod_{i=1}^{length(t)} \frac{1}{|j|(i,j) \in a|} \sum_{\forall(i,j)\in a} w(t_i|s_j) \tag{2-5}$$

其中，s, t 分别为源端、目标端短语，a 为词对齐，$(i,j) \in a$ 表示词对齐 a 包含 (t_i, s_j) 的对齐关系。$w(t_i|s_j)$ 表示词汇化翻译概率，其计算方法如公式2-6所示。类似地，逆向短语翻译概率 $p(s|t,a)$ 的计算也使用极大似然估计，只是计算方向是从目标端到源端。

$$w(t_i|s_j) = \frac{count(t_i, s_j)}{count(s_j)} \tag{2-6}$$

2.2.3 词汇化调序模型

词汇化调序模型[19]考虑的是当前被翻译的源端短语 A 与前一个被翻译的源端短语 B 之间的相对位置关系。因为在短语机器翻译中，翻译是从左到右生成目标端的过程，每次选择一个源端短语进行翻译，并拼接到当前翻译之后，所以考虑当前源端短语 A 与前一源端短语 B 之间的相对关系相当于在翻译时考虑了挑选短语的顺序。我们通常考虑三种调序方向：单调调序（monotone, m），交换调序（swap, s），非连续调序（discontinuous, d），即 $p(orientation|s,t)$，其中 $orientation \in \{m, s, d\}$。具体地：

1. 单调调序指 A 紧接着 B 之后。
2. 交换调序指 B 紧接着 A 之后。
3. 非连续调序指 A 与 B 不连续。

在上述调序模型概念的基础上，对每个短语对的每一种调序类型，同样根据极大似然原理，统计每种情况出现的频率对调序模型进行参数估计。具体计算方法如公式2-7所示。

$$p(orientation|s,t) = \frac{count(orientation, t, s)}{\sum_{orientation'} count(orientation', t, s)} \tag{2-7}$$

最终，在实际使用中，由于短语翻译概率、词汇化翻译概率、词汇化调序概率都以短语对为基本单位，所以我们将三者统一整合到短语表中。

2.2.4 语言模型

语言模型刻画的是目标端语言句子可能出现的概率，它能够帮助机器翻译系统生成更流畅的译文。由于句子是连续的词序列，所以句子的概率可以用词的条件概率的连乘来刻画（如公式2-8）。当前主流的语言模型采用n元文法（*n*-gram）模型[20]。根据马尔科夫有限视野假设，*n*元文法模型认为当前词的概率只与当前词之前的$n-1$个词相关，即$p(w_m|w_1,w_2...w_{m-1}) = p(w_m|w_{m-n+1}...w_{m-1})$，所以在*n*元文法语言模型的限定下，给定一个长度为*m*的词序列，句子的概率可以近似地表示为公式2-9。其中，*n*-gram概率依然使用极大似然估计进行计算（公式2-10所示），其中$count(w_1,...,w_m)$表示该词序列在单语语料中出现的次数。

$$p(w_1,...,w_m) = p(w_1)p(w_2|w_1)p(w_3|w_1,w_2)...p(w_m|w_1,...,w_{m-1}) \tag{2-8}$$

$$p(w_1,...,w_m) \approx p(w_1|NULL)p(w_2|NULL,w_1)...p(w_m|w_{m-n+1},...,w_{m-1}) \tag{2-9}$$

$$p(w_n|w_1,...,w_{n-1}) = \frac{count(w_1,...,w_n)}{count(w_1,...,w_{n-1})} \tag{2-10}$$

2.3 参数训练

在第2.2节中，我们介绍了机器翻译中的对数线性模型建模方法。在对数线性模型中，每一维特征*h*都应有一个对应的权重*w*，而*w*通常需要通过参数训练的方法来获取。

Och等人[21]于2003年提出最小错误率训练（Minimum Error Rate Training, MERT），用于训练机器翻译中对数线性模型的参数。MERT需要使用一个小规模的双语平行语料（开发集，Development Set），该语料通常包括数百个互为翻译的句对。初始时，在参数空间中随机选择一组参数，翻译系统使用该组参数对开发集中所有句子进行解码并计算损失函数。接下来，选择某一维参数，固定其他参数，并以最小化损失函数为目标优化选中的参数。不断迭代以上过程直到满足终止条件（损失函数变化小于某阈值或迭代轮数超过某个阈值），从而获得最终参数。公式2-11给出了损失函数的形式。

$$L(T,R) = \sum_{i=1}^{N} L(t_i,r_i) \tag{2-11}$$

其中 T 为整个开发集的翻译结果集合，R 为整个数据集的参考译文集合，N 为开发集句对数，t_i，r_i 分别为第 i 个源语言句子的机器翻译输出结果和对应的参考译文。

公式2-12给出了最小化损失函数以求解最优参数的形式化表示：

$$\begin{aligned} \lambda_1^M &= \arg\min[\sum_{i=1}^{N} L(f(t_i, \lambda_1^M), r_i)] \\ t' = f(s_i, \lambda_1^M) &= \arg\max_{t \in c_i}[\sum_{m=1}^{M} \lambda_m h_m(s_i, t)] \end{aligned} \tag{2-12}$$

其中 h_i 和 λ_i 分别表示第 i 维特征的特征值和其对应的权重；$t' = f(s_i, \lambda_1^M)$ 表示在给定模型参数 λ_1^M 的情况下，机器翻译系统解码开发集中第 i 个源语言句子 s_i 得到的最佳翻译译文（模型得分最高的译文）；c_i 表示 s_i 对应的所有可能的翻译译文。

2.4 解码

解码（Decoding）即翻译。在短语机器翻译系统中，解码是使用翻译模型、语言模型、词汇化调序模型等搜索最佳译文的过程。与自然语言处理中很多问题类似，机器翻译的解码是一个结构化搜索问题，其搜索空间是指数级的，所以不可能穷尽整个搜索空间。我们需要一种复杂度更低的算法来搜索最佳译文。

Koehn 等人[18] 提出柱搜索算法（Beam Search）来降低搜索复杂度。柱搜索是一种启发式搜索算法，避免了全空间搜索，所以能够降低搜索复杂度。由于柱搜索不进行全空间搜索，所以会导致搜索到的最终解不是最优解，而是一个近似最优的解。

2.4.1 解码流程

我们首先以图2-2中的例子来解释短语系统的解码过程。第一行表示已经分词的源语言句子：“他 去了 一家 大公司 上班”。第二行中的每个方框里是一个源端短语，对应的下标表示该短语在解码时被选中的次序。第三行表示最终生成的翻译结果，每个方框中是一个目标端短语。系统首先选择源端短语“他”进行翻译，其对应的翻译结果为实线箭头所指向的英文部分（”he”）；

在翻译完“他”之后，系统可以按照源语言的顺序翻译“去了”，也可以不按照源语言的顺序而跳过若干词，先翻译后面的短语。在该例中，系统不按照源语言的顺序翻译，跳过若干词选择了“上班”，并将其翻译成”got a job”，进而将这个翻译连接到之前的部分翻译之后，得到”he got a job”。系统不断选择未翻译的短语进行翻译，直至将最后一个未翻译的短语“大公司”翻译为”big company” 并连接在部分翻译之后，从而生成最终的翻译结果”he got a job in a big company”。

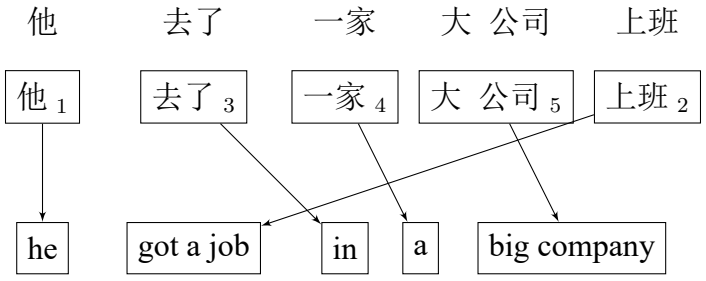

图 2-2: 短语翻译过程示例

从例子中我们可以看到，短语翻译系统的解码是从左向右生成目标端的过程，生成的部分翻译一般被称为翻译假设（Hypothesis）。假设扩展指的是翻译系统选择未被翻译的源端短语，进而从翻译模型（或称为短语表）中挑选合适的翻译选项（Translation Option），并将其翻译连接到当前翻译假设之后，同时更新翻译假设模型得分的过程。由此可知短语翻译系统的解码过程实际上是不断进行假设扩展（Hypothesis Expanding）的过程。

给定一个输入源语言句子，翻译系统首先加载各种统计模型，进而迭代地从短语表中选取翻译选项进行假设扩展。当源语言句子中所有词都被翻译后，生成一个完整假设，完整假设中模型得分最高的被认为是最优翻译假设。

2.4.2 短语翻译中的柱搜索

由于在假设扩展的过程中，可能有多个翻译选项可用于假设扩展，所以系统需要使用每一个可能的翻译选项进行假设扩展，这就导致了搜索空间的指数级增长。Koehn 等人提出了基于栈的启发式搜索算法（柱搜索，Beam Search），通过在假设扩展的同时进行大量剪枝以达到降低解码复杂度的目标。

当翻译系统在进行假设扩展时，首先需要将选取的翻译选项的目标端连接

到当前翻译假设的目标端之后，从而更新假设的目标端；其次，需要更新假设得分，并更新假设栈，同时根据假设得分进行栈剪枝。由于需要进行剪枝操作，而剪枝操作需要对剪枝的对象进行打分并将得分低的剪去，所以柱搜索算法就需要比较局部假设的得分。我们希望局部假设的比较相对公平，所以通常的做法是将翻译了相同数量的源端词的假设放在同一个栈中进行比较。在此基础上，为了进一步增强局部假设的可比较性，系统在计算局部假设的得分时，除了对数模型得分之外，通常还需要加上未来代价估计（Future Cost Estimation）得分。在进行栈剪枝时，不仅需要考虑假设得分，也要考虑假设重组的情况。我们将在下文中详细介绍上述内容。

算法 2.1 柱搜索伪代码

```
1:  Put empty hypothesis into stack 0
2:  for all stacks from 0 to n-1 do
3:    for all hypothesis in stack do
4:      for all translation options from phrase table do
5:        if valid then
6:          Create new hypothesis;
7:          Recombine with existing hypothesis if possible;
8:          Place in corresponding stack;
9:          Prune stack if too big;
10:       end if
11:     end for
12:   end for
13: end for
```

算法2.1给出了短语翻译系统中的柱搜索算法的伪代码。初始时将空假设放入 0 号栈，并开始假设扩展。对所有栈中的每一个假设，系统都从短语表中取出所有可用的翻译选项，并逐个用于假设扩展，从而生成新的假设，同时更新假设特征值、进行假设重组等。生成新的假设后，系统将新的假设加入到对应的栈中，然后根据假设得分进行栈剪枝等操作。由此可看出，柱搜索算法的复杂度为 $O(n*b*m)$，其中 n 为句子长度，b 为假设栈的大小，m 为翻译选项数量。

2.4.2.1 翻译选项

翻译选项（Translation Options），或称为翻译候选，是指源端句子中的短语、该源端词短语在短语表中对应的目标端候选，这两者构成的短语对。短语表由第2.2.2节中的方式按照词对齐信息事先抽取获得。对于源端句子 s 中的

任意短语 s_i^j[①]，只要在短语表中能够找到源端与 s_i^j 完全相同的翻译选项，就可以使用该翻译选项进行假设扩展。

由算法2.1可知，在翻译一个句子时需要遍历所有可能的翻译选项。为了降低复杂度，在抽取短语表时，通常的做法是对短语的最长长度进行限制。对于一个长度为 n 的源端词序列，我们通过将翻译选项的最长长度设置为固定值，就可以使合法翻译选项的规模降为 $O(n)$。进一步地，在假设扩展时引入扭曲限制（Distortion），限制短语候选的选取必须在当前假设的前后固定窗口大小 d 中，这就使得可用于假设扩展的合法翻译选项数降低到常数级。通常情况下，我们还会将假设栈的大小 b 设置为某个固定常数，所以柱搜索的复杂度为 $O(n)$。

2.4.2.2 假设剪枝与未来代价估计

假设剪枝是指当假设栈的大小达到一定程度时，需要控制栈大小不再增长。通常的做法是把得分低的假设丢弃，得分高的假设加入栈中，这样就可以避免栈大小的无线增长，从而避免搜索空间的指数级增长，大大降低搜索复杂度。常用的方法包括直方图剪枝（Histogram Pruning）和阈值剪枝（Threshold Pruning）。

由于需要进行栈剪枝，虽然大大降低了搜索复杂度，但也带来了显著的搜索错误。系统可能将一些合理的假设剪掉，从而使得之后的假设无法使用这些合理的假设进行扩展。上文中我们描述到，同一个栈中存放的是翻译了相同数目源端词的假设，而非翻译了相同源端短语的假设，这就导致了局部假设的比较存在不公平性。因为不同的源端短语，即使它们的词数相同，它们的翻译难度或翻译代价是不同的。例如两个源端短语分别为“你好”和“第一次”，将“你好”翻译成”hi” 的对数线性模型得分比将“第一次”翻译成”the first time” 的得分高，但是这并不能证明前者的翻译更好。针对这样的问题，Koehn 等人又提出使用对数线性模型得分加上未来代价估计进行局部假设的得分估计的方法，目标是使局部假设的假设得分估计更准确。

未来代价估计是指预计系统选取了某个翻译选项用于假设扩展后，源端句子中还未翻译的部分的翻译代价或翻译难度。与对数线性模型得分类似，未来代价估计同样需要考虑翻译概率得分，语言模型得分和词汇化调序模型得分等。首先，翻译选项的翻译模型得分可以直接从短语表中获取；其次，由于系

[①]覆盖源端第 i 到第 j 个词的短语

算法 2.2 未来代价估计算法伪代码

```
1:  for len from 1 ... n  do
2:    for s from 1 ... n+1 - len do
3:      e = s + len
4:      cost(s, e) = INFINITE
5:      if translation option (s, e) exists then
6:        cost(s, e) = score(s, e)
7:      end if
8:      for i from s ... e -1 do
9:        if cost(s, i) + cost(i + 1, e) < cost(s, e) then
10:         cost(s, e) = cost(s, i) + cost(i + 1, e)
11:       end if
12:     end for
13:   end for
14: end for
```

