

文章编号: 1003-0077(2007)03-0028-06

单纯形算法在统计机器翻译 Re-ranking 中的应用

付雷^{1,2}, 刘群²

(1. 中国科学院 研究生院 北京 100049;

2. 中国科学院 计算技术研究所 多语言交互技术评测实验室, 北京 100080)

摘要: 近年来, discriminative re-ranking 技术已经被应用到很多自然语言处理相关的分支中, 像句法分析, 词性标注, 机器翻译等, 并都取得了比较好的效果, 在各自相应的评估标准下都有所提高。本文将统计机器翻译为例, 详细地讲解利用单纯形算法(Simplex Algorithm)对翻译结果进行 re-rank 的原理和过程, 算法的实现和使用方法, 以及 re-rank 实验中特征选择的方法, 并给出该算法在 NIST-2002(开发集)和 NIST-2005(测试集)中英文机器翻译测试集合上的实验结果, 在开发集和测试集上, BLEU 分值分别获得了 1.26% 和 1.16% 的提高。

关键词: 人工智能; 机器翻译; discriminative re-ranking; 单纯形算法; 统计机器翻译

中图分类号: TP391

文献标识码: A

Re-ranking for Statistical Machine Translation Using Simplex Algorithm

FU Lei^{1,2}, LIU Qun²

(1. Graduate School of Chinese Academy of Sciences, Beijing 100049, China;

2. Multilingual Interaction Technology Laboratory Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: Recently, discriminative re-ranking technique has been applied in many fields relative to NLP (Natural Language Processing), such as parsing, pos-tagging, and machine translation etc., and performs very well. We will take SMT as an example to explain how to re-rank the translation candidates using Simplex Algorithm in detail and give the experiment results on NIST-2002(development set) and NIST_2005(test set) Chinese-to-English test sets. Our experiments show that we can gain significant improvements in BLEU by re-ranking. It can provide 1.26% absolute increase in development set and 1.16% absolute increase in test set.

Key words: artificial intelligence; machine translation; discriminative re-ranking; simplex algorithm; SMT

1 前言

所谓 discriminative re-ranking 就是针对某个评估标准对机器翻译程序输出的多个结果进行重新选择, 致力于从中选择出使该标准达到最优时的翻译结果。

以统计机器翻译为例, 实现 discriminative re-ranking 的基本做法如下: 首先由机器翻译系统对开发集和测试集中的每个句子都生成 N 个候选译

文, 称之为“N-best list”, 然后根据需要选取合适的译文特征, 生成所有候选译文的特征分数, 接着, 就可以利用某种 re-ranking 算法针对机器翻译相应的评估标准, 如 BLEU、NIST, 在开发集上训练出各特征在相应评估标准达到最优值时的权重, 最后, 利用开发集上训练得到的各特征的权重直接从测试集的 N-best list 中选出每个句子的最佳译文。

discriminative re-ranking 技术由国外的研究人员首先提出并广泛应用于句法分析中, 同时取得了很好的效果, 机器翻译的研究人员受其启发, 将其应用在机器翻译系统的后处理阶段, 以期获得翻译系

收稿日期: 2006-09-04 定稿日期: 2007-01-19

基金项目: 国家自然科学基金资助项目(60573188)

作者简介: 付雷(1986—), 硕士生, 主要研究方向是自然语言处理、统计机器翻译。

统在性能上的提高。这项技术之所以能够在统计机器翻译后处理中得到重要应用, 还在于:

1. 当前的统计机器翻译系统很多是基于对数线性模型(Log-linear Model)的, 而该模型建立时一个很重要的问题就是模型特征的选择, 最好不要选计算过于复杂的特征, 那会使模型的训练和解码的复杂度让人不可忍受, 有了 re-ranking 技术, 就为在模型建立时采用相对简单的特征, 而在 re-ranking 阶段使用比较复杂的特征提供了条件, 既可降低模型的复杂度, 又能取得较好的整体性能;

