基于短语统计机器翻译解码算法的研究与实现

罗 毅,李 淼,朱 鉴,胡冠龙
LUO Yi,LI Miao,ZHU Jian,HU Guan-long

中国科学院 合肥智能机械研究所,合肥 230031
中国科学技术大学 信息科学技术学院,合肥 230027
Institute of Intelligent Machines,Chinese Academy of Sciences,Hefei 230031,China
School of Information Science and Technology,University of Science and Technology of China,Hefei 230027,China
E-mail:luoyi@mail.ustc.edu.cn

LUO Yi,LI Miao,ZHU Jian,et al.Research and implement of phrase-based statistical machine translation decoding algorithm.Computer Engineering and Applications,2007,43(30):171-173.

Abstract: Decoder is the important part of statistical machine translation research.Based on phrase-based statistical machine translation,a dynamical programming beam search decoding algorithm is put forward combining multi futures model using log-liner model approach.Introduces the implement of the decoder in detail based on this algorithm,and analyzes to the translation speed and precision at last.

Key words: statistical machine translation;decoding algorithm;beam search;feature model

摘 要: 解码器是统计机器翻译研究的关键部分。在基于短语的统计机器翻译的基础上,结合对数线性模型的思想加入多个特征模型,研究了一种动态规划的柱搜索解码算法。详细介绍此算法在解码器中的具体实现,并对翻译速度和精度作了分析。

关键词: 统计机器翻译;解码算法;柱搜索;特征模型
文章编号: 1002-8331(2007)30-0171-03 **文献标识码:** A **中图分类号:** TP301.6

1 引言

近年来,由于基于统计的机器翻译方法在国际翻译评测中的评价越来越高,基于统计的机器翻译方法的逐渐成为机器翻译领域中的研究热点。

统计机器翻译认为翻译过程就是从给定源语言句子 $f^J=f_1\cdots f_i\cdots f_J$ 的所有目标语言句子 $e^I=e_1\cdots e_i\cdots e_I$ 中寻找概率最大的句子 $\hat{e}^I$ 作为最佳译文。最初的信道模型[1]中将翻译过程表示为:

$$\hat{e}^I=\arg\max_{e^I}\{\Pr(e^I|f^J)\}= \tag{1}$$

$$\arg\max_{e^I}\{\Pr(e^I)\Pr(f^J|e^I)\} \tag{2}$$

$\Pr(e^I)$ 为目标语言模型,反映目标语言句子的质量;$\Pr(f^J|e^I)$ 为翻译模型,体现源语言句子到目标语言句子的互翻译可能性;argmax 是搜索最大概率 e^I 的算子,这个搜索过程在统计机器翻译中又称为解码过程。在信道模型中统计翻译的质量很大程度上决定于语言模型和翻译模型好坏。后来发展出对数线性模型[2],该模型表示为

$$\Pr(e^I|f^J)=\exp\left(\sum_m^M\lambda_m h_m(e^I,f^J)\right)Z(f^J) \tag{3}$$

其中 $Z(f^J)$ 为一个标准常量,此时翻译过程又可以表示为:

$$\hat{e}^I=\arg\max_{e^I}\left(\sum_m^M\lambda_m h_m(e^I,f^J)\right) \tag{4}$$

$h_m(e^I,f^J)$ 为 e^I 和 f^J 之间的特征模型,λ_m 为特征模型的权重因子。解码器负责搜索出具有各个特征模型的最大加权评分值的目标句子作为翻译译文。好的解码算法可以提高翻译速度和翻译的质量。

统计机器翻译最初采用的是基于词的逐词翻译方法,该方法对多个词语之间上下文关系反映较差。后来研究出基于短语的方法,该方法将源句子切分为多个短语并进行短语间的翻译。本文研究的解码算法解决了在考虑源语言短语翻译位置重排情况下,快速准确地搜索出最大特征模型评分的目标语言句子的问题。

2 特征模型

在对数线性模型方法中,所有的特征模型可以轻松地加入到系统中来。原来的信道模型只是对数线性模型的一个特例,通过加入多个反映双语语言特征的模型,提高翻译的忠实度和流利度。