统无法得知未来翻译的真实情况，所以短语右边界的语言模型得分无法估计，从而只能考虑短语翻译内部的语言模型得分。最后，由于无法得知未来翻译的真实情况，所以无法得知短语翻译调序情况，所以在未来代价估计中忽略词汇化调序模型得分。我们将上述得分相加，在所有可能的估计值上选取代价最小的估计值作为未来代价估计。

未来代价估计的伪代码可以由算法2.2表示。对于所有未被翻译的源端词区间，如果存在恰好可以覆盖当前区间的翻译选项，则使用该翻译选项的得分表示该区间的未来代价；否则，将该区间划分成若干个子区间，使用能够恰好覆盖其子区间的翻译选项的得分之和作为未来代价，最终选取所有可能的未来代价的最小值作为最终未来代价估计值。

2.4.2.3 假设重组

为了进一步降低搜索空间，提高假设之间的可比较性，短语翻译系统在假设扩展时通常还需要进行假设重组（Hypothesis Recombination）。当两个局部假设 H_1 和 H_2 翻译了相同的源端部分，但是两个假设由两个不同翻译路径产生时，可能会导致其中一个假设 H_1 的局部得分比另一个假设 H_2 的得分高。当满足一定条件时，在之后的假设扩展中，经过 H_1 扩展的假设 H_1' 永远比经过 H_2 扩展的假设 H_2' 高，此时由于再扩展 H_2 已经没有意义，所以将 H_1 和 H_2 进行假设重组，丢弃得分低的假设 H_2，保留得分高的假设 H_1。

在现有的短语机器翻译系统中，需要保证两个假设 H_1，H_2 的某些特征完

全一致才能够保证当前得分更高的假设在后续扩展时得分仍然更高，具体地：

1. H_1 和 H_2 覆盖的源端相同，保证待翻译的内容一致。
2. 在使用 n 元文法条件下，H_1 和 H_2 的最后 $n-1$ 个翻译词相同，保证后续扩展的语言模型得分一致。
3. H_1 和 H_2 最后翻译的源端短语相同，保证调序模型得分一致。

2.5 自动评价指标

人工评价虽然评价效果优，但是由于效率低、昂贵等缺陷，无法使用在大规模机器翻译评测任务中。常用的机器学习问题评价指标，准确率、召回率难以被直接使用到机器翻译的译文自动评价中。因为在机器翻译系统译文评价中，单词之间的位置关系显然应该是重要的评价标准之一，但准确率和召回率都无法描述翻译单词之间的位置关系信息。合理的机器翻译评价标准不仅应该可以体现准确率、召回率，而且应该体现出翻译顺序的正确程度。

Papineni 等人[22] 于 2002 年提出 Bilingual Evaluation Understudy, BLEU。BLEU 中不仅包含了单词的准确率、召回率等信息，还包括了词序信息。由于其简单易用，与人工评价的一致性高等优势，自 2002 年以来一直是机器翻译系统译文自动评价方法中使用最广的方法之一。

BLEU 的计算方法中，首先需要计算的是 n 元文法的匹配准确率，公式 2-13 给出了 n 元 BLEU 的计算方法。

$$BLEU_n = exp[\sum_{i=1}^{n} \lambda_i log(p(i))] \tag{2-13}$$

其中，λ_i 为 i 元文法的权重，一般情况下都取值为 1，$p(i)$ 为 i 元文法的匹配率。由此可知 BLEU 得分与 n 元文法的匹配程度成正比。我们通常基于整个数据集来计算 BLEU 得分。

在公式 2-13的基础上，引入一个基于长度的惩罚项，该惩罚项表示：生成译文与参考译文的单词数越接近，则生成译文质量越高（如公式 2-14所示）。其中，r, c 分别代表参考译文的长度和翻译译文的长度。

$$\begin{aligned} BLEU_n &= BP * exp[\sum_{i=1}^{n} \lambda_i log(p(i))] \\ BP &= \min(1, exp(1 - \frac{r}{c})) \end{aligned} \tag{2-14}$$

2.6 本章小结

本章主要介绍了基于短语的统计机器翻译系统的各部分重点内容。首先介绍了短语机器翻译的整体框架与流程。在此基础上，分别介绍了短语机器翻译的建模方法，参数训练方法，解码方法等内容。

短语翻译系统通过最小错误率训练进行对数线性模型的参数调节，在开发集上，直接对翻译评价指标进行参数调节，每次调节一维参数，经过多轮迭代获得最优参数。统计机器翻译最常用的自动评价指标为 BLEU。

短语翻译系统采用了对数线性模型进行建模，可以自然地使用各种特征。短语翻译系统主要使用了翻译概率、词汇化调序概率、语言模型概率、词计数、短语计数等特征。为了减少搜索空间，降低搜索复杂度，短语翻译系统采用了一种启发式搜索算法：柱搜索解码算法。柱搜索算法通过源端词数维护搜索栈，即翻译了相同源端词数的假设置于同一个栈中。柱搜索算法的核心是假设扩展和假设剪枝。翻译系统通过将翻译选项的目标端连接到局部假设之后，并同时更新假设的各项特征值来进行假设扩展。翻译系统通过局部假设的特征值得分和未来代价估计相加进行局部假设的得分估计，并使用该得分估计进行剪枝，提高了局部假设之间的可比较性，从而降低了搜索错误。在假设扩展过程中，为了进一步减少搜索空间，翻译系统通过假设重组，丢弃无效假设。

本文中提出的交互式机器翻译相关内容均建立在短语翻译系统的基础上，特别是短语翻译系统的短语表、解码方法、模型参数训练等内容上。对短语机器翻译系统中的关键部分的理解有助于对本文的理解。

第三章　交互翻译框架

3.1　引言

传统的交互式机器翻译采用自左向右（Left-to-right, L2R）的交互框架。在第1.2节中我们描述到，在L2R交互框架中，给定一个待翻译源语言句子，用户可以自左向右进行翻译补全或错误修正，系统根据用户确认的正确翻译前缀提供后续翻译的补全建议。

虽然自左向右的交互框架取得了成功，但是该框架存在一个潜在的弱点，即该框架难以直接修正句末的关键翻译错误。关键错误是指对句子中其他词或短语的翻译质量有巨大影响的翻译错误。关键错误经常是由翻译源端短语的内在困难性导致的。Mohit等人[23]于2007年提出了使用分类器来识别难以翻译的短语（Difficult-to-translate Phrases, DTPs）的方法。这项工作证明了让真实用户翻译DTPs比让用户翻译其他短语能够带来更显著的翻译质量的提升。关键错误与DTPs有类似的性质。

当一个翻译歧义点出现在句末，并且这个翻译歧义点引起了句末的关键错误，而该关键错误又导致了句首的翻译错误时，从左到右进行翻译修正就会延迟对该关键错误的修正，从而可能导致交互效率的低下。首先修正这个关键错误可能会对之前的翻译带来很大的正面影响，这样就可以在交互翻译的过程中显著减少人工代价。

在上述背景下，本章提出了一种新的交互式机器翻译框架，在该框架下，用户可以使用两种简单的操作完成交互翻译：选择一个关键翻译错误，修正该翻译错误。我们称这种框架为选择 - 修正（Pick-Revise, PR）交互翻译框架。用户操作不局限于自左向右的框架，可以在句子的任意位置选择短语并修正其翻译，以此来提高人机交互效率。

3.2 选择 -修正交互框架

3.2.1 选择 -修正交互翻译系统

首先，我们用表3-1来解释 PR 框架与 L2R 框架[5] 之间的区别。在该例子中，我们分别使用 PR 框架和 L2R 框架进行一次交互。第一行表示已分词的中文句子和每个词对应的英文翻译，接下来每一行分别表示参考译文、基线系统的翻译结果、一次 L2R 交互的翻译结果、一次 PR 交互的翻译结果。虚线下划线表示 L2R 框架修正的错误；实线下划线表示 PR 框架修正的错误；加粗的部分表示修正了翻译错误并重新解码后，对其他部分的翻译带来的正面影响。

表 3-1: 使用 L2R 框架和 PR 框架修正中英翻译的例子

源端	南亚 各国 外长 商讨 自由 贸易区 和 反 恐 问题
	(south asian)(countries)(foreign minister)(discuss) (free)(trade zone)(and)(anti)(terrorism)(issue)
参考译文	south asian foreign ministers discuss free trade zone and anti-terrorism issues
基线	south asian foreign ministers to discuss the issue of free trade area and the
L2R	south asian foreign ministers discuss the issue of free trade area and the
PR	south asian foreign ministers **discuss free trade area and** anti-terrorism issues

对于给定的源语言句子，短语翻译系统首先生成了一个基线翻译。在 L2R 框架下，用户修正最左端的错误，将”to discuss” 修改为”discuss”，但这个修正并没有对其他部分带来正面影响。句子的翻译质量仍然未满足用户需求，所以用户需要更多操作来提升翻译质量。在 PR 框架下，用户认为” 反 恐” 是最关键的翻译错误进而选择该短语，并根据短语表将其翻译从”the” 修改为”anti-terrorism”。限制解码器（Constrained Decoder）接收到用户信息并重新翻译该句子，不仅将该短语翻译正确，而且还提升了该短语翻译附近的翻译质量（粗体部分），带来了翻译顺序的改善。与 L2R 框架相比，PR 框架能够直接对关键错误进行修正，提升了用户交互的效率。

图3-1展示了我们的 PR 交互框架的整体流程图。给定一个源语言句子 $s_1...s_n$ 和该句的初始翻译，用户首先从整句中选择一个翻译错误，然后选择一个短语表中存在的正确翻译结果。如果短语表中不存在正确的翻译，用户也可以自行输入翻译结果来替代错误的翻译结果。系统随后获取该信息，将该信息作为解码的限制，重新翻译该句。PR 框架迭代地使用限制解码器生成新翻译，限制信息来自于先前的 PR 过程。PR 过程中产生的信息还将被用于进行模型自

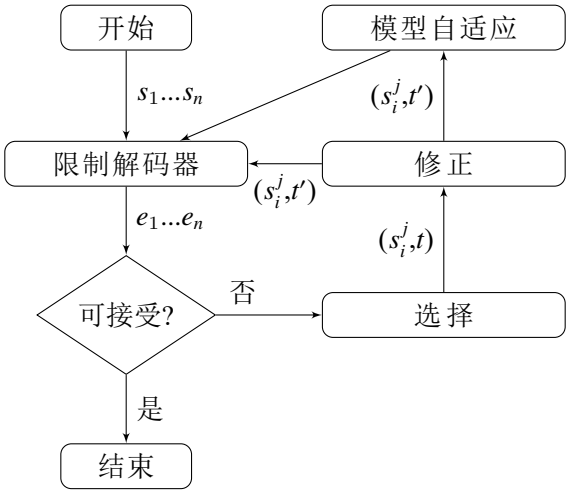

图 3-1: PRIMT 框架流程图

适应。整个交互过程一直持续到用户认为当前翻译结果已经可接受为止。我们将在下文中解释该框架中的关键内容。

3.2.2 选择

在选择（Picking）步骤中，用户选择翻译错误的短语 (s_i^j, t)（覆盖了源端第 i 个词到第 j 个词，并且翻译为 t 的短语对）进行修正。选择步骤的目标在于寻找翻译中由短语表的错误或内在的翻译歧义导致的关键错误。错误越关键，修正该错误带来的翻译质量的提升就越大[23]。

为了使得选择步骤更容易被加入到机器翻译系统中，我们限制用户只能从上一次 PR 周期的输出中选择翻译错误。如果是第一次 PR 周期，那么用户只能从基线翻译系统的输出中选择翻译错误。为了让用户可以更便捷地进行交互，在我们当前的系统中，用户既可以从源端进行选择，也可以从目标端进行选择，用户只需进行简单的鼠标点击操作即可完成交互。我们的系统将源端与目标端的对应或对齐关系进行了可视化，使得用户更容易观察。

Green 等人[24] 的实验结果证明，让真实用户进行后编辑操作可以比进行 L2R 交互翻译更快地得到可接受的翻译结果。这样的结果也表明了识别关键错误对于真实用户而言并不困难。

3.2.3 修正

在修正（Revising）步骤中，用户从短语表中选取正确的翻译选项，将 s_i^j 的翻译修正为 t'。当短语表中没有正确翻译选项时，用户也可以手动输入正确翻译。用户是否需要自己添加一个正确翻译选项由短语表的质量决定。当翻译系统由足够大的平行语料训练得到时，短语表的质量通常能够达到提供正确翻译的需求。

对于一个选择了的短语，短语表的翻译选项可以以列表的形式呈现给用户。用户只需要使用鼠标点击正确的翻译结果即可，用户也可以在输入框中另行输入一个短语翻译。

3.2.4 解码器和模型自适应

对于一个源端句子，在一个 PR 周期后，一个“选择 -修正对”（Pick-Revise Pair, PRP）就可以被系统捕获。我们使用了一个限制解码器来根据之前捕获的 PRP 作为限制来搜索到新的最佳译文。PR 框架中的限制解码算法与典型的短语翻译系统中的柱搜索解码算法类似，但是多了一个比较操作，该比较操作使解码算法忽略与 PRP 有冲突的翻译选项。

判断一个翻译选项是否与 PRP 有冲突的方法如下：

1. 若一个翻译选项的源端与某个 PRP 的源端完全一致，且该翻译选项的目标端与 PRP 的目标端不一致，则该翻译选项与 PRP 有冲突。
2. 若一个翻译选项的源端与某个 PRP 的源端有重叠，但不完全一致，则该翻译选项与 PRP 有冲突。
3. 其他情况均无冲突。

由于在限制解码算法中，所有与 PRP 有冲突的翻译选项都被系统忽略，所以整个搜索空间就比标准解码过程小了很多。在这样的设置下，我们就可以确保用户提供的 PRP 都能够被完全正确地翻译出来，并且确保整个过程可以实时进行。限制解码算法的伪代码由算法 3.1给出，其中 CONFLICT 函数即为判断当前翻译选项与 PRP 是否冲突的函数。我们可以给出 CONFLICT 函数的伪代码（算法 3.2）。

系统可以捕获到所有 PRP，在捕获到一定数量的 PRP 后进行模型自适应，重新训练模型，从而提升翻译系统本身的性能。我们将在后续章节中详细介绍本文提出的模型自适应方法。