2. discriminative re-ranking 技术使得为模型添加一些有用的全局特征成为可能, 像候选译文从句法结构特征, 从而使统计机器翻译系统也可以利用一些句法信息来提高系统的性能。

目前, 国际上针对 discriminative re-ranking 提出了多种可行算法, 这些算法大致上可以分为两类: 一类是将 re-ranking 问题看作类似于分类问题, 通过改造现有的机器学习算法来进行 re-rank, 具有代表性的算法有感知机(Perceptron)算法^[9], 类感知机(Perceptron-Like)算法^[7], 支持向量机(SVM)算法^[6]等; 另一类是将 re-ranking 问题看作是多维求极值问题, 具有代表性的算法有鲍威尔(Powell)算法, 最小分类错误率算法(MCE)^[3], 单纯形算法^[2]等。

在这些算法中, 应用比较广泛的是类感知机算法^[7], 最小分类错误率算法(MCE)^[3], 和单纯形算法^[2]。

类感知机算法又叫做多偏差感知机算法(Multi-Bias Perceptron Algorithm), 是 Shen 针对机器翻译结果进行 re-rank 这一特殊应用通过改造传统的感知机算法提出来的。由于对机器翻译系统产生的结果进行评估时每个句子都有多个参考译文, 而不像句法分析可以有一个唯一的最好的参考结果, 因此, 对于一个句子的多个候选译文很难明确的区分出好与坏, 因为可能某个候选译文和某个参考译文很接近, 而另一个候选译文则和另一个参考译文很接近, 但两个候选译文之间相差较大, 这不能说明哪个译文质量更好, 于是, Shen 提出了一种解决的方法, 他将每个汉语句子的 N-best list 中的候选按照对 BLEU 分值的影响大小重新排序, 然后将前 r 个看作好的翻译, 后 k 个看作差的翻译来训练分类的超平面, 最后每个汉语句子的超平面将好与差的候选译文分开, 并且要求这些超平面相互平行, 具有相同的特征向量 w, 感知机算法训练的目的就是要找到这个 w, 有了 w 之后就可以直接从

测试集的 N-best list 中重新选择一个最好的候选译文作为最终翻译结果了。

最小分类错误率算法, 目前广泛应用于统计机器翻译系统模型构建的最小错误率训练中, 其基本思想和实现方法在文献[1]和[3]中有详尽的描述。

对于单纯形算法, 意大利的 ITC-irst(The Center for Scientific and Technological Research)在参加 IWSLT-2005 (the International Workshop on Spoken Language Translation)评测的机器翻译系统^[2]中曾采用过, 用来对其统计机器翻译系统产生的候选译文进行 re-rank, 并取得了很好的效果, 但其对应应用该算法的基本原理, 实现细节和特征选择等关键部分都未详细讲解, 本文将结合实验详细介绍上述几方面, 详细地介绍单纯形算法在机器翻译系统 re-ranking 中的应用。

下面, 我将分以下几个部分来阐述:

(1) 基于对数线性模型的统计机器翻译系统中 re-ranking 的基本原理;

(2) 如何利用单纯形算法进行 re-rank 以及 re-rank 中单纯形算法的实现;

(3) 利用单纯形算法进行 re-rank 实验中所选特征的介绍以及和利用最小分类错误率算法(MCE)进行 re-rank 的实验结果比较。

2 基于对数线性模型的统计机器翻译系统中 re-ranking 的基本原理

根据对数线性模型的原理公式:

$$p_{\theta}(e|f) = \frac{e^{\sum_{n=1}^N \theta_n \times h_n(e, f)}}{\sum_{k=1}^{|E|} e^{\sum_{n=1}^N \theta_n \times h_n(e, f)}} \quad (1)$$