2.1 短语翻译模型

短语翻译模型是对数线性模型中唯一必须的模型,它反映了源/目标语言短语之间的互译信息。本文利用短语抽取算法[3]抽取出短语级对齐的源/目标语言短语对。短语对为同时满足双语词语对齐矩阵的连续词语集。短语翻译模型表示为:

基金项目: 中国科学院知识创新工程重要方向项目(No.KGCX2-SW-511)。

作者简介: 罗毅(1983-),男,硕士研究生,研究方向:机器翻译、解码算法;李淼(1955-),女,研究员(通信作者),博士生导师,研究方向:人工智能与农业知识工程;朱鉴(1982-),男,硕士研究生,研究方向:机器翻译;胡冠龙(1982-),男,硕士研究生,研究方向:词性标注、机器翻译。

$$\Pr(e|f)=\prod_{i=1}^{I}\Phi(e_i|f_i) \tag{5}$$

2.2 目标语言模型

目标语言模型用于评价翻译译文的质量。本文加入基于N-gram的目标语言模型(N=3)

$$\Pr(e)=\sum_{i=2}^{I}p(e_i|e_{i-1},e_{i-2}) \tag{6}$$

2.3 扭曲模型

在允许对源语言句子翻译进行短语翻译位置重排时,扭曲模型考虑短语重排的费用。

$$\Pr(e,f)=\exp(-\sum_{i=1}^{I}d_i) \tag{7}$$

d_i的大小为翻译时第i个源短语的第一词语的位置与i-1个源短语最后一个词语位置的差值加1。d_i反映了翻译过程源语言句子短语的位置扭曲幅度。

2.4 词语惩罚模型

为了防止目标语言句子过长,通过加入词语惩罚模型对短句子进行补偿。加入的词语惩罚模型表示为:

$$\Pr(e)=\exp(I) \tag{8}$$

其中I为译文句子长度。

3 解码算法的设计与实现

在解码过程中采用柱搜索算法进行搜索,它是一种动态规划的算法。柱搜索算法可以在接近全局的空间进行有效地搜索,应用剪枝策略可以灵活的在翻译的速度和精度上取得折衷。

3.1 核心算法

柱搜索算法根据动态规划的思想,在搜索过程中不断地扩展出所有可能的翻译状态,直到翻译完成。这种翻译状态称作假设,每个假设都代表一个部分翻译结果。表1中列出假设包含的主要信息。

表1 翻译假设包含的信息

项 名	说 明
PrevHypo	指向父假设的链接,用于回溯
OtherHypoes	包括所重组的兄弟假设链接组,用于保存N-best路径分支
TrgPhrase	假设翻译的目标短语
TranslateMap	源语言句子已翻译词语的位置映射,用于进行判断以扩展新假设
TrgNgram	前n-1个已翻译的目标语言单词,用于计算目标语言模型评分
SrcPosRange	已翻译的源语言短语在句子中的位置范围,用于计算扭曲模型评分
TotalScore	假设的总评分值,用于评估假设

如图1所示,柱搜索算法从一个没有作任何翻译的初始假设开始扩展,每一次扩展通过翻译源语言句子中任意一个未翻译的短语来生成新的假设。

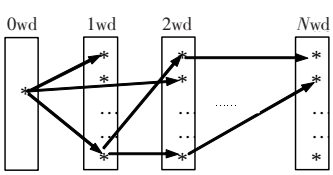

图1 假设扩展示意图

新的假设根据其所翻译的源语言单词的个数放到相应的假设栈中。假设栈用于存放翻译相同源语言单词个数的所有假设。依次扩展所有的假设栈,直至翻译完成。对单词个数为nf源语言句子的解码算法描述如下:

(1)分析源语言句子,构建翻译备选项列表;

(2)构建将来评分表;

(3)初始化假设栈hypoStack[0…nf];

(4)生成初始假设加入到假设栈hypoStack[0]中;

(5)遍历i从0到nf-1的所有假设栈hypoStack[i];

(6)对hypoStack[i]进行剪枝;