算法 3.1 PR 框架中的限制解码算法

输入:
源端句子: $s_1...s_n$
PRP: (s_i^j, t')
用于假设扩展的翻译选项: (s_a^b, t)
输出:
新翻译 $t'_1...t'_m$

```
while Hypothesis Expanding do
  if CONFLICT((s_a^b, t), (s_i^j, t')) then
    continue;
  end if
  Expand Hypothesis Using (s_i^j, t');
end while
```

算法 3.2 选择 -修正中的冲突判断

输入:
PRP: (s_a^b, t)
当前用于假设扩展的翻译选项: (s_i^j, t')
输出:
翻译选项与 PRP 是否有冲突

```
CONFLICT((s_a^b, t), (s_i^j, t'))
if j >= a and b >= i then
  if a == i and b == j and t == t' then
    return FALSE;
  end if
  return TRUE;
end if
return FALSE;
```

3.3 实验及结果分析

3.3.1 实验配置

在所有实验中，我们使用了一个自主研发的短语翻译系统来进行中英翻译。我们将选择 -修正交互翻译框架加入到短语翻译系统中。用于训练翻译模型的双语平行语料包括 820 万句对（LDC2002E18, LDC2003E14, LDC2004E12, LDC2004T08, LDC2005T10, LDC2007T09）。我们训练了一个 5 元文法的语言模型，并使用 Modified Kneser–Ney 平滑[25]，其训练数据为 Gigaword 的 Xinhua 部分，包含了 1,460 万英文句子。我们使用 NIST02 和 NIST03 两个开发集的并集来训练翻译系统的参数。我们使用 NIST04 和 NIST05 作为测试数据。翻译质

量的自动评估使用不区分大小写的 4 元文法 BLEU。我们的系统的表现与领先的开源短语翻译系统 Moses[17] 具有可比性。

3.3.2 实验方法

因为真实的人工交互代价昂贵且耗费时间，所以我们在实验中使用模拟人工交互进行选择和修正操作。

在没有人工标注的情况下，在翻译中直接识别关键错误是一个困难的任务。我们将直接识别关键错误的任务转化为判断给定错误在其上下文下对翻译的影响的任务，以此来寻找关键错误。具体地，对于一个句子，我们依次选择基线系统输出的每一个短语，然后使用模拟修正方法（将在下文描述）修正其翻译，系统进行限制解码后，我们就可以获得新翻译。新翻译的质量与原始翻译的质量的差异就可以衡量该短语对翻译质量的影响程度。我们将提升翻译质量最明显的短语作为模拟人工选择的结果。由于 BLEU 是最常用的机器翻译评价方法，所以我们选取使得 BLEU 提升最多的短语作为模拟人工选择的结果。

在没有人工标注的情况下，给定一个源语言短语，模拟修正操作相对模拟选择操作更为直接。具体地，在所有正确的翻译选项中（关于正确的翻译选项的定义见第 4.2.2.1 节），我们选择最长的翻译选项作为模拟人工修正的结果。

在选择 -修正交互操作下，一个 PR 周期只需用户点击两次鼠标，不需要键盘输入。为了比较公平，我们对 L2R 框架使用相同的模拟修正操作，只是在使用 L2R 框架时，从左向右地选择翻译错误并修正其翻译，所以每个 L2R 周期也需要用户点击两次鼠标。同样地，我们选择最关键的错误作为后编辑（Post-Edit, PE）修正的对象。由于 PE 的过程只是字符串的修改，所以 PE 系统每次交互所需要的键盘输入次数是正确短语翻译的字符数。

3.3.3 理想环境下翻译质量提升

第一组实验用于测试选择 -修正交互翻译框架在理想环境下的性能表现。我们在使用当前系统进行强制解码可以成功解码出参考译文的句子上开展实验。强制解码是指强制翻译系统生成与参考译文完全一致的翻译结果。一个句子的参考译文可以通过强制解码产生，意味着这个句子不需要输入新单词来生成正确的翻译，这与我们的模拟实验中不要求用户输入正确翻译结果，只在短

语表中选择翻译候选的设计方法一致。

NIST04 和 NIST05 中分别有 186 句和 92 句可以成功强制解码。表 3-2中 PR*n 表示系统进行 n 轮 PR 交互操作；PE 系统对最关键的错误进行后编辑操作；L2R 系统修改最左端的错误。实验结果表明，选择并修正最关键的错误（PR*1）可以在 NIST04 (forced) 和 NIST05 (forced) 两个数据集上分别带来 +18 和 +13 BLEU 的提升，此时 KSMR 都是 2.2%。选择最左错误（L2R*1）并修正其翻译结果，同样需要 2.2%KSMR，但在两个数据集上都只带来 +5 左右的 BLEU 提升。这样的结果证明了选择关键错误在 PR 框架下非常关键。对比 L2R 方法，我们的 PR 框架有可以优先处理关键错误的优势，优先修正这些错误，BLEU 的提升相比自左向右的修正更大。

表 3-2: 理想环境下的交互翻译结果

数据	NIST04 (forced)		NIST05 (forced)	
	BLEU	KSMR	BLEU	KSMR
基线	44.59	0	41.48	0
PR*1	**63.21 (+18.62)**	2.2	**55.10 (+13.62)**	2.2
PR*2	**70.82 (+26.23)**	4.3	**63.03 (+21.55)**	4.4
PR*3	**73.99 (+29.50)**	6.5	**68.56 (+27.08)**	6.7
PR*4	**75.48 (+30.89)**	8.6	**72.20 (+30.72)**	8.9
PR*5	**76.59 (+32.00)**	10.8	**73.90 (+32.42)**	11.1
PR*6	**78.07 (+33.48)**	12.9	**75.22 (+33.74)**	13.3
PR*7	**79.27 (+34.68)**	15.1	**75.57 (+34.09)**	15.5
PR*8	**79.54 (+34.93)**	17.2	**76.02 (+34.54)**	17.8
L2R*1	49.32 (+4.73)	2.2	46.34 (+4.86)	2.2
PE*1	49.77 (+5.18)	8.3	46.81 (+5.33)	8.2

后编辑最关键错误（PE*1）需要 8% KSMR，但只带来 +5 BLEU 左右的提升。对比于无法影响周边翻译的后编辑方法，我们的 PR 框架可以重新解码获得更好翻译，从而减少用户交互。

持续进行 PR 交互能够持续提升翻译质量。在 8 次 PR 周期后（PR*8，约 17% KSMR），我们的系统对比基线系统，在两个数据集上都带来了约 +35 BLEU 的提升，达到了 75 BLEU 左右，这样的翻译结果质量已经非常高。这样

的结果证明了用户在使用选择 - 修正交互翻译系统进行交互翻译时，使用较少的交互次数即可获得高质量的翻译结果。

3.3.4 通用环境下翻译质量提升

我们也在通用环境下（NIST04 和 NIST05 整个数据集）验证选择 -修正交互翻译框架的性能表现。

表 3-3: 通用环境下的交互翻译结果

数据	NIST04		NIST05	
	BLEU	KSMR	BLEU	KSMR
基线	31.83	0	30.64	0
PR*1	**42.88 (+11.05)**	1.1	**41.47 (+10.83)**	1.1
PR*2	**48.21 (+16.38)**	2.2	**45.76 (+15.12)**	2.2
PR*3	**50.12 (+18.29)**	3.3	**48.33 (+17.69)**	3.3
L2R*1	35.61 (+3.78)	1.1	33.85 (+3.21)	1.1
PE*1	34.74 (+2.91)	4.3	34.18 (+2.54)	4.8

表 3-3给出了实验结果，表中各行的意义与表 3-2一致，唯一的区别在于第一行的 NIST04 和 NIST05 表示这两个数据集的全部句子，而不仅仅是其中可以成功强制解码的句子。

通用环境下，NIST04 和 NIST05 两个数据集的 BLEU 提升比理想环境低，导致这种现象的原因主要有两点：一是在通用环境下，句子的复杂性通常较高，短语翻译系统本身的能力限制了 PR 交互框架对 BLEU 的提升能力；二是在通用环境下，由于当前翻译系统的统计模型并不完美，所以当前的短语表可能无法涵盖所有短语的正确翻译，从而可能需要用户输入正确翻译结果。而我们的模拟交互方法并没有使用真实用户进行翻译，只模拟让用户从短语表中选择翻译选项的情况，并不模拟让输入正确的翻译结果的情况。让真实用户输入短语表中不存在的正确翻译结果可以进一步提升翻译质量。

尽管通用环境下 BLEU 提升比理想环境低，但是实验结果仍然展示了与理想环境中类似的趋势。一次 PR 周期（PR*1）可以在 NIST04 和 NIST05 两个数据集上都带来 +11 左右的 BLEU 提升。持续进行 PR 交互可以持续提升翻译质量。三次 PR 周期只需约 3.3% KSMR，却可以获得 +17 的 BLEU 提升。一次

L2R 交互（L2R*1）和一次 PE 操作（PE*1）分别在两个数据集上只带来 +3.2 和 +2.5 左右的 BLEU 提升。这样的结果证明，对比 L2R 和 PE 框架，我们的框架在通用环境下仍然有显著优势。

3.4 本章小结

本章提出了一种选择 -修正的交互式机器翻译框架（Pick-Revise IMT, PRIMT），在该交互翻译框架下，用户只需要进行两种简单的交互操作即可完成交互翻译。用户可以选择句子中任意位置的关键错误并修正其翻译，以此提高交互翻译效率。

实验结果证明优先修正关键错误而非最左翻译错误，可以使我们的框架可以更快、更有效率地提升翻译质量。用户使用选择 -修正交互翻译框架进行交互翻译，使用较少的交互操作就能获得高质量的翻译结果。进一步提升选择 -修正框架的方法有多种，主要包括：支持其他类型的交互，使用更强的统计模型等方法。

第四章　自动建议模型

4.1　引言

在第3章中，我们描述了基于短语翻译系统的选择 -修正（PR）交互翻译框架。与传统的 L2R 交互翻译框架不同，在该框架下，用户可以使用两种简单的操作（选择与修正）即可直接修正任意位置的关键错误，以此提升翻译质量和交互效率。

Ueffing 等人于 2003 年提出统计机器翻译中的置信度度量（Confidence Measure）[26]。给定一个源语言句子及对应的机器翻译系统生成的翻译结果，置信度度量要预测的是翻译结果中每个翻译单元是正确翻译的概率。在置信度度量中，翻译单元是每个目标端词。系统认为概率超过某个阈值的单词是正确的翻译，反之则是错误的翻译。

受到统计机器翻译中置信度度量的启发，为了进一步降低用户交互代价，我们在选择 - 修正交互框架的基础上，针对选择和修正两种交互操作提出了两种自动建议模型（Automatic Suggestion Model）。第一种自动建议模型是选择建议模型，另一种是修正建议模型。这两种建议模型可以分别对两种交互操作进行自动预测，并为用户提供操作建议，用户既可以接受建议也可以拒绝建议。在自动建议模型的辅助下，用户只需要进行一种操作（选择或者修正）即可完成交互翻译工作，交互操作进一步被简化，以此进一步提高人机交互效率。

4.2　自动建议模型

自动建议模型可以在用户进行两种交互操作时提供建议，用户可以接受建议也可以拒绝建议。由于选择和修正两种操作都需要从多个候选中选择一个，所以我们使用基于分类器的方法来对这两个步骤进行建模。在下面的章节中，我们将介绍把选择和修正两个步骤定义成分类问题的方法，并介绍建模时用到的特征。

4.2.1 选择建议模型

选择建议模型（Picking Suggestion Model, PSM）可以在用户进行选择操作时，为用户提供建议。因为用户进行选择操作的目标是找到在给定上下文下，对翻译质量有巨大影响的关键错误，所以选择建议模型的目标则是自动地识别那些可能是关键错误的短语，并且建议用户来选择这些短语。用户既可以接受该建议，进而修正 PSM 提供的关键错误，也可以拒绝该建议，重新选择关键错误。

需要注意的是，我们的选择建议模型是受到统计机器翻译中的置信度度量的启发而提出的。选择建议模型与置信度度量既有相似之处，又有不同之处。与置信度度量相同，我们的选择建议模型同样需要衡量的是翻译结果中每个翻译单元是正确翻译的概率，并且认为概率超过某个阈值的翻译单元是正确翻译，反之则是错误翻译。与置信度度量不同，我们的选择建议模型建立在选择-修正交互翻译框架下，其衡量的翻译单元是每个翻译短语，而非每个单词。

4.2.1.1 选择建议模型的训练

在一个源端句子的所有合法的短语中，我们需要将翻译正确的短语和翻译错误的短语区分开。

因为使用人工标记翻译错误的短语的代价高，所以我们提出了一个衡量标准，并根据这两个标准自动地区分翻译错误的短语（正例）和翻译正确的短语（负例），从而获得训练和测试数据。

因为翻译错误经常会造成翻译质量的低下，所以我们的衡量标准与翻译质量的评价指标有关。我们使用在修正了短语的翻译之后，翻译质量的变化作为衡量标准。当修正了某个短语的翻译并进行限制解码后，得到的新的翻译结果的质量较之前有提升，我们认为该短语是被错误翻译的；类似的，当得到的翻译结果的质量较之前有所下降，我们认为该短语是被正确翻译的。由于翻译质量自动评价的常用指标为 BLEU，所以我们选择那些引起 BLEU 提升或下降超过一定阈值的短语作为正例或负例。在本文中，我们将基线系统提供的翻译结果的 BLEU 得分的 10% 作为阈值。

在以上标准下，我们对数据集中每个句子的基线系统输出中的每个短语都进行模拟交互操作（第节），并将 BLEU 得分提升超过阈值的短语作为正例，BLEU 得分下降超过阈值的短语作为负例，以此来自动地获得训练数据。

4.2.1.2 选择建议模型的特征

在建模选择步骤时，我们需要两方面的信息：一方面，我们需要判断一个源端短语是否有一定的翻译难度；另一方面，我们需要判断源端短语的当前翻译结果是否正确。在此基础上，我们使用了翻译模型，语言模型，词汇化调序模型，以及计数特征，词汇化特征等进行选择建议模型的建模（见表 4-1）。

表 4-1: 选择建议模型使用的特征

类型	描述
翻译模型	基线翻译的翻译模型得分
	基线翻译的归一化翻译模型得分
	所有翻译选项的翻译得分熵
语言模型	基线翻译的语言模型得分
	每个目标端词的语言模型得分
	当前短语前/后短语的语言模型得分
	当前短语前/后边界 2 元文法的语言模型得分
词汇化调序模型	基线翻译的词汇化调序模型得分
	当前短语前后短语的词汇化调序模型得分
计数	源端/目标端词数
	当前源端短语的翻译选项数
词性标注	源端词的词性标注
	当前短语的前/后词的词性标注
词汇	源端词
	目标端词