其中, θ_n 为特征的权重, $h_n(e, f)$ 为特征分数, $|E|$ 为候选译文的个数。

我们要找的译文需满足:

$$e^* = \arg \max_e p_{\theta}(e|f) \quad (2)$$

因为公式(1)中的分母是一个归一化的因子, 对于每个句子的所有译文相当于是一个常数, 所以公式(2)可以等价的表示为下面的这种形式:

$$e^* = \arg \max_e \sum_{n=1}^N \theta_n \times h_n(e, f) \quad (3)$$

亦即

$$e^* = \arg \max_e \left(\sum_{n=1}^N \theta_n \times h_n(e, f) \right) \quad (4)$$

从公式(4)可以看出, 每个句子的最佳译文即为

该句子的所有候选译文中各特征分数乘以相应权重的和值中最大的那个,每一个候选译文 e 都对应一个特征分数乘以权重的和值,即:

$$e \Rightarrow \sum_{n=1}^N \theta_n \times h_n(e, f) \quad (5)$$

所以,当我们调整了某个或某些特征对应的权重时,则每个候选译文 e 对应的特征分数乘以相应权重的和值就会发生改变,相应的对每个句子的最佳译文的选择也会随之发生变化,而对最佳译文选择的变化,反映到机器翻译的评估标准上,如 BLEU,将导致 BLEU 分值在两次选择之间产生差异,可能升高也可能降低,当然,我们做 re-ranking 的目的就是要让这种差异向好的方向发展,即 BLEU 值升高的方向,于是,我们假定评估标准 BLEU 和候选译文 e 的选择之间潜在的存在着某种函数关系:

$$BLEU = f\left(\sum_{n=1}^N \theta_n \times h_n(e, f)\right) \quad (6)$$

我们要做的就是用算法来拟合这种关系,从而将 e 的变化反映到评估标准 BLEU 上,选择那些能够使 BLEU 值达到最优的 e 。

基于上述原理,本文将利用单纯形算法来拟合这种函数关系,达到优化 BLEU 值的目的。由于单纯形算法是用来求函数极小值的,而我们的目的是要求使 BLEU 值达到最大时的 e ,所以在算法实现时,我们将对 BLEU 值和特征值都取相反数。

3 如何利用单纯形算法进行 re-rank 以及 re-rank 中单纯形算法的实现

单纯形算法^[12],也称为 Nelder-Mead Simplex Algorithm,由 Nelder-Mead 最先提出,用来求函数在无约束条件下的多维极小值问题,又叫多维下降

单纯形算法。不同于其他求极值的算法,单纯形算法在迭代过程中不需求导,求梯度下降方向等,因此实现起来相对简单一些,收敛的速度也比较快,对于维数不是太高的(<30)求极小值的问题还是比较适合的;另外,单纯形算法不像其他的算法,它不是从一个点开始迭代,而是从 $N+1$ 个点同时开始,每次从 $N+1$ 个点中选出函数值最大,次大和最小的三个点,然后,通过算法迭代,将函数值最大的点拉向最小点,接着再重新选择三个这样的点,反复迭代,直到函数值达到一个局部最优值或达到最大迭代次数为止^[12]。

3.1 如何利用单纯形算法进行 re-rank

要利用单纯形算法对统计机器翻译输出的结果进行 re-rank,首先,要选择一些额外的特征,这些特征不同于解码器中所用到的特征,可以把它们看成是对解码器特征的一种扩充,这些特征将和解码器本身的特征一同用于对解码器输出的 N-best list 进行重新选择。在统计机器翻译 re-ranking 阶段,特征选择的自由度很大,你可以选择解码器在训练和解码时由于系统资源和算法复杂度等原因无法选择的一些特征,如在训练和解码阶段采用低元的语言模型而在 re-ranking 阶段采用高元的语言模型。另外,由于每个候选译文都是一个完整的英文句子,而不像在解码阶段是一个个的短语片断,因此在 re-ranking 阶段也可以采用像候选译文的句法结构特征这种全局性的特征。还可以利用词性信息,句子长度信息,词语对齐信息,外部词典信息等作为特征,尽可能多的利用一些语言学方面的特征,以对统计机器翻译系统的输出结果进行评估,把更符合语法和句法结构的相对好一些的句子从每个句子的 N-best list 中挑选出来。