(7)遍历hypoStack[i]中的每一个假设hypo;

(8)查找hypo所有可用的翻译备选项进行逐个扩展,每次扩展生成一个新的假设newhypo;

(9)计算newhypo的TotalScore值,根据newhypo翻译的源语言单词个数加入到相应假设栈中;

(10)最后从假设栈hypoStack[nf]中的最佳假设开始回溯得到翻译译文。

在(1)中预先从短语翻译模型寻找出源语言句子的所有可用翻译短语对,避免了在进行假设扩展时对整个短语翻译模型的重复搜索。

在(9)中计算假设的TotalScore的时候,不仅要计算假设当前的各个加权评分值,还要加上通过查询(2)生成的将来评分表估计的假设将来评分[5],使得总评分值能充分反映假设的质量。

在(10)中还可以从假设栈hypoStack[nf]中回溯n个最佳(N-best)的候选目标句子,以获得多个候选译文。

3.2 构建翻译备选项列表

翻译备选项用于记录短语互译信息,主要包括源/目标短语、翻译评分值、目标语言短语的语言模型评分值。通过事先构建源句子中所有可能翻译备选项,提高假设扩展时翻译信息获取速度。

构建翻译备选项列表时,收集短语翻译模型中所有源短语与源语言句子匹配的翻译信息,并计算出它们的模型评分值。

翻译备选项列表的数目越多,假设扩展次数也会越多。为了减少假设扩展的次数提高翻译速度可以对翻译备选列表进行裁减。通过限制翻译备选项列表的大小和设定翻译选项评分阀值对评分值较差的翻译备选项进行裁减。

3.3 构建将来评分表

将来评分估计假设未翻译部分的评分。由于未翻译部分的扩展过程和扩展生成的译文不能确定,从而无法对扭曲模型和词语惩罚模型进行将来评分的估计,将来评分只能对语言模型和短语翻译模型的评分估计。

将来评分为由假设未翻译部分的所有可用翻译备选项所构成的翻译路径中最大评分值。为了提高假设将来评分的计算速度,事先构建一个将来评分表,目的把每次进行计算将来评分时对翻译选项的搜索比较过程简化为一个查表过程。

表中类似"i-j"的数字表示从源语言句子的第i到j位置部分的将来评分值$F(i,j)$,其计算如下:

$$F(i,j)=\max(c(i,j),\max_{\{a|i<a\leqslant j\}}F(i,a-1)+F(a,j)))$$

式中$c(i,j)$为源短语为源句子中从i到j位置短语的所有翻译备选项中最大评分值。计算时按表从上至下,从右至左的顺序计算将来评分表中的值,这样可以充分利用已生成的评分值。

如图 2，查表时对翻译了 2、3 位置的源句子的单词的假设,只需要将表中的 $F(0,1)$和 $F(4,4)$项求和就可以得到其将来评分。

0–0				
0–1	1–1			
0–2	1–2	2–2		
0–3	1–3	2–3	3–3	
0–4	1–4	2–4	3–4	4–4

图 2　将来评分表

3.4　假设栈的剪枝策略

在假设扩展时,若对扩展的源语言句子短语的位置重排没有任何限制,会使整个搜索变为一个 N–P 问题[4]。为此,一般通过限制源语言句子短语在翻译过程的位置扭曲的范围简化问题。

与此同时,若不对假设栈进行剪枝,每次假设栈扩展后都会使后继栈的大小呈指数增长。为了缩小假设栈搜索空间,采用了 3 种剪枝策略:

(1)假设的重组。将假设栈中 TrgPhrase、TranslateMap、TrgNgram 相同的假设进行合并,取它们中评分最大的假设。假设重组由于只合并了相同的已扩展路径的假设,它不会影响翻译结果。