在建模使用到的特征中，翻译模型既包含了源端信息又包含了目标端信息，翻译选项的各项翻译概率可以很好地描述翻译的可能性。翻译选项的可能性越大，正确性通常越强。归一化后的翻译模型得分能够进一步描述翻译选项的正确性。对于一个源端短语，其对应的所有翻译选项的翻译模型得分熵可以描述翻译选项的分布情况。短语的翻译模型得分熵越大，其对应的翻译选项分布越均匀，歧义也就越大，所以其翻译难度也越大。语言模型包含了目标端的信息，可以有效地衡量目标端的正确性。前后短语翻译的语言模型得分，当前短语翻译前、后边界的语言模型得分可以有效地描述短语翻译在当前上下文中

的正确性。词汇化调序模型等可以很好地描述短语的翻译顺序的正确性，结合前后短语的词汇化调序模型得分，可以更好地描述短语在当前上下文中的翻译正确性。计数特征可以描述短语及其翻译的长度信息。词性标注和词汇化特征可以更直接地描述源端短语和翻译结果。

4.2.2 修正建议模型

修正建议模型（Revising Suggestion Model, RSM）可以在用户进行修正操作时，为用户提供建议。因为用户进行修正操作的目标是在给定源端短语和上下文的情况下，选择一个正确的翻译结果，所以修正建议模型的目标是预测正确翻译结果并建议用户用预测结果替换错误翻译结果。用户同样可以接受建议，使用该结果替换错误翻译结果，也可以拒绝建议，重新从短语表中选取或自行输入一个正确翻译结果。

与选择建议模型不同，修正建议模型衡量的是给定上下文下，一个源端短语在短语表中存在的所有翻译候选是正确的概率，并且认为概率概率超过某个阈值的翻译候选是正确翻译，反之则是错误。

4.2.2.1 修正建议模型训练

对一个源端短语而言，短语表中可能存在多个翻译选项。在翻译选项集中，我们需要将正确翻译选项与错误翻译选项分开。因为使用人工标记翻译选项的代价高，所以我们提出了两个判断标准，并根据这两个标准自动地区分正确、错误翻译选项，从而获得训练和测试数据。

首先，正确的翻译选项应该是参考译文的一个子序列，该标准可以保证翻译选项本身的正确性；其次，正确的翻译选项应该与当前句对预先训练好的词对齐一致[①]，该标准可以保证不会翻译出本不该由该源端短语翻译出的词。所有不符合这两个标准中任意一个的翻译选项都被认为是错误的翻译选项。

在上述两个标准下，我们选择所有正确的翻译选项作为正例，随机抽样了相同数量的错误翻译选项作为负例。特别地，我们认为基线系统给出的翻译选项也是负例。

[①]我们使用 GIZA++[15] 进行词对齐的训练

4.2.2.2 修正建议模型的特征

因为在给定源端短语的情况下，源端信息都相同，所以特征中不需要包含源端信息，而主要关注于翻译候选的翻译质量，所以特征相比选择建议模型更简单。在此基础上，我们使用了翻译模型，语言模型，词汇化调序模型，以及计数特征，词汇化特征等进行修正建议模型的建模（见表4-2）。

表 4-2: 修正建议模型的特征

类型	描述
翻译模型	当前翻译选项的翻译模型得分
语言模型	当前翻译选项的语言模型得分
	每个目标端词的语言模型得分
	前/后边界的 2 元文法语言模型得分
词汇化调序模型	当前翻译选项的词汇化翻译模型得分
计数	目标端词数
词汇	目标端词

在建模使用到的特征中，当前翻译选项的翻译模型得分可以描述该翻译选项的可能性，可能性越大，正确性通常更强。当前翻译选项目标端词的语言模型得分可以描述该翻译选项的流畅性，句子越流畅，正确通常越强。前后边界的语言模型得分能够描述当前翻译选项在给定上下文下的正确性。翻译选项的词汇化翻译模型得分能够描述短语翻译翻译顺序的正确性。计数特征能够描述翻译选项的长度信息。词汇化特征能够直接描述翻译选项本身。

4.3 实验及结果分析

4.3.1 实验配置

本章使用的短语翻译系统与第3.3章中描述的系统一致。我们使用不区分大小写的 4 元文法 BLEU 来进行翻译质量的自动评价。

我们使用三种分类模型来对自动建议模型建模。分别为：最大熵模型（Maximum Entropy Model, ME）, 支持向量机模型（Support Vector Machine, SVM）, 神经网络模型（Neural Network, NN）。

1. 最大熵模型使用开源工具实现[27]。使用了 L-BFGS 训练方法，设置高斯先验为 2，截断值为 1，迭代 30 轮。
2. 支持向量机模型使用开源工具 LibSVM 实现[28]。使用了 RBF 核，L2 正则化，设置 $c = 128, \gamma = 0.5$。
3. 神经网络模型使用开源工具 Computational Network Toolkit, CNTK 实现[29]。使用了前馈神经网络模型（FeedForward Neural Network），设置一个隐层，该隐层包括 80 个神经元，Dropout 比例设置为 0.4，并使用 Sigmoid 函数作为激活函数。

在使用最大熵模型时，我们直接使用字符串形式表示源端和目标端词；在使用支持向量机模型时，我们用一位热码（One-Hot）表示源端和目标端词；在使用神经网络模型时，我们用预先训练的词嵌入（Word Embedding）[30] 表示源端和目标端词。

4.3.2 实验方法

首先，我们分别使用第4.2.1.1节和第4.2.1.2节中的方法，从 NIST02 和 NIST03 两个数据集的并集中自动抽取出选择建议模型和修正建议模型的训练样本；从 NIST04 和 NIST05 中自动抽取出选择建议模型和修正建议模型的测试样本，以此避免人工标注的昂贵代价。其次，我们利用抽取出的训练数据，分别训练三种分类模型，并在测试数据上分别测试三种分类模型的分类表现。最后，我们将训练好的自动建议模型应用到交互翻译任务中，并测试其在翻译任务中的表现。

4.3.3 自动建议模型的分类表现

表4-3中展示了使用不同分类器建模两种自动建议模型时的表现。第一行表示分类器和数据信息，接下来分别表示使用三种分类模型建模选择建议模型、使用三种分类模型建模修正建议模型在 NIST04 和 NIST05 两个数据集上的分类表现。

具体地，每个单元格中的三个数值分别代表准确率、召回率和 F 度量。在本实验中，准确率和召回率都基于测试集中的正例计算而得，因为只有被预测为正例的实例才可以被用于交互机器翻译系统中。需要注意的是，因为自动确定正确的翻译选项是一个较难的任务，所以当所有翻译选项都被预测为负例

时，我们保持原先翻译选项不变。

表 4-3: 自动建议模型的分类器表现

建议模型	分类器	NIST04	NIST05
选择建议模型	最大熵	0.70/0.62/0.66	0.69/0.60/0.64
	支持向量机	0.71/0.68/0.69	0.69/0.66/0.67
	前馈神经网络	0.71/0.73/0.72	0.68/0.70/0.69
修正建议模型	最大熵	0.71/0.58/0.63	0.70/0.57/0.63
	支持向量机	0.70/0.61/0.0.65	0.68/0.62/0.65
	前馈神经网络	0.66/0.67/0.66	0.65/0.65/0.65

由表中数据可知，在选择建议模型中，最大熵模型、支持向量机模型、前馈神经网络模型的表现相近，前馈神经网络稍有优势。使用前馈神经网络建模选择建议模型，在 NIST04 和 NIST05 数据集上分别可以以 F- 度量值 0.72 和 0.69 来识别关键错误；同样地，在修正建议模型中，上述三种分类模型的表现也相近，前馈神经网络同样稍占优势。使用前馈神经网络建模修正建议模型，在 NIST04 和 NIST05 数据集上分别可以以 F- 度量值 0.66 和 0.65 来识别正确翻译。因为两个任务的难度都较大，所以两个自动建议模型的 F 度量都在 0.60 和 0.70 之间是可接受的。

4.3.4 自动建议模型的翻译表现

我们也测试了在 PR 框架下使用自动建议模型进行交互翻译的表现（见表 4-4）。表中第一行表示数据信息；第二行表示基线系统在 NIST04 和 NIST05 两个数据集上的翻译表现；接下来表示随机选择、使用三种不同分类模型进行选择，然后进行模拟修正后，翻译系统在两个数据集上的表现；紧接着的是给定最关键错误，随机修正、使用三种不同分类模型进行修正后，翻译系统在两个数据集上的表现。

由表中数据可知，如果随机选择一个源端短语并进行模拟修正，BLEU 几乎无提升。相比而言，使用选择建议模型并进行模拟修正可以在两个数据集上都获得显著提升（+2 BLEU 左右）。这表明了 BLEU 提升并不来源于模拟修正的参考译文匹配长度，选择操作在 PR 框架下非常关键。选择最关键的

错误后，随机修正也无法带来 BLEU 提升，而使用修正建议模型可以提升 1.5 BLEU。

表 4-4: 随机选择和使用自动建议模型下的翻译质量变化情况

		NIST04	NIST05
	基线	31.83	30.64
	随机选择	31.92 (+0.09)	30.69 (+0.05)
选择建议模型	最大熵	33.89 (+2.06)	32.57 (+1.93)
	支持向量机	34.01 (+2.18)	32.66 (+2.02)
	前馈神经网络	34.23 (+2.40)	32.81 (+2.17)
	随机修正	31.90 (+0.07)	30.71 (+0.08)
修正建议模型	最大熵	33.62 (+1.79)	32.38 (+1.74)
	支持向量机	33.73 (+1.90)	32.42 (+1.78)
	前馈神经网络	33.77 (+1.94)	32.44 (+1.80)

使用任意一种建议模型都可以提升翻译质量，此时用户只需要进行一种操作（选择或修正）。这种情况下可能更适用于单个用户操作。诚然，使用某一种自动建议模型时，翻译质量的提升较全部使用模拟交互更小，这表明了用户的参与仍然对于翻译质量的提升至关重要。我们可以使用更好的模型、更大的数据量来提升自动建议模型的质量。

4.4 本章小结

本章提出了基于选择 -修正交互翻译框架的选择建议模型和修正建议模型。这两种自动建议模型分别可以为用户的选择操作和修正操作提供建议，用户既可以接受建议，也可以拒绝建议，以此来提高交互效率。

我们使用了基于分类器的方法来建议两个自动建议模型，并使用模拟交互方法进行两种自动建议模型的训练、测试数据的自动获取。实验结果一方面证明了自动建议模型可以达到较好的分类效果；另一方面证明了自动建议模型能够指导用户操作，获得显著的翻译质量的提升，从而能够在选择 -修正框架的基础上进一步降低用户交互代价。使用自动建议模型让不同的用户负责不同的操作，可以使用户专注于某一种操作，这也可能更适合真实用户使用。

第五章　交互框架的扩展

5.1　引言

在第3章和第4章中，我们分别描述了基于短语翻译系统的选择 -修正（PR）交互翻译框架和该框架下的自动建议模型。在该框架下，用户可以直接对任意位置的关键错误进行修正，以此来提升翻译质量。自动建议模型可以根据用户历史信息为用户提供交互操作的自动建议。

在当前的 PR 交互翻译框架中，并不是任何情况下都能够获得与参考译文完全相同的翻译结果。我们在实验过程中发现在通常环境下，如果仅要求用户从短语表中选择翻译选项，而不要求用户输入正确翻译，系统能达到的最大 BLEU 在 60 到 70 之间。翻译结果无法完全达到参考译文的原因主要有两个：一是短语表质量的不足，导致某些正确翻译选项的丢失；二是即使句子中所有短语的翻译都是正确的，但由于语言模型、调序模型存在模型错误，使得翻译系统倾向于生成错误的短语翻译顺序，从而导致了翻译质量的损失。针对第一种原因，我们可以扩大训练数据的规模以获得更优的短语表，从而缓解该问题；除此之外，我们也可以让用户输入正确翻译结果，从而解决该问题。针对第二种原因，我们可以通过扩大训练数据规模生成更优的统计模型，从而缓解该问题；除此之外，我们可以在当前的交互翻译框架中支持用户控制短语翻译顺序的交互操作，让用户决定翻译顺序。

本章主要针对第二种原因，即由于语言模型、词汇化调序模型的错误而导致的短语翻译调序错误，在 PR 交互翻译框架的基础上提出一种限制翻译片段的交互方式。限制翻译片段的交互方法作为 PR 交互框架的补充，一定程度上可以缓解 PR 交互框架无法让用户控制翻译调序的问题，从而提高交互翻译系统的能力。

5.2　选择 -修正框架的不足

我们首先用一个例子来说明 PR 交互翻译框架中可能存在的翻译顺序错误的问题。我们仅考虑短语表的质量足够高，能够提供所以短语的正确翻译，但

翻译系统倾向于选择错误的翻译顺序的情况。这样的问题在目前的框架中难以很好地被解决。

表 5-1中给出了一个使用 PR 交互框架进行交互翻译的例子，表中第一行是源端句子及其每个词对应的翻译。接下来的参考译文、基线翻译、PR*n 等的定义与表 3-1 中的定义相同。源端下划线标记的短语 P_i （表中上标为 i）表示该短语在第 i 次 PR 周期中被选择为关键错误。P_i 的原始翻译为 PR* $(i-1)$ 中无上标的下划线表示部分（基线系统为 P_0）； P_i 的修正后的翻译结果为 PR*i 中带上标的下划线部分，粗体部分表示修正后带来的正面影响。

表 5-1: 使用选择 -修正交互框架进行交互的例子

源端	然而， 以色列 的[2] 回答 无法[1] 充分扫除 美国 的 疑问 。
	(however) (israel's) (reply) (fail) (full clear) (the us) () (doubt) (.)
参考译文	however , israel 's reply failed to fully clear the us doubts .
基线翻译	however , the israeli response to the full removal of united states .
PR*1	however , the israeli response failed to[1] **fully clear** doubts . the us
PR*2	however , israel 's[2] **reply** failed to fully clear doubts . the us