用单纯形算法进行 re-rank 的流程如下:

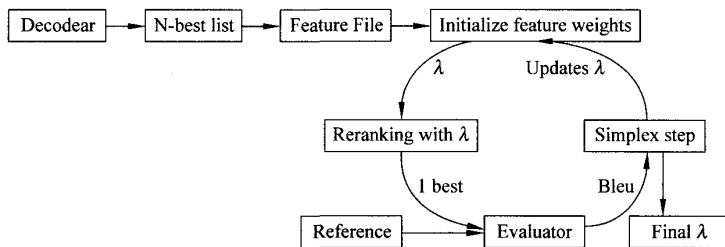


图1 re-rank 的流程

该算法的终止条件为: BLEU 分值的提高幅度小于某个容差值(如 10^{-6})或算法达到人为设定的

最大迭代次数仍不收敛时停止,同时输出达到最优值时的各特征权重,然后,直接用这组权重根据公式

(4), 利用测试集的 N-best list 的特征文件选出每个句子的最佳译文, 至此, 完成了 re-rank 的整个过程。

3.2 re-rank 中单纯形算法的实现

算法实现的伪码如下:

```
Initialization:
MaxIter=500; //Maximum number of iteration
Ftol=0.000001; //Fault-tolerance for BLEU
X[N+1][N]; //If you have N features, you
            //need a matrix of (N+1) * N
 $\alpha=2$ ; //  $\alpha$  is Expand factor
 $\beta=0.5$ ; //  $\beta$  is Contract factor
```

```
Algorithm:
Compute_BLEU ();
//利用 X 矩阵中的 N+1 组权重计算出 N+1 个
BLEU 值。
```

```
High_Mid_Low ()
//从 N+1 个 BLEU 值之中选出最大, 次大和最小的 3
个值, 并记录下它们对应的权重。
for (i = 1; i <= N+1; i++)
{
     $f_{High} = f(X_{High}) = \max(f_i) \quad 1 \leq i \leq n+1$ 
     $f_{Mid} = f(X_{Mid}) = \max(f_i) \quad 1 \leq i \leq n+1, i \neq High$ 
     $f_{Low} = f(X_{Low}) = \max(f_i) \quad 1 \leq i \leq n+1$ 
};
```

利用单纯形算法进行机器翻译结果 re-rank 的主体迭代算法如下:

```
Simplex_Iteration()
do
{
1:  $tol = 2 * (f_{High} - f_{Low}) / (f_{High} + f_{Low})$ ;
2: if (  $tol < Ftol$  ) return ;
3: if (Iteration > MaxIter) return ;
4: Compute_BLEU ();
5: High_Mid_Low ();
6: Iteration++;
7:  $X_F = 1/n * \sum_{i=1}^{N+1} X_i \quad i \neq High$ ;
8:  $X_T = 2 * X_F - X_{High}$ ;
9: if (  $f(X_T) < f(X_{Low})$  )
10:  $X_E = (1 + \alpha) * X_T - \alpha * X_F$ ;
11: if (  $f(X_E) < f(X_{Low})$  )
     $X_{High} = X_E$ ;  $f(X_{High}) = f(X_E)$ ;
12: else  $X_{High} = X_T$ ;  $f(X_{High}) = f(X_T)$ ;
13: else
14: if (  $f(X_T) \leq f(X_{Mid})$  )
15:  $X_{High} = X_T$ ;  $f(X_{High}) = f(X_T)$ ;
16: else
17: if (  $f(X_T) > f(X_{High})$  )
18:  $X_{High} = X_T$ ;  $f(X_{High}) = f(X_T)$ ;
```

```
19:  $X_C = \beta * X_{High} + (1 - \beta) * X_F$ ;
20: if (  $f(X_C) > f(X_{High})$  )
21:  $X_i = 1/2 * (X_i + X_{Low})$ ;
22:  $f_i = f(X_i)$ ;
23: else
24:  $X_{High} = X_C$ ;  $f(X_{High}) = f(X_C)$ ;
25: else
26:  $X_{High} = X_T$ ;  $f(X_{High}) = f(X_T)$ ;
} while (1)
```