(2)假设栈大小剪枝。进行剪枝时,若假设栈中假设的数目超过设定的最大值,将假设栈中评分低的假设从假设栈中删除以限制假设栈下次扩展时假设的数目。

(3)假设的阈值剪枝。设定一个阈值,表示假设栈中最低假设评分与最高假设评分之间的最大差值。进行剪枝时,删除评分超出这个阈值的假设。

3.5　N–best 回溯

N–best 回溯用于从最后一个假设栈 hypoStack[*nf*]中回溯出前 n 个评分最好的目标句子。这样就可以获得 n 个候选的翻译译文。

在 N–best 回溯时,首先需要保存翻译假设的各分支路径。如图 3 所示。

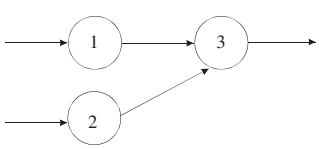

图 3　N–best 的路径保存

图中 1、2、3 表示假设,箭头代表假设的扩展顺序。假定 2 的评分低于 1,1、2 满足假设重组条件。在进行假设重组的时候,把 2 直接从假设栈删除,这样在 N–best 回溯的时候只能从 3 往 1 回溯,丢失了 2 的路径信息。因此需要进行 N–best 回溯时，把 2 作为分支保存到 1 的 OtherHypoes 项中去，以便从 1 中进行分支路径的搜索。

在假设扩展的过程时，由于保存了假设的分支路径信息，所以生成 N–best 路径的过程可以视为一个从假设栈 hypoStack[*nf*]中的前 n 个最佳假设为树根的 n 棵子树构成的森林中寻找出 n 条最佳分支路径过程。

N–best 路径回溯算法描述如下:

① 对假设栈 hypoStack[*nf*]中评分最好的前 n 个假设进行回溯(不包括对 OtherHypoes 的回溯)并将主干路径加入到路径集合 PathColl 中;

② 初始化 N–best 路径集合 PathSet;

③ While PathSet 的路径数小于 N do{

弹出 PathColl 中最佳路径 curBestPath;

if curBestPath 是分支路径{

定位 curBestPath 的分支点;

从分支点开始回溯子分支路径，将所有子分支路径加入到 PathColl 中;

}

else {

回溯 curBestPath 的所有分支路径并加入到 PathColl 中;

}

弹出 PathColl 中当前最佳路径到 PathSet 中;

}

4　实验结果

实验使用哈尔滨工业大学信息检索研究实验室提供的“HIT–IRLab–10 万汉英双语句对”语料库。以英语为目标语言利用 SRILM[6]N–gram 语言模型训练工具训练 3–gram 的目标语言模型。在 GIZA++[7]工具训练词语对齐的基础上根据文献[3]的短语抽取算法抽取短语,并用极大似然法进行短语评分构建汉英短语翻译模型。

选用 200 句长度为 15 到 20 个词语的源语言句子作为测试数据，比较了假设栈剪枝策略下解码器的翻译性能和精度。表 2 为假设栈大小剪枝实验数据,表中错误率表示搜索过程中丢失假设栈空间最大评分译文的出错率。

表 2　假设栈大小剪枝

栈大小	500	100	50	20	10
时间/s	249	213	207	196	192
错误率/%	1	3	7	10	19

由表 2 可见在假设栈大小为 100 时,翻译的精度和性能取得比较满意的结果。表 3 为假设栈阈值剪枝实验室数据。

表 3　假设栈的阈值剪枝

栈阈值	0.000 1	0.001	0.01	0.1	0.3
时间/s	246	213	198	167	124
错误率/%	0.5	3	6	13	24

由表 3 可见假设栈阈值取 0.001 时,翻译的精度和性能取得较好结果。

通过实验表明,柱搜索算法通过灵活地调整假设栈的各剪枝系数可以在翻译的速度和精度上取得的满意的结果。

5　结束语

解码算法是统计机器翻译的关键部分,解码的精度和效率一直是研究解码算法的重点。本文在基于短语的统计机器翻译基础上,加入了多个特征模型,深入研究了在进行解码过程柱搜索算法的设计与实现。通过构建翻译备选项和将来概率表的方法提高扩展速度。实验表明柱搜索算法在能够很好的在翻译性能和精度上取得折衷。同时通过增加的 N–best 功能,使得解码器支持多个候选译文结果提高了翻译的灵活性。

(收稿日期:2007 年 2 月)

(下转 178 页)