首先，系统给出基线翻译，该翻译的质量并未达到用户的要求，所以需要通过人机交互提高翻译质量。在第一轮交互（PR*1）中，用户选择源端短语”无法” 作为关键错误，并将其翻译从”to the” 修正为”failed to”。系统随后得到一个 PRP，（“无法”，”failed to”），并使用算法 3.1 进行限制解码，搜索到最优翻译（PR*1 中的翻译结果）。在该轮 PR 交互后，限制解码方法使得该 PRP 周围的短语翻译质量得到了提升（” 充分扫除” 的翻译从”full removal of” 修正为”fully clear”）。在第二轮交互（PR*2）中，用户选择“以色列 的”作为关键错误，并将其翻译从”the israeli” 修正为”israel 's”，限制解码器将源语言句子重新解码后，“回答”的翻译从”response” 修正为”reply”，更符合参考译文。在第二轮 PR 交互后，所有短语翻译都已被正确翻译，但由于语言模型和词汇化调序模型倾向于错误的短语顺序，导致了翻译结果与参考译文有不同（“美国”的翻译”the us” 应紧跟在“充分扫除”的翻译”fully clear” 之后）。此时，由于所有短语的翻译都是正确的，而 PR 操作只能修改短语翻译，所以继续进行 PR 操作也无法提高翻译质量。

由上述例子我们可以看出，在当前的 PR 框架下，仍然有无法处理的问

题，这些问题一方面可以通过使用更好的统计模型来缓解；另一方面也可以允许用户适当地控制短语翻译调序来缓解。我们将在下面章节中描述允许用户控制短语翻译顺序的交互方法：限制翻译片段的交互方法。

5.3 限制翻译片段交互方法

5.3.1 限制片段

限制翻译片段交互方法需要用户提供限制片段（Constrained Span）信息。限制片段是源端句子的一个连续词序列。当用户认为 s_i^j 应作为一个整体被连续翻译，其内部的子短语不应与外部的短语之间互相调序，而基线翻译系统并没有按照这样的顺序进行翻译，此时可以通过简单的操作（例如鼠标点击、拖动等），将该信息提供给翻译系统。系统接收到的信息就是源端句子的一个片段，该片段 s_i^j 即为一个限制片段。

5.3.2 限制片段交互流程

表 5-2: 限制翻译片段交互的例子

源端	在 美国 九一一 恐怖 攻击 周年 左右 ， 东南亚 各地 的 西方 外交 使节 团 纷纷 关闭 。
参考译文	at the time around the anniversary of the 911 terrorist attacks in united states , western diplomatic missions across southeast asia have closed their doors one after the other .
基线翻译	the 11 september terrorist attacks in the united states , southeast asia around the anniversary of the western diplomatic missions have been closed .
限制片段	the 11 september terrorist attacks anniversary in the united states , southeast asia across western diplomatic missions have been closed .

我们首先用表5-2中的例子来说明使用限制翻译片段交互方法进行交互翻译的过程。第一行中是源语言句子，第二行是参考译文，第三行是基线系统译文，第四行是经过一轮限制片段交互后的译文。源端句子中的下划线部分表示用户提供的限制片段。在基线系统翻译中，系统将“周年”的翻译”anniversary”置于” 东南亚 各地” 的翻译”southeast asia around” 之后，这与参考译文的翻译顺序不符。用户认为” 九一一 恐怖 攻击 周年” 应作为整体被翻译，进而选取该片段” 九一一 恐怖 攻击 周年” 作为限制片段。系统接收到用户提供的限制片段信息，然后使用限制解码器进行解码，搜索最优翻译结果。在新的翻译结果

中，”九一一 恐怖 攻击 周年” 的翻译是连续的，系统生成了连续的目标端词序列，”the 11 september terrorist attacks anniversary”，从而使得短语翻译顺序得到了改善。用户若对当前翻译结果仍不满意，可以进一步迭代地结合使用 PR 交互操作与限制翻译片段操作来提高翻译质量。

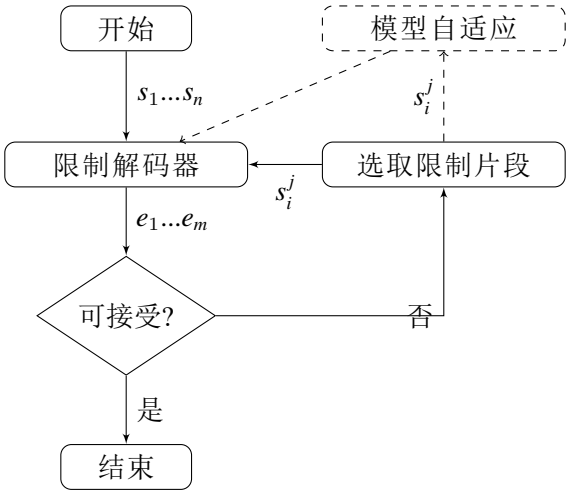

图 5-1: 限制翻译片段流程图

我们在图5-1中给出了限制翻译片段的交互流程图。与 PR 交互框架一致，限制翻译片段的交互方法也是一个不断迭代的过程，所以可以与 PR 交互框架非常自然地结合到同一个交互翻译系统中。

对于一个源端句子 $s_1...s_n$，限制翻译片段交互方法迭代地使用限制解码器生成新翻译。限制信息信息还可被用于进行模型自适应。由于限制片段交互方法与 PR 交互方法相同，都是迭代的过程，所以用户可以结合两种交互方法进行交互翻译。整个交互过程一直持续到用户认为当前翻译结果已经可接受为止。我们将在下文中解释限制片段交互方法中的关键内容。

5.3.3 选取限制片段

在选取限制片段时，用户需要选择源语言句子的一个连续词序列。选取限制片段的目标是寻找翻译中应被作为整体被连续翻译，但实际解码中并未被连续翻译的词序列。用户选择合适的限制片段能够带来翻译顺序的显著提升。

为了让用户可以更便捷地进行交互，在我们当前的系统实现中，用户进行限制翻译片段交互时，每次需要四次点击操作：用户首先鼠标点击一个指示器，表示开始限制片段交互，且下一次鼠标点击动作点击的应是限制片段的起

始词；用户接下来点击限制片段的起始词；下一步，用户鼠标点击另一个指示器，表示下一次鼠标点击动作点击的是限制片段的结束词；最后，用户点击限制片段的结束词完成交互操作。这样的设置使得用户只需进行简单的鼠标点击操作即可完成交互。

5.3.4 解码器和模型自适应

对于一个源端句子，在进行一次限制片段交互后，一个“限制片段”就可以被系统捕获。我们使用了一个限制解码器来利用之前捕获的限制片段作为限制进行搜索，从而获得新的最佳译文。限制翻译片段中的限制解码算法与典型的短语翻译系统中的解码算法类似，但是在给定限制片段的情况下，限制解码器会忽略与限制片段有冲突的翻译选项，从而导致搜索顺序的改变。

在限制片段交互方法中，判断翻译选项与限制片段是否有冲突的方法如下：

1. 如果当前的局部假设 H_a^b（H_a^b 表示覆盖源端第 a 个词到第 b 个词的翻译假设）的源端与限制片段 s_i^j 无交集，则所有翻译选项均与限制片段无冲突。
2. 如果限制片段 s_i^j 已经被完全翻译，则在接下来的假设扩展过程中所有翻译选项均与限制片段无冲突。
3. 如果当前假设 H_a^b 的源端与限制片段 s_i^j 有交集，且 s_i^j 还未被完全翻译，则所有超出区间 $[i, j]$ 的翻译选项均与限制片段有冲突。

算法 5.1 限制片段中的限制解码算法

输入:
源端句子: $s_1...s_n$
限制片段: s_i^j
输出:
新翻译 $t'_1...t'_m$

```
1: while Expanding H_a^b Using Translation Option (s_k^l, t) do
2:   if CONFLICT((s_k^l, t), H_a^b, s_i^j) then
3:     continue;
4:   end if
5:   Expand Hypothesis H_a^b Using (s_k^l, t);
6: end while
```

算法5.1给出了限制翻译片段中的限制解码算法伪代码，其中 CONFLICT 函数用于判断翻译选项与限制片段是否有冲突。我们同样可以给出 CONFLICT

算法 5.2 限制片段中的冲突判断

输入:
限制片段: s_i^j
当前扩展的假设: H_a^b
当前用户假设扩展的翻译选项: (s_k^l, t)
输出:
当前翻译选项与限制片段是否冲突

```
1: CONFLICT((s_k^l, t), H_a^b, s_i^j)
2: if j ≥ a and b ≥ i then
3:   if ∃idx ∈ [i, j] and CoverageVector[idx] == 0 then
4:     if k > j or l < i then
5:       return TRUE;
6:     end if
7:   end if
8: end if
9: return FALSE;
```

算法 5.3 PR 交互方法与限制片段交互方法相结合的冲突判断算法

输入:
当前翻译选项 (s_k^l, t)
当前局部假设 H_a^b
当前限制片段 s_i^j
当前 PRP (s_x^y, t')
输出:
当前翻译选项与限制片段、PRP 是否冲突

```
1: CONFLICT((s_k^l, t), (s_x^y, t'), H_a^b, s_i^j)
2: if j ≥ a and b ≥ i then
3:   if ∃x ∈ [i, j] and CoverageVector[x] == 0 then
4:     if k > j or l < i then
5:       return TRUE;
6:     end if
7:   end if
8: end if
9: if y >= k and l >= x then
10:  if x == k and y == l and t == t' then
11:    return FALSE;
12:  end if
13:  return TRUE;
14: end if
15: return FALSE;
```

函数的伪代码（算法5.2）。

系统使用限制解码算法进行解码，可以避免限制片段被作为多个子部分被分别翻译时，与其他部分发生的调序错误，从而使限制片段中的各个子部分被连续翻译，最终生成目标语言的连续词序列。

至此，我们可以将限制片段的交互方法与选择 -修正交互翻译框架结合在一起。我们只需要将算法3.1 与算法5.2结合在一起，形成算法5.3，就可以将两种交互方法融合进短语翻译系统中。需要注意的是，受限于短语表的规模与质量，当用户提供了多个限制片段时，不同限制片段的交界处可能不存在同时满足两个限制片段的翻译选项。在这种情况下，进行限制解码可能会导致解码失败，所以在实际应用中，通常限制片段交互不应过多。我们通常要求用户对每句只进行一到两次限制片段交互。

在限制翻译片段的交互方法中，系统同样可以捕获到所有限制片段，在捕获到一定数量的限制片段后进行模型自适应，重新训练模型，提升翻译系统本身的性能。

5.4 实验及结果分析

5.4.1 实验配置及实验方法

本章使用的短语翻译系统与第3.3章中描述的系统一致。在此基础上，我们将限制片段的交互方法加入到系统中。我们同样使用不区分大小写的 4 元文法 BLEU 来进行翻译质量的自动评价。

在限制片段交互方法中，由于机器翻译调序的复杂性高，调序的搜索空间大，所以难以进行合适的模拟交互实验。在这样的背景下，我们让真实用户参与交互翻译工作，以此来测试限制片段交互方法的性能。

因为让真实用户进行交互翻译实验的代价高，耗时长，所以我们只能让用户进行较小范围的交互实验。我们从 NIST03 数据集中随机抽取了 120 句作为测试数据集，并邀请了 10 位有着良好中英互译水平的硕士研究生，要求他们使用限制片段交互方法在测试数据集上进行中英翻译的交互翻译工作。

我们提供了一个基于网页的支持限制翻译片段的交互式翻译平台，并要求每个用户独立使用该平台进行交互翻译。我们将测试数据集平均分成十个小数据集，每个用户负责其中一个小数据集的交互翻译工作。在用户进行交互翻译的过程中，我们限制用户对每句最多进行一次限制翻译片段的交互操作。

5.4.2 限制片段交互方法对翻译质量的影响

表 5-3 中给出了小规模真实用户实验的相关数据。从第二行开始，每一行分别表示测试数据集的句对总数、基线系统的 BLEU 得分、需要进行限制片段交互的句对总数、限制片段交互之后的 BLEU 得分以及在测试数据集上进行限制片段交互所需要的 KSMR。

表 5-3: 限制翻译片段交互翻译结果

数据	NIST_{sample}
句对总数	120
基线系统得分	30.23
需要限制片段的句对总数	71
限制片段后得分	32.78 (+2.55)
KSMR [%]	1.39

在 120 个句子中，用户对基线系统的输出进行限制短语片段交互的有 71 个，比例高达 60%。这样的结果说明了短语调序错误在短语机器翻译中非常明显。用户对测试集中的 71 句只进行一次限制片段交互翻译，系统完成限制解码后，测试数据集的翻译结果的 BLEU 得分由 30.23 增长为 32.78 (+2.55)，这样的增长在统计意义上是显著的。

在第节中所描述的人机交互方法下，每句进行一次限制片段交互时，平均需要进行四次鼠标点击操作。用户对 120 个句对中的 71 个使用了限制翻译片段交互，所以整个测试数据集需要用户进行约 284 次鼠标点击操作。经统计，测试数据集中句子的参考译文的平均字符数为 170，所以根据第 1.2节中 KSMR 的计算方法可知，用户使用限制片段交互方法对测试集中的句子进行交互翻译，平均每句进行一次限制片段交互所需要的 KSMR 仅为 1.39% 。

实验结果表明，限制片段的交互方法在完全不需要用户提供短语正确翻译的情形下，使用极少的交互即可获得明显的翻译质量的增长。另外，我们还可以进一步优化人机交互方式，例如支持用户拖动，选中等操作，使每次限制片段交互使用更少的鼠标点击数，进一步降低 KSMR。

5.5 本章小结

本章在 PR 交互翻译框架的基础上，针对 PR 交互翻译框架中无法让用户对短语翻译调序进行控制的问题，提出了限制翻译片段的交互方法及对应的限制解码算法。限制翻译片段交互方法也是一种迭代交互方法，但是该方法不需要用户提供任何源端短语的正确翻译信息，只要求用户提供源端的一个连续词序列（片段）信息就可以让用户控制解码的短语翻译顺序。用户提供的词序列在真实翻译中应该作为整体被连续翻译，其内部短语不应与外部其他短语调序。我们将 PR 交互框架中的限制解码算法与限制片段交互方法中的限制解码算法结合起来，作为一个统一的限制解码算法整合到交互式机器翻译系统中。

为了进行合理的交互翻译实验，我们邀请了真实用户来使用限制片段交互方法进行交互翻译实验。由于让真实用户进行实验代价高，耗时长，所以我们只进行了小规模的实验。实验结果表明，限制翻译片段的交互方法在不需要用户提供正确翻译的前提下，仍然能够带来翻译质量的显著提升。同时，用户进行交互翻译时所需要的交互次数也非常少。我们可以进一步优化交互方式来进一步降低交互次数。