该迭代算法的核心就是在每次迭代过程中都把单纯形中函数值达到最大的点替换掉, 采取的方法是先求出函数值达到最大的单纯形顶点的对称点 X_T , 这个步骤称为反射, 之后, 将对单纯形在某个方向上进行扩展以加大步长, 当到达“谷底”时, 单纯形将作横向收缩, 且自行拉向最低点附近。

其中, 需要说明的是算法初始化时, 假设共有 N 个特征, 其中解码器的特征数为 D , 额外的用于 re-rank 的特征数为 R , 那么按单纯形算法的要求, 需要初始化一个 $(N+1) \times N$ 维矩阵, 每一维是 N 元向量(对应 N 个特征的权重)。矩阵的第一行 p_1 的 N 个值应按这样设定: 前 D 个解码器的特征权重应设定为解码器在开发集上训练得到的权重, 即产生开发集上 1-best 时所使用的权重, 后 R 个对应于 re-rank 特征的权重都置为 0。矩阵的其余 N 行按下面的公式^[12]从第一行中派生得到:

$$p_i = p_1 + \lambda \times E_i$$

E_i 为 N 元单位向量, λ 为对特征权重调整力度的估值, 即如果你认为该维特征较之其他维应该起到更大的作用, 就可以将该维的 λ 设的大一些, 反之就小一些, 每一维对应的 λ 可以相同, 也可以不同。之所以要这样来设置初始值, 是为了更好的保证单纯形算法有一个好的基点, 第一行权重的设置得到的结果(BLEU)为解码器在开发集上训练得到的最好结果, 以此为基点, 单纯形算法调整后至少不会比这个基点差, 也就是说, 保证了算法迭代一定是在向好的方向发展。

4 利用单纯形算法进行 re-rank 实验中所选特征的介绍以及利用最小分类错误率算法(MCE)进行 re-rank 的实验结果比较

实验中用 NIST^①-2002 (878 个汉语句) 做开

① NIST: National Institute of Standards and Technology, 美国国家技术与标准研究所, 每年举办多语种间的机器自动翻译评测。

发集,用 NIST-2005(1 082 个汉语句)做测试集,解码器采用实验室开发的孔夫子(Confucius)^[13]统计机器翻译系统,该系统采用最大熵模型,是一个基于短语^[11](Phrase-based)的统计机器翻译系统,实验中对于开发集和测试集中的每个汉语句输出 1 000 个候选英语译文,即生成 1 000-best。

进行 re-rank 实验的单纯形算法按照文中所述的算法伪码实现,最小分类错误率算法的原理和实现请参见文献[1]和[3]。

下面介绍一下进行 re-rank 实验中用到的特征及实验结果比较。在 re-rank 的特征选择方面,我一共用了以下 12 个特征:

1. 2-gram 英语语言模型特征
2. 4-gram 英语语言模型特征
3. 5-gram 英语语言模型特征
4. 2-gram 英语词性语言模型特征
5. 3-gram 英语词性语言模型特征
6. 4-gram 英语词性语言模型特征
7. 5-gram 英语词性语言模型特征
8. IBM Model 训练生成的词典特征
9. CLA 训练生成的词典特征
10. 英汉句子长度比例的特征
11. 局部 N-gram 出现频率的特征
12. 句首标点判断特征(2 值特征)