第六章　模型自适应

6.1　引言

在前三章中，我们分别描述了基于短语翻译系统的 PR 交互翻译框架，PR 交互翻译框架下的自动建议模型，以及 PR 交互翻译框架的扩展方法，即限制翻译片段的交互方法。在我们的交互框架下，用户可以使用简单的鼠标点击等操作进行人机交互，翻译系统根据交互信息在当前句子中进行限制解码，从而获得更优的翻译结果。

在上述交互框架中，系统只根据当前会话中保存的交互信息来提高当前句子的翻译质量，并没有将交互信息用于翻译系统中的统计模型本身中。本章主要描述交互翻译系统根据交互信息进行模型自适应的方法。模型自适应方法可以交互信息与翻译系统中的统计模型相结合，从而提升翻译系统翻译能力。在上一章中，由于我们并没有进行大规模的限制片段模拟交互实验，而是进行了小规模的人工交互实验，所以产生的交互信息少，没有足够的数据进行模型自适应。在此背景下，本章主要描述使用 PR 交互框架下产生的交互信息进行模型自适应的方法。

6.2　典型方法及其不足

Germann 等人[31] 于 2014 年提出了使用动态短语表的模型自适应方法。该方法的主要思想是按需加载短语表[32] 和双语采样。在翻译一个源语言句子时，首先按照该句的内容在训练数据集中进行双语采样，进而在采样出的小规模样本上抽取短语表并加载到内存中。对于一个源端句子，当用户完成交互翻译后，最终生成的翻译都被认为是正确的，所以系统可将该句及其最终翻译对作为训练数据加入训练语料中。系统就可以在更新后的训练语料上利用动态短语表来影响后续句子的翻译，从而起到模型自适应的作用。

为了降低双语采样的时间开销，Germann 等人使用了后缀数组来组织双语训练语料。在此基础上，Germann 等人还将短语表表示为二进制形式，并使用内存映射等技巧来提高读入效率。经过上述优化后，与传统的短语表表示方

法相比（预先将短语表全部读入内存，使用时基于哈希技术进行常数时间查询），时间开销可比较。

虽然 Germann 等人的方法可以做到实时更新训练语料，但由于该方法必须从语料中采样出少部分样本，所以会带来搜索空间的损失，进一步可能导致翻译质量的损失；其次，该方法也没有更新翻译系统的对数线性模型的参数，忽略了模型参数对翻译系统质量的影响。

Marie 等人[33] 于 2015 年提出基于触控的预 -后处理（Touch-based Pre-post-editing, PPE）的方法。该方法首先让翻译系统生成原始翻译，进而让用户标记原始翻译结果中哪些片段应该出现在最终翻译中（正确的片段），哪些片段不应该出现在最终翻译中（错误的片段）。当用户对每个句子的翻译标记结束后，系统从这些标记好的片段中根据词对齐抽取正、负翻译模型（正确片段中抽出的短语对为正，反之为负），并将原先短语表中对应的翻译选项去除；同时，从这些标记好的片段中训练 n 元文法正、负语言模型，正负的定义与正负翻译模型一致；最后，PPE 方法将四个模型作为额外特征加入到机器翻译系统中的对数线性模型中，并重新训练模型参数。

虽然 PPE 方法能够抽取出补充模型并更新模型参数，但是 PPE 方法实际上并没有将用户信息充分利用起来。用户标记的内容只被用于被用户标记过的句子中，对新句子不起作用，所以实际上并不能起到模型自适应的作用。

6.3 自适应流程及方法

图 6-1给出了模型自适应的流程。用户使用本文提出的 PR 交互翻译框架进行交互翻译，系统可以在人机交互不断进行的过程中捕获所有交互信息，当捕获到的信息量足够多时进行模型自适应。

本文提出的模型自适应方法对上文中描述的典型方法中存在的问题作出改进。一方面，我们仍然使用传统的短语表表示方法，即在翻译任务开始之前，预先将短语表读入内存中，使得翻译系统在使用短语表时可以以常数时间读取，且不丢失已有的翻译选项。我们将短语表表示为二进制文件，并使用内存映射等方法提升短语表、语言模型等的读入效率。另一方面，我们使用了三种不同的方法进行模型自适应，分别是补充翻译模型方法，翻译模型插值方法和翻译模型修正方法。我们的方法改进了 Marie 等人的 PPE 方法，解决了其不具有泛化能力的问题。在上述三种方法的基础上，我们根据对真实实验结果的分

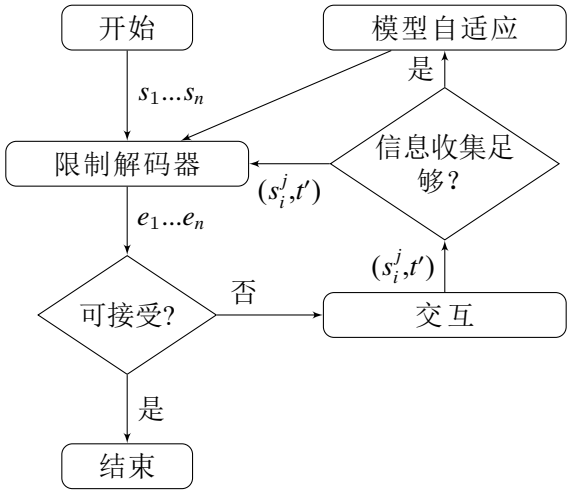

图 6-1: 模型自适应流程图

析，进一步将翻译模型插值方法和翻译模型修正方法相结合，更大程度上提升了翻译系统的翻译性能。

6.3.1 补充翻译模型

补充翻译模型方法（Supplementary Translation Model, STM）的主要思想是在现有的模型的基础上补充新的翻译模型。我们首先给出 PRP 与翻译选项一致的定义：当一个给定 PRP 与一个给定翻译选项的源端完全相同，目标端也完全相同，则称两者一致。

我们将交互过程中产生的每个 PRP 记录下来，并将基线系统的翻译模型中与该 PRP 一致的翻译选项加入到补充翻译模型中，所以补充翻译模型实际上是基线系统中的翻译模型的子集。我们将补充翻译模型作为 4 维扩展特征，其对应的词汇化调序模型作为另外 6 维扩展特征，共 10 维扩展特征加入到短语翻译系统的对数线性模型中并调节模型参数。模型自适应的具体过程如下：

1. 初始时，补充翻译模型为空，被加入到对数线性模型中。初始化补充翻译模型的权重的方法与标准的翻译模型的权重初始化方法一致。由于初始时补充翻译模型中不包含任何翻译选项，所以其各项特征值都为 0。
2. 用户使用 PR 交互框架进行交互翻译，对于数据集中的每个源语言句子，每次 PR 交互产生一个 PRP，(s_i^j, t)。
3. 系统接收 (s_i^j, t)，进行限制解码，生成新的句子翻译 e，并将 (s_i^j, t) 存储起来。

4. 当接收到一定量的数据时，将基线系统的翻译模型中所有与 PRP 一致的翻译选项组织成补充翻译模型 TM_s，以二进制形式存储。
5. 系统将补充翻译模型读入内存，将其作为对数线性模型的新特征，使用开发集进行最小化错误率训练，更新模型参数。
6. 重复步骤 2 到 4 直至翻译任务结束。

用户可以使用 PR 交互框架进行多轮交互，系统可以根据收集到的交互信息进行多轮模型自适应以持续提升翻译质量。

6.3.2 翻译模型插值

在第6.3.1节中，我们描述了使用补充翻译模型进行模型自适应的方法。这种方法在翻译系统中加入了一个补充翻译模型 TM_s，并调节模型参数。该方法有一个潜在缺点是，当翻译系统使用与某个 PRP 一致的翻译选项进行假设扩展时，TM 中的特征与 TM_s 中的特征同时被激活，这就削弱了系统对 PRP 与非 PRP 的区别能力。

本节中，我们针对上述问题，对补充翻译模型作出改进，提出翻译模型插值（Translation Model Interpolation, TMI）的方法。该方法与补充翻译模型方法类似，都是迭代更新的过程。不同的是，当收集到一定数量的 PRP 后，系统一方面将与 PRP 一致的翻译选项组织成补充翻译模型 TM_s，TM_s 中各项概率值的设置方法与上一节一致；另一方面，将基线系统中的翻译模型 TM 中所有与 PRP 一致的翻译选项去除，得到 TM'。由于 TM_s 实际上是 TM 的一个子集，所以这样的设置方法保证了 TM' 与 TM_s 的交集为空。在这种设置下，我们将 TM' 与 TM_s 作为两个不同的翻译模型，加入到对数线性模型中。我们用 TM' 取代 TM，以 TM' 作为翻译系统主要使用的翻译模型，TM_s 作为补充翻译模型，并使用 MERT 进行模型参数的调节。

由于 TM_s 与 TM' 交集为空，所以 TM_s 与 TM' 作为两个独立的部分在对数线性模型中起作用，两者互不影响。该方法实际上相当于两个翻译模型的对数线性插值，故本方法被称为翻译模型插值法。与补充翻译模型的方法相同，系统可以进行多轮模型自适应训练，从而逐步提高翻译质量。

6.3.3 翻译模型修正

在第6.3.1节和第6.3.2节中，我们分别描述了补充翻译模型和翻译模型插值两种模型自适应的方法。这两种方法均使用了补充翻译模型的概念，都在现有的对数线性模型上加入额外的 10 维特征进行参数训练。这两种方法存在一定的弱点，一方面都扩大了参数空间，使得参数训练过程相对更复杂；另一方面，都没有更新翻译模型本身的概率分布，而传统的模型自适应方法通常采用更新模型概率分布的方法并取得了一定成功。

本节中，我们针对上述问题，提出了翻译模型修正（Translation Model Modification, TMM）的方法。该方法仍然是迭代更新的过程。与前两节中的方法不同，一方面，该方法不需要扩大参数空间，其参数训练的过程与翻译系统本身的参数训练过程完全一致；另一方面，该方法根据交互信息对翻译模型本身的概率分布做修正从而达到模型自适应的目标。

直观而言，人机交互中产生的 PRP 在翻译中应该起更重要的作用，所以其翻译概率应该更高。我们基于这种思想利用收集到的 PRP 来更新翻译模型的概率分布。具体地，每当 PRP 出现一次（用户在交互过程中使用了该 PRP），该 PRP 的计数加一。当收集到一定量的 PRP 后，根据 PRP 的出现次数提高与该 PRP 一致的翻译选项的翻译概率，并利用新得到的翻译模型重新训练模型参数。

翻译概率的具体更新方法见算法6.1，其根本思想是根据 PRP 出现的次数来修正翻译选项的正向、逆向翻译概率，并进行归一化。当一个翻译选项与某个 PRP，(s,t)，一致时，以公式6-1的幅度增加其正向、逆向翻译概率，并进行归一化。

$$count(PRP|PRP.source == s, PRP.target == t) * \alpha \tag{6-1}$$

其中，*count* 表示该 PRP，(s,t)，出现的次数，α 是一个超参（Hyper Parameter），表示用户对 PRP 的信任程度。另外，我们还需要对短语表中其他相关的翻译选项的正向、逆向翻译概率进行归一化。当翻译选项 (s,t) 的源端 s 与某个 PRP 的源端 *PRP.source* 相同，但目标端不同时，需要使用公式6-2归一化其正向翻译概率。

$$p(t|s) = {p(t|s)}/{(1 + \alpha * count(PRP|PRP.source == s))} \tag{6-2}$$

其中 $count(PRP|PRP.source == s)$ 表示源端为 s 的 PRP 的数目，α 的定义与公

式6-1中一致。

类似的，当 (s,t) 的目标端 t 与某个 PRP 的源端 *PRP.target* 相同，但源端不相同时，需要使用公式6-3归一化其逆向翻译概率。

$$p(s|t) = {}^{p(s|t)}\big/_{(1+\alpha * count(PRP|PRP.target == t))} \tag{6-3}$$

其中 $count(PRP|PRP.target == t)$ 表示目标端为 t 的 PRP 的数目，α 的定义与公式6-1中一致。

当 (s,t) 不与任何 PRP 相关（源端、目标端都不相同），则其翻译概率不需要改变；这样的概率更新方法可以使得与 PRP 一致的翻译选项的翻译概率得到提升，其他相关的翻译选项的翻译概率下降，完全无关的翻译选项则不受影响。

算法 6.1 翻译模型概率修正计算方法

for all (s,t) ∈ TM **do**
  if $\exists PRP \in PRPs$ and $PRP.source == s$ and $PRP.target == t$ **then**
    $p(t|s) = {}^{(\alpha * count(PRP) + p(t|s))}\big/_{(1+\alpha * count(PRP))}$
    $p(s|t) = {}^{(\alpha * count(PRP) + p(s|t))}\big/_{(1+\alpha * count(PRP))}$
  else if $\exists PRP \in PRPs$ and $PRP.source == s$ and $PRP.target != t$ **then**
    $p(t|s) = {}^{p(t|s)}\big/_{(1+\alpha * \sum_{PRP'.source==s} count(PRP'))}$
  else if $\exists PRP \in PRPs$ and $PRP.source != s$ and $PRP.target == t$ **then**
    $p(s|t) = {}^{p(s|t)}\big/_{(1+\alpha * \sum_{PRP'.target==t} count(PRP'))}$
  else
    continue;
  end if
end for

6.3.4 模型插值与模型修正相结合

在前三节中，我们分别描述了补充翻译模型、模型插值和模型修正三种基础的模型自适应方法。这三种模型自适应方法各有特点，也可以在一定程度上提升翻译系统的性能。

我们将模型插值方法与模型修正方法相结合，综合两者的优势，期望能够最大程度地提升翻译系统的性能。具体地，我们在第6.3.2节中的模型插值方法的基础上作出改进，一方面仍然将 PRP 组织成补充翻译模型 TM_s，并将短语系统本身的翻译模型 TM 中对应的部分去除，得到 TM'；另一方面，使用第6.3.3节中的方法修正 TM_s 和 TM' 中翻译选项的概率分布，而非直接使用

TM 中的概率分布。在此基础上，用 TM' 替代 TM，并将 TM_s 加入对数线性模型，使用 MERT 调节模型参数。

6.4 实验及结果分析

6.4.1 实验配置及实验方法

本章使用的短语翻译系统仍然与第3.3章中描述的一致，具体在此不再赘述。

因为真实的人工交互代价昂贵且耗费时间，所以我们在实验中使用了模拟 PR 交互操作（见第4.2.1.1节）。我们在 NIST02 与 NIST03 数据集的并集 NIST0203 上进行交互翻译实验，每轮交互都修正数据集中每句的最关键的错误。当对数据集中每个句子都进行一次 PR 交互后，完成一轮整个数据集上的 PR 交互，随后系统开始使用 NIST0203 作为开发集 c 开始模型自适应，进而完成一轮模型自适应。

6.4.2 模型自适应对翻译质量的影响

6.4.2.1 补充翻译模型对翻译质量的影响

表6-1给出了补充翻译模型的模型自适应方法对翻译质量的影响。第一行给出了数据相关信息，第二行表示基线系统在三个数据集上的翻译得分。STM*n 表示进行了 n 轮补充翻译模型自适应方法后，翻译系统在三个数据集上的翻译得分。

表 6-1: 补充翻译模型自适应方法对翻译质量的影响

数据	NIST0203	NIST04	NIST05
基线系统	30.29	31.83	30.64
STM*1	32.87 (+2.58)	32.17 (+0.34)	30.62 (-0.02)
STM*2	34.15 (+3.86)	32.63 (+0.80)	30.91 (+0.34)
STM*3	35.12 (+4.83)	33.04 (+1.21)	31.22 (+0.58)

从表6-1中可以看出，在第一轮模型自适应后，虽然系统平均在每个句子

中只捕获了一个 PRP，但是对于当前开发数据 NIST0203 而言，翻译质量提升了 2.58 BLEU，这样的提升已经非常明显。这样的结果证明了补充翻译模型的模型自适应方法的有效性。对于 NIST04 数据集，使用模型自适应后的系统进行解码，BLEU 得分从 31.83 变为 32.17 (+0.34)；对于 NIST05 数据集，BLEU 得分从 30.64 变为 30.62 (-0.02)。NIST04 数据集的翻译质量上升，NIST05 几乎不变。这样的结果表明当前的系统泛化能力还不够强，需要更多交互数据继续进行模型自适应。

第二轮模型自适应（STM*2）后，NIST0203 的翻译质量进一步提升（+3.86 BLEU）。NIST04 和 NIST05 两个数据集的 BLEU 得分分别提高了 0.80 和 0.34。相比第一轮模型自适应，NIST04 的翻译质量进一步提升，NIST05 的翻译质量从几乎无变化到统计显著的提升。这样的结果说明在两轮自适应后，系统的泛化能力得到了一定程度的增强。第三轮模型自适应（STM*2）后，系统在三个数据集上的翻译质量继续提升，这说明补充翻译模型的模型自适应方法可以在不断迭代的过程中持续提升翻译系统的翻译能力。

6.4.2.2 翻译模型插值对翻译质量的影响

表 6-2给出了翻译模型插值的模型自适应方法对翻译质量的影响。第一行给出了数据相关信息，第二行表示基线系统在三个数据集上的翻译得分。TMI*n 表示进行了 n 轮翻译模型插值自适应方法后，翻译系统在三个数据集上的翻译得分。

表 6-2: 模型插值自适应对翻译质量的影响

数据	NIST0203	NIST04	NIST05
基线系统	30.29	31.83	30.64
TMI*1	33.27 (+2.98)	32.50 (+0.67)	30.98 (+0.34)
TMI*2	35.01 (+4.72)	32.95 (+1.12)	31.47 (+0.83)
TMI*3	36.35 (+6.06)	33.54 (+1.71)	31.87 (+1.23)

从表 6-2中可以看出，在第一轮模型自适应后，与补充翻译模型的自适应方法相似，当前开发数据（NIST0203）的翻译质量有明显的提升（+2.98），且提升更为明显，这说明了模型插值的自适应方法对于开发集更有效。对于 NIST04 数据集，使用模型自适应后的系统进行解码，BLEU 得分进一步

提升（+0.67），相比补充翻译模型的方法，翻译质量的提升程度更大；对于NIST05 数据集，BLEU 得分有 +0.34 的提升，仍然比补充翻译模型的方法提升程度更大。

第二轮模型自适应（TMI*2）后，对于三个数据集，NIST0203，NIST04，NIST05 分别有 +4.72，+1.12，+0.83 的 BLEU 提升。继续进行第三轮模型插值自适应（TMI*3），NIST0203 的翻译质量提升了 +6.06 BLEU，NIST04 和NIST05 也分别有 +1.71 和 +1.23 的 BLEU 提升，这样的提升已经非常显著，说明翻译模型插值的方法同样可以在不断迭代的过程中持续提升翻译系统的翻译能力。相比补充翻译模型的方法，翻译模型插值方法对翻译系统的翻译能力提升更明显，这样的结果说明了模型插值的方法相比基于补充翻译模型的方法更优。

6.4.2.3 翻译模型修正对翻译质量的影响

表 6-3给出了翻译模型修正的自适应方法对翻译质量的影响。第一行给出了数据相关信息，第二行表示基线系统在三个数据集上的翻译得分。TMM*n 表示进行了 n 轮翻译模型修正自适应方法后，翻译系统在三个数据集上的翻译得分。接下来两行分别表示 α 取值不同时，进行一轮模型自适应后三个数据集上的翻译得分。其中 α 系统表示对 PRP 的信任程度，可根据实际情况调整。

表 6-3: 模型修正自适应对翻译质量的影响

数据	NIST0203	NIST04	NIST05
基线系统	30.29	31.83	30.64
TMM*1 ($\alpha = 0.1$)	31.90 (+1.61)	31.92 (+0.09)	30.63 (-0.01)
TMM*2 ($\alpha = 0.1$)	32.71 (+2.42)	32.31 (+0.48)	31.16 (+0.52)
TMM*3 ($\alpha = 0.1$)	33.41 (+3.12)	32.85 (+1.02)	31.58 (+0.94)
TMM*1 ($\alpha = 0.3$)	32.34 (+2.05)	31.88 (+0.05)	30.63 (-0.01)
TMM*1 ($\alpha = 0.5$)	32.93 (+2.64)	31.69 (-0.14)	30.47 (-0.15)

从表 6-1中可以看出，在第一轮模型自适应后，当前开发数据（NIST0203）的翻译质量有较明显的提升（+1.61），相比补充翻译模型和模型插值方法，模型修正的方法带来的翻译质量的提升幅度相对较小。NIST04 和 NIST05 数据集的翻译质量几乎无变化，说明当前模型自适应的泛化能力还较弱。

第二轮模型自适应（TMM*2）之后，NIST0203 的 BLEU 得分从 31.83 变为 32.71 (+2.42)，NIST04 的 BLEU 得分从 31.83 变为 32.31 (+0.48)，NIST05 的 BLEU 得分从 30.64 提升到 31.16 (+0.52)，虽然翻译质量的提升幅度没有前两种方法大，但是提升也已经比较明显，说明系统在交互信息变多的情况下，泛化性得到了增强。在第三轮模型自适应（TMM* 3）之后，三个数据集的结果相比之前的结果又有了一定提升，说明该方法的泛化性能进一步提升。这样的结果说明翻译模型修正的自适应方法也可以在不断迭代的过程中持续提升翻译系统的翻译能力。

我们同样对比了不同 α 对翻译质量的影响。当 α 分别为 0.1, 0.3, 0.5 时，对当前开发数据集 NIST0203 而言，BLEU 得分分别为 31.90 (+1.61)，32.34 (+2.05)，32.93 (+2.64)，这说明 α 越大，开发集的翻译质量提升越明显。对于 NIST04 和 NIST05 两个数据集，当 $\alpha = 0.1$ 和 $\alpha = 0.3$ 时，进行一轮模型自适应后翻译质量几乎无变化；当 $\alpha = 0.5$ 时，两个数据集上的翻译质量都有一定程度的降低，这说明此时 α 的取值过大，造成了模型的过拟合，导致泛化能力降低。在实际应用中，我们需要根据实际情况进行 α 的取值。

6.4.2.4 翻译模型插值与翻译模型修正相结合对翻译质量的影响

表 6-4给出了翻译模型插值和翻译模型修正相结合的自适应方法对翻译质量的影响。第一行给出了数据相关信息，第二行表示基线系统在三个数据集上的翻译得分。(I+M)*n 表示进行了 n 轮模型自适应方法后，翻译系统在三个数据集上的翻译得分。其中 α 的取值为 0.1。

表 6-4: 模型插值与模型修正结合自适应对翻译质量的影响

数据	NIST0203	NIST04	NIST05
基线系统	30.29	31.83	30.64
(I+M)*1 ($\alpha = 0.1$)	33.32 (+3.03)	32.56 (+0.73)	31.09 (+0.45)
(I+M)*2 ($\alpha = 0.1$)	35.53 (+5.24)	33.11 (+1.28)	31.84 (+1.20)
(I+M)*3 ($\alpha = 0.1$)	36.48 (+6.19)	33.78 (+1.95)	32.21 (+1.57)

从表 6-4中可以看出，在第一轮模型自适应后，当前开发数据集（NIST0203）的翻译质量有明显的提升（+3.03），相比前三种方法，翻译质量的提升更大。

对于 NIST04 和 NIST05 数据集，BLEU 得分分别提高了 0.73 和 0.45，相比前三种方法，翻译质量的提升仍然更高，这说明了该方法的泛化能力更强。

在第二轮模型自适应（(I+M)*2）之后，NIST0203 的 BLEU 得分继续提升（+5.24），NIST04 与 NIST05 的 BLEU 得分分别提高了 +1.28 和 +1.20。在第三轮模型自适应（(I+M)* 3）之后，三个数据集的结果相比之前的结果又有了一定提升，分别提升了 6.19，1.95 和 1.57 BLEU，且提升都比前三种方法更大，这说明将翻译模型插值与翻译模型修正相结合的方法可以同时发挥模型插值和模型修正两种方法各自的优势，更大程度地提升翻译系统的能力。

6.4.3 模型自适应方法对比

本章所提出的模型自适应方法之间互有联系也互有区别。

首先，本文提出的模型自适应方法都是迭代的过程，可以不断地使用人机交互信息提升翻译系统的性能；其次，都需要使用 MERT 来调整模型参数。

在对翻译质量的提升方面，三种基本的模型自适应方法中翻译模型插值的方法能力最强；其次是补充翻译模型的方法；最后是翻译模型修正的方法。其中，模型修正的方法需要结合用户先验和实际系统确定 α 的取值，取值的不同对翻译质量有一定影响。在交互信息不断增多的情况下，三种方法对翻译质量的提升能力都会逐步加强。将翻译模型插值与翻译模型修正的方法相结合的方法能够发挥两者的优势，更大程度地提高翻译系统的翻译性能。

在时间复杂度方面，因为三种基本方法都使用 MERT 来进行参数学习，所以三者的时间复杂度没有本质差别。补充翻译模型和翻译模型插值的方法都在短语翻译系统的对数线性模型的基础上增加了 10 维特征，扩大了特征空间；翻译模型修正的方法直接修正短语表中的翻译概率，不影响特征空间，所以训练过程相对较快。将翻模型插值与翻译模型修正的方法相结合的方法的训练时间复杂度与翻译模型插值方法一致。

在空间复杂度方面，除翻译模型修正方法之外，其他模型自适应方法都需要额外的空间存储一个新的翻译模型。随着交互翻译的不断进行，产生的交互信息越来越多，新的翻译模型所需的空间也将越来越大。

6.5 本章小结

本章针对传统的机器翻译系统中典型的模型自适应方法中存在的若干问题，提出了基于选择 -修正交互框架的翻译系统模型自适应的方法。

我们提出了三种基本方法，分别是补充翻译模型、翻译模型插值、翻译模型修正的方法。补充翻译模型的方法将基线系统的翻译模型中与 PRP 一致的翻译选项整合成新的翻译模型，并将其作为新特征加入到对数线性模型中，进而使用 MERT 调节模型参数；翻译模型插值的方法建立在补充翻译模型的基础上，将补充翻译模型与基线系统的翻译模型完全区分开，同样使用 MERT 调节参数；翻译模型修正方法根据 PRP 的出现频次对翻译选项的概率分布进行修正，进而采用 MERT 进行参数调节。在三种基本方法的基础上，将翻译模型插值方法和模型修正方法相结合可以发挥两者各自的优势。

实验结果表明，在进行一定量的人机交互之后，本章提出的模型自适应方法不仅能带来开发集翻译质量的显著提升，而且具有较强的泛化能力，可以切实提升翻译系统的能力。将模型插值与模型修正相结合的方法对翻译系统能力的提升最大。

模型自适应方法可以与本文提出的选择 -修正交互翻译框架融为一体，作为交互翻译系统的一部分，利用交互过程中产生的信息来提升翻译系统的翻译能力。用户在使用交互翻译系统的过程中，一方面可以使用本文提出的 PR 交互框架进行交互，实时提高当前翻译的质量；另一方面，将交互信息提供给系统，系统利用模型自适应方法提升翻译性能，从而降低交互需求。

第七章　总结与展望

7.1　工作总结

主流的交互式机器翻译框架采用自左向右进行句子翻译补全的方法（Left-to-Right, L2R），用户输入句子翻译，系统根据用户的输入不断提供补全建议，用户既可以接受建议也可以拒绝建议。虽然 L2R 的框架取得了较大成功，但该框架难以优先修正句末的关键错误。当一个关键错误出现在句末，而该翻译错误又导致了句首的翻译错误，从左到右进行翻译修正就会延迟对该错误的修正，从而可能导致交互效率的低下。

对于以上问题，本文从以下几方面展开了具体工作：

1. 提出了选择 -修正的交互式机器翻译框架（Pick-Revise, PR）。选择 -修正框架是一个迭代式的交互翻译框架，在这种框架下，用户的交互只需要两个简单的操作即可完成（选择和修正），用户可以修正任何句子中任意位置的翻译错误，以此来提高交互翻译效率，减少用户交互次数。基于该框架，我们提出交互翻译的限制解码算法，使机器翻译系统能够自然地利用用户提供的信息进行解码，并获得更优翻译结果。实验结果表明，对比 L2R 和 PE 系统，我们的 PR 交互框架可以在极少的用户操作下（3.3% KSMR），带来翻译质量的大幅提升（+17 BLEU）。
2. 在 PR 翻译框架的基础上，对用户的两种操作分别提出了自动建议模型，包括选择建议模型和修正建议模型。选择自动建议模型可以预测出可能是翻译关键错误的短语并建议用户进行选择；修正自动建议模型可以预测出可能是正确翻译结果的翻译选项并建议用户用该翻译选项替换错误翻译。用户对于两个建议模型的建议都可以采用或拒绝。在两个自动建议模型的辅助下，用户的操作进一步被简化，可以进一步提高人机交互效率。实验结果表明，当用户只做一种操作，而让自动建议模型完成另一种操作，只需要很小的交互代价就可以带来较大的翻译质量的提升。
3. 针对翻译系统的短语调序错误，我们扩展了 PR 交互翻译框架，引入了限制翻译片段的交互方法。限制翻译片段的交互方法也是一种迭代式的交互方法，该交互方法中，我们要求用户提供限制片段信息，系统利用本文提出的