其中,英语词性的语言模型同普通语言模型的训练方法相同,就是将英语语料标注词性后,将词性序列提取出来,单独训练一个 5 元词性语言模型;IBM Model 的特征是利用 GIZA++ 训练生成的词典,然后按照 IBM Model-1^[10] 计算翻译概率的公式算出概率值并取对数,CLA (Competitive Linking Algorithm)^[5] 的特征计算同 IBM Model 特征,只是利用的词典是由 CLA 算法生成的;局部 N-gram 出现频率的特征是指对于每个汉语句子的 1 000 个英语候选译文,统计每个候选译文的 1-gram, 2-gram, 3-gram 和 4-gram 在整个 1 000 句中出现的频率,然后按照一组权重线性相加作为一个特征;句首标点判断特征是指判断英语译文的句首是否为逗号、句号、叹号等明显不能作为句首的标点符号,为 2 值特征。

在这些特征中,直观的来讲,英语语言模型的特征与句子的流利度相关联;词性的语言模型可以在一定程度上反映句子的局部片段搭配是否符合语言习惯;IBM Model 1 就是一个词的翻译模型,因此可以处理句子的丢词现象;而 N-gram 出现的频率特征则使算法倾向于选择出现频率高的 N-gram,以期

在句子片段的选择时能够达到局部的最优,选择所有句子中出现最频繁的片段来组成整句。

实际实验中,需进行多次单纯形算法和最小分类错误率算法的迭代,每一次迭代采用上一次迭代生成的权重作为初始权重,直到 BLEU 分值不再升高为止。实验以孔夫子解码器在开发集和测试集上直接输出的译文结果作为基线分数。

利用单纯形算法进行 re-rank 在 NIST-2002(开发集)上的实验结果见下表,表中每一行为单独使用每个特征时对 BLEU 值提高的贡献,最后一行为使用所有特征时对 BLEU 值提高的总贡献。

表 1 单纯形算法在开发集上的实验结果

| Feature | BLEU% | Contribution% |
|------------------|-------|---------------|
| baseline | 26.76 | |
| 2-gram lm | 26.82 | 0.06 ↑ |
| 4-gram lm | 26.80 | 0.04 ↑ |
| 5-gram lm | 27.27 | 0.51 ↑ |
| 2-gram pos lm | 26.91 | 0.15 ↑ |
| 3-gram pos lm | 27.11 | 0.35 ↑ |
| 4-gram pos lm | 27.24 | 0.48 ↑ |
| 5-gram pos lm | 27.43 | 0.67 ↑ |
| IBM Model1 | 26.88 | 0.12 ↑ |
| CLA | 27.07 | 0.31 ↑ |
| length ratio | 26.79 | 0.03 ↑ |
| N-gram frequency | 26.92 | 0.16 ↑ |
| punctuation | 26.76 | 0 ↑ |
| + all features | 28.02 | 1.26 ↑ |

在 NIST-2005(测试集)上的实验结果如下:

表 2 单纯形算法在测试集上的实验结果

| baseline | re-ranking | Contribution |
|----------|------------|--------------|
| 25.38% | 26.54% | 1.16% ↑ |

利用最小分类错误率算法(MCE)在开发集和测试集上的实验结果见下表:

表 3 MCE 算法在开发集和测试集上的实验结果

| | BLEU% | Contribution% |
|--------------------------|-------|---------------|
| Baseline(dev set) | 26.76 | |
| Dev Set(+ all features) | 27.72 | 0.96 ↑ |
| Baseline(test set) | 25.38 | |
| Test Set(+ all features) | 26.31 | 0.93 ↑ |

在表 1 中虽然有些特征单独使用时对 BLEU 值的提高不是很明显甚至没有提高, 但是如果去掉任何一个特征都没有使用所有特征时的效果好, 即每个特征对最终结果的贡献都是正向的。另外, 从表 1 也可以看出, 高元的语言模型特征对系统性能的提升很有效, 而恰恰这个特征在解码阶段由于计算量太大致使多数翻译系统不会去使用, 而以低元的语言模型特征来代替。