限制解码算法，根据限制片段来进行解码，从而提高翻译质量。实验证明，用户只需要进行很少的交互操作（1.39% KSMR），且不需要提供正确目标端翻译信息，就可以带来明显的翻译质量的提升（+2.55 BLEU）。

4. 在 PR 交互框架的基础上，本文提出了三种基础性的模型自适应的方法，分别是补充翻译模型、翻译模型插值、翻译模型修正方法。在此基础上，将翻译模型插值方法与翻译模型修正方法相结合可以取得更好的自适应效果。模型自适应是迭代更新的过程，不断收集人机交互过程中产生的信息进行自适应。在不断进行人机交互和模型自适应时，翻译系统的翻译能力也能够逐步提高。

7.2 未来工作

现有工作中，仍然存在若干值得继续研究的点，未来工作可以基于这些点进行更深入的研究：

1. 在当前的 PR 交互翻译框架下，我们提出的自动建议模型虽然能够将用户的操作简化成单一操作，目前的性能也达到可接受的程度，但其模型能力仍然有较大提升的空间。我们可以通过更大的数据规模，更多的特征，更优的模型来进一步提升模型能力。
2. 我们没有进行限制片段翻译方法的大规模实验。未来工作可以让更多的真实用户参与。与 PR 框架中的自动建议模型类似，限制片段中的自动建议模型的设计和实现也是一个值得研究的方向，另外，基于限制片段的模型自适应的学习也可以在大规模的数据上展开。
3. 模型自适应方法也是一个值得深入研究的方向。一方面，可以研究提高自适应的效率的方法，做到更快的、实时的模型自适应。另一方面，可以继续研究如何改进当前的模型自适应方法。

参考文献

[1] ALLEN J. Post-editing[J]. Benjamins Translation Library, 2003, 35 : 297 – 318.

[2] FOSTER G, ISABELLE P, PLAMONDON P. Word completion: A first step toward target-text mediated IMT[C] // Proceedings of the 16th conference on Computational linguistics-Volume 1. 1996 : 394 – 399.

[3] FOSTER G, ISABELLE P, PLAMONDON P. Target-text mediated interactive machine translation[J]. Machine Translation, 1997, 12(1-2) : 175 – 194.

[4] LANGLAIS P, FOSTER G, LAPALME G. TransType: a computer-aided translation typing system[C] // Proceedings of the 2000 NAACL-ANLP Workshop on Embedded machine translation systems-Volume 5. 2000 : 46 – 51.

[5] FOSTER G, LANGLAIS P, LAPALME G. User-friendly text prediction for translators[C] // Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. 2002 : 148 – 155.

[6] BARRACHINA S, BENDER O, CASACUBERTA F, et al. Statistical approaches to computer-assisted translation[J]. Computational Linguistics, 2009, 35(1) : 3 – 28.

[7] KOEHN P. A web-based interactive computer aided translation tool[C] // Proceedings of the ACL-IJCNLP 2009 Software Demonstrations. 2009 : 17 – 20.

[8] KOEHN P, TSOUKALA C, SAINT-AMAND H. Refinements to Interactive Translation Prediction Based on Search Graphs[J], 2014.

[9] GREEN S, CHUANG J, HEER J, et al. Predictive Translation Memory: A mixed-initiative system for human language translation[C] // Proceedings of the 27th annual ACM symposium on User interface software and technology. 2014 : 177 – 187.

[10] OCH F J, ZENS R, NEY H. Efficient search for interactive statistical machine translation[C] // Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1. 2003 : 387 – 393.

[11] GONZÁLEZ-RUBIO J, ORTIZ-MARTÍNEZ D, BENEDÍ J-M, et al. Interactive Machine Translation using Hierarchical Translation Models.[C] // EMNLP. 2013 : 244 – 254.

[12] SANCHIS-TRILLES G, ORTIZ-MARTINEZ D, CASACUBERTA F. Efficient wordgraph pruning for interactive translation prediction[C] // Annual Conference of the European Association for Machine Translation (EAMT). 2014.

[13] ORTIZ-MARTINEZ D, GARCIA-VAREA I, CASACUBERTA F. Online learning for interactive statistical machine translation[C] // Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2010 : 546 – 554.

[14] OCH F J, NEY H. Discriminative training and maximum entropy models for statistical machine translation[C] // Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. 2002 : 295 – 302.

[15] OCH F J, NEY H. A Systematic Comparison of Various Statistical Alignment Models[J]. Computational Linguistics, 2003, 29(1) : 19 – 51.

[16] MARCU D, WONG W. A phrase-based, joint probability model for statistical machine translation[C] // Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. 2002 : 133 – 139.

[17] KOEHN P, OCH F J, MARCU D. Statistical phrase-based translation[C] // Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. 2003 : 48 – 54.

[18] KOEHN P. Statistical machine translation[M]. [S.l.] : Cambridge University Press, 2009.

[19] TILLMANN C. A unigram orientation model for statistical machine translation[C] // Proceedings of HLT-NAACL 2004: Short Papers. 2004 : 101 – 104.

[20] MANNING C D, SCHÜTZE H. Foundations of statistical natural language processing: Vol 999[M]. [S.l.]: MIT Press, 1999.

[21] OCH F J. Minimum error rate training in statistical machine translation[C] // Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1. 2003: 160–167.

[22] PAPINENI K, ROUKOS S, WARD T, et al. BLEU: a method for automatic evaluation of machine translation[C] // Proceedings of the 40th annual meeting on association for computational linguistics. 2002: 311–318.

[23] MOHIT B, HWA R. Localization of difficult-to-translate phrases[C] // Proceedings of the Second Workshop on Statistical Machine Translation. 2007: 248–255.

[24] GREEN S, WANG S I, CHUANG J, et al. Human Effort and Machine Learnability in Computer Aided Translation.[C] // EMNLP. 2014: 1225–1236.

[25] CHEN S F, GOODMAN J. An empirical study of smoothing techniques for language modeling[J]. Computer Speech & Language, 1999, 13(4): 359–393.

[26] UEFFING N, MACHEREY K, NEY H. Confidence measures for statistical machine translation[C] // In Proc. MT Summit IX. 2003.

[27] ZHANG L. Maximum entropy modeling toolkit for python and c++[J], 2004.

[28] CHANG C-C, LIN C-J. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27.

[29] AGARWAL A, AKCHURIN E, BASOGLU C, et al. An Introduction to Computational Networks and the Computational Network Toolkit: MSR-TR-2014-112[R/OL]. 2014. http://research.microsoft.com/apps/pubs/default.aspx?id=226641.

[30] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C] // Advances in neural information processing systems. 2013: 3111–3119.

[31] GERMANN U. Dynamic Phrase Tables for Machine Translation in an Interactive Post-editing Scenario[C] // AMTA 2014 Workshop on Interactive and Adaptive Machine Translation, Vancouver, BC, Canada. 2014 : 20 – 31.

[32] ZENS R, NEY H. Efficient Phrase-Table Representation for Machine Translation with Applications to Online MT and Speech Translation.[C] // HLT-NAACL. 2007 : 492 – 499.

[33] MARIE B, et MACHINA L, MAX A. Touch-Based Pre-Post-Editing of Machine Translation Output[J], 2015.

[34] GOOD I J. The population frequencies of species and the estimation of population parameters[J]. Biometrika, 1953, 40(3-4) : 237 – 264.

[35] KUHN R, DE MORI R. A cache-based natural language model for speech recognition[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1990, 12(6) : 570 – 583.

[36] BROWN P F, PIETRA V J D, PIETRA S A D, et al. The mathematics of statistical machine translation: Parameter estimation[J]. Computational linguistics, 1993, 19(2) : 263 – 311.

[37] ARNOLD D. Machine translation: an introductory guide[M]. [S.l.] : Blackwell Pub, 1994.

[38] KNIGHT K. Decoding complexity in word-replacement translation models[J]. Computational Linguistics, 1999, 25(4) : 607 – 615.

[39] ZHANG H P, LIU Q. ICTCLAS[J]. Institute of Computing Technology, Chinese Academy of Sciences: http://www. ict. ac. cn/freeware/003_ictclas. asp, 2002.

[40] ZHANG H-P, YU H-K, XIONG D-Y, et al. HHMM-based Chinese lexical analyzer ICTCLAS[C] // Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17. 2003 : 184 – 187.

[41] CHENG S, HUANG S, CHEN H, et al. PRIMT: A Pick-Revise Framework for Interactive Machine Translation[C] // The 15th Annual Conference of the North

American Chapter of Association for Computational Linguistics: Human Language Technologies. 2016.

[42] CIVERA J, CUBEL E, LAGARDA A L, et al. From Machine Translation to Computer Assisted Translation using Finite-State Models.[C] // EMNLP. 2004 : 349 – 356.

[43] BENDER O, HASAN S, VILAR D, et al. Comparison of generation strategies for interactive machine translation[C] // Proceedings of the 10th Annual Conference of the European Association for Machine Translation (EAMT 05). 2005 : 33 – 40.

[44] MOHIT B, LIBERATO F, HWA R. Language Model Adaptation for Difficult to Translate Phrases[C] // Proceedings of the 13th Annual Conference of the EAMT. 2009 : 160 – 167.

[45] PETROV S, BARRETT L, THIBAUX R, et al. Berkeley parser[J]. GNU General Public License, 2010, 2.

[46] ORTIZ D. Advances in fully-automatic and interactive phrase-based statistical machine translation[D]. [S.l.] : Universitat Politècnica de València, 2011.

[47] ALABAU V, SANCHIS A, CASACUBERTA F. Improving on-line handwritten recognition using translation models in multimodal interactive machine translation[C] // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2. 2011 : 389 – 394.

[48] TOSELLI A H, VIDAL E, CASACUBERTA F. Interactive Machine Translation[G] // Multimodal Interactive Pattern Recognition and Applications. [S.l.] : Springer, 2011 : 135 – 152.

[49] ALABAU V, BUCK C, CARL M, et al. Casmacat: A computer-assisted translation workbench[C] // Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. 2014 : 25 – 28.

[50] SANCHEZ-CORTINA I, ANDRÉS-FERRER J, SANCHIS A, et al. Speaker-adapted confidence measures for speech recognition of video lectures[J]. Computer Speech & Language, 2016, 37 : 11 – 23.

致　　谢

转眼间，三年的硕士学习生活即将进入尾声，即将开始新的生活，此时的我感慨良多。从三年前进入南京大学计算机科学与技术系，自然语言处理研究组以来，我成长了许多，也改变了许多，需要感谢的人太多，他们在我的工作生活中给了我太多帮助，在此我对我的家人、老师、同学表达我最诚挚的感谢！

首先我要感谢实验室的陈家骏教授。大一时第一次听陈老师的课，我便喜爱上了这位治学严谨，求真务实，和蔼可亲的老师。陈老师在工作和生活中给予了我非常大的帮助，平时为人幽默风趣，工作认真负责，是同学们学习的好榜样。

其次，我要感谢我硕士三年间，在工作和生活上都无比关心我的黄书剑老师。四年前，我的本科毕业设计就是在黄书剑老师的指导下顺利完成的，在那时，黄书剑老师就给予了我很大的帮助，是他带领我进入自然语言处理实验室；四年后，黄书剑老师再次指导我的硕士毕业论文，还是他见证我即将离开实验室，在这里说一声：辛苦了！在进入自然语言处理实验室时，黄老师悉心教导，讲解相关知识，使我获益良多；在日后的学习工作中同样给予了我莫大的帮助。

同时，我也要感谢戴新宇，尹存燕，沈思，李斌，张建兵等实验室的老师们。他们在我的工作，生活中给予了很大的帮助和有价值的建议，使我能够在学习工作中走得更好。感谢赵迎功、张颖杰、周浩、陈华栋四位博士师兄师姐给予我工作上的帮助和建议。我要特别感谢程川，黄家君，胡光能，牛力强，周逸初，我们亲如兄弟，在这离别之际，我希望你们在今后的人生中越走越好！另外，我还要感谢尚迪，郁振庭，周启元，季红洁，李小婉，汤莲瑞，王韶杰，以及实验室的其他兄弟姐妹们！

最后，我要感谢我的父母，是他们的鼓励和支持才使得我能够顺利完成我的学业，没有你们就不会有今天的我。希望你们永远幸福，身体健康！

附录

攻读硕士学位期间完成的学术成果

1. CHENG S, HUANG S, CHEN H, et al. PRIMT: A Pick-Revise Framework for Interactive Machine Translation[C]//The 15th Annual Conference of the North American Chapter of Association for Computational Linguistics: Human Language Technologies. 2016.

攻读硕士学位期间申请的专利

1. 黄书剑, 程善伯, 戴新宇, 陈家骏, 张建兵. 一种计算机中限定翻译片段的交互式翻译方法. 国家发明专利 (已公开). 申请/专利号: 201510330285.X.

攻读硕士学位期间获得的奖项

1. 2015 –荣获南京大学二〇一五年“计算机科学与技术系研究生优秀奖学金”
2. 2015 –荣获南京大学二〇一五年“计算机科学与技术系优秀研究生”

学位论文出版授权书

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：_____

_____年_____月_____日

论文题名	短语翻译系统中的交互翻译研究				
研究生学号	MF1333006	所在院系	计算机科学与技术系	学位年度	2016
论文级别	□ 硕士 ☑ 硕士专业学位 □ 博士 □ 博士专业学位 （请在方框内画勾）				
作者 Email	chengsb@nlp.nju.edu.cn				
导师姓名	陈家骏 教授 黄书剑 助理研究员				

论文涉密情况：

☑ 不保密

□ 保密，保密期：_____年_____月_____日 至 _____年_____月_____日

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。