综合表 1, 2, 3 可以看出, 对机器翻译系统输出的 N-best 进行 re-rank 之后, 无论是在开发集还是在测试集上, BLEU 值都有比较明显的提高, re-rank 过程对孔夫子翻译系统的整体性能的确起到了促进的作用。但相比之下, 单纯形算法要比最小分类错误率算法的效果好一些, 而且从算法的收敛速度上来看, 单纯形算法的收敛速度也要稍快一些。

5 总结

文中结合实验详细地介绍了利用单纯形算法对统计机器翻译系统产生的结果进行 re-rank 的基本原理, 算法的实现细节以及 re-rank 过程中特征的选择方法, 弥补了以往文章中言之不全的几个关键方面, 同时给出了相关实验结果, 实验结果表明 re-rank 过程对统计机器翻译系统的性能的确起到了明显的提升作用, 单纯形算法在实际应用中的确具有一定的优势。

目前, 统计机器翻译系统的 BLEU 值每提高一个点都要付出很大的努力, 要么增加使用的资源, 要么寻找新的特征或改进算法, 而 discriminative re-ranking 技术作为机器翻译后处理中的一个环节, 思想简明, 算法实现比较简单, 又可以在一定程度上减轻系统训练和解码的复杂度, 同时对统计机器翻译系统的性能却有很好的提升作用, 不失为在现有条件下提高机器翻译系统性能的比较较好的一条途径。

参考文献:

- [1] Ashish Venugopal and Stephan Vogel. Considerations in Maximum Mutual Information and Minimum Classification Error training for Statistical Machine Translation [A]. In: EAMT 2005 Conference Proceedings [C].
- [2] B. Chen, R. Cattoni, N. Bertoldi, M. Cettolo, M. Federico. The ITC-irst SMT System for IWSLT-2005 [A].
- [3] Franz Josef Och. Minimum error rate training in statistical machine translation [A]. In: Proc. of ACL 2003 [C].
- [4] Franz Josef Och and Hermann Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation [A]. In: Proceedings of the 40th Annual Meeting of the ACL [C]. Philadelphia, July 2002, pp. 295-302.
- [5] I. Dan Melamed. A Word-to-Word Model of Translational Equivalence [A]. In: Proc. of 35th Conference of the Association for Computational Linguistics (ACL '97) [C]. Madrid, 1997. 490-497.
- [6] Libin Shen and A. K. Joshi. An SVM based voting algorithm with application to parse reranking [A]. In: Proc. of CoNLL 2003 [C].
- [7] Libin Shen, Anoop Sarkar, Franz Josef Och. Discriminative Reranking for Machine Translation [A]. In: Proc. HLTNAACL 2004 [C].
- [8] M. Cettolo, M. Federico, N. Bertoldi, R. Cattoni and B. Chen. A Look inside the ITC-irst SMT System [A]. In: Proceedings of the 10th MT-Summit [C]. Phuket, Thailand. 2005.
- [9] M. Collins and N. Duffy. New ranking algorithm for parsing and tagging; Kernels over discrete structures, and the voted perceptron [A]. In: Proceedings of ACL 2002 [C].
- [10] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer. The Mathematics of Statistical Machine Translation [J]. Computational Linguistics, 1993, 19(2).
- [11] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation [A]. In: Proceedings of HLTNAACL [C]. 2003. 127-133.
- [12] W. M. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing [M]. Cambridge Univ. Press, 1992.
- [13] Zhongjun He, Yang Liu, Deyi Xiong, Hongxu Hou, and Qun Liu. ICT System Description for the 2006 TC-STAR Run # 2 SLT Evaluation [A]. TC-STAR Evaluation Workshop [C]. Barcelona, Spain, June 19-21. 2006.
- [14] 刘群. 统计机器翻译综述 [J]. 中文信息学报, 2003, 19(4): 1-12.