利用单语网页挖掘辅助汉英人名反向音译

赵军，杨帆

（中国科学院自动化研究所模式识别国家重点实验室，北京 100080）

摘 要： 本文提出一种利用单语言网页挖掘辅助汉英人名反向音译的方法。该方法由两个阶段组成。第一个阶段是矫正过程，统计音译结果被划分成音节，然后将这些音节组成查询，利用基于音节的搜索过程从一个大规模 Web 词典中搜索与音译候选相似的单词，使得错误的音译候选得到纠正，从而提高召回率；第二个阶段是重排序过程，将矫正过的音译候选作为查询在 Web 中提取其上下文信息和点击率信息，然后利用 AdaBoost 分类器判断其是否是正确的音译。这个阶段可以调整每个音译候选的得分，使之更合理，从而提高音译的精确率。实验结果显示，通过矫正过程，音译的封闭测试 top-100 召回率从 72.52%提升到 85.78%，开放测试 Top-100 召回率从 41.73%提升到 59.28%。通过重排序过程，音译的封闭测试 top-5 精确率从 42.83%提升到 76.35%，开放测试 top-5 精确率从 19.69%提升到 52.19%。实验结果显示，这种方法适合于反向音译任务。

关键词： 人名音译；汉英反向音译；网页挖掘；统计翻译模型

Chinese-English Backward Transliteration Assisted with Mining Monolingual Web Pages

Jun ZHAO，Fan YANG

(National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100080)

Abstract: In this paper, we present a novel backward transliteration approach which can further assist the existing statistical model by mining monolingual web resources. Firstly, we employ the syllable-based search to revise the transliteration candidates from the statistical model. By mapping all of them into existing words, we can filter or correct some pseudo candidates and improve the overall recall. Secondly, an AdaBoost model is used to re-rank the revised candidates based on the information extracted from monolingual web pages. To get a better precision during the re-ranking process, a variety of web-based information is exploited to adjust the ranking score, so that some candidates which are less possible to be transliteration names will be assigned with lower ranks. The experimental results show that the proposed framework can significantly outperform the baseline transliteration system in both precision and recall.

key words: Name Transliteration; Chinese-English Backward Transliteration; Web Mining; Statistical Translation Model ;

1 引言

人名翻译接收一个源语言表示的人名作为输入，输出该人名以目标语言表示的翻译。在人名翻译过程中，在保持源语言和目标语言发音基本不变的原则下，调整源语言人名使之符合目标语言的语言习惯。人名自动翻译是很多跨语言应用的重要组成部分。近年来，人名音译研究受到越来越多的关注，特别是当音译涉及的两种语言的字符集差异比较大的

本文受国家 863 计划项目（2006AA01Z144）、国家自然科学基金项目（60673042）的资助

情况（例如：英文和中文这两种语言）。尽管关于中英文跨语言应用有很多，但是对这两种语言之间的自动音译目前还缺乏全面系统的研究。

由于大量的命名实体都是通过音译方式翻译的，所以过去有许多研究都致力于建立音译模型来翻译命名实体，这类方法统称为音译。机器自动音译通常分为两个方向：正向音译和反向音译。给一个双语人名对(s,t)，s代表源语言人名，t代表目标语言人名，正向音译是把s翻译成t，反向音译是把t翻译成s。例如："Clinton->克林顿"是正向翻译，"克林顿->Clinton"是反向音译。反向音译是人名音译的难点，主要困难在于：（1）传统的统计翻译模型根据语言模型来选择最可能的翻译结果，这种方法在意译任务中是有效的，但是在音译任务中效果不明显[Gao,2004]。（2）反向音译比正向音译要更难，其中一个原因是：在正向音译过程中，源语言人名中的很多不发音音节信息已经丢失了，反向音译过程要恢复出这些丢失的音节非常困难。例如：当"Campbell"正向音译为"坎贝尔"时，"p"的发音信息已经丢失了，反向音译要恢复出"p"来很困难。

在机器自动音译不能得到好的翻译效果的情况下，研究人员尝试使用网络挖掘辅助音译的方法从混合语言网页中抽取双语实体对应[Wang et al., 2004; Cheng et al., 2004; Nagata et al., 2001; Zhang et al, 2005]。但是这种方法存在以下两个主要缺陷：（1）混合语言网页资源非常有限；（2）目前的搜索引擎技术只支持用英语查询搜索汉英双语混合语言网页，不支持用汉语查询搜索汉英双语混合语言网页，因此这种方法无法应用到汉英实体翻译上。

本文对汉英人名反向音译任务进行研究，针对反向音译的难点以及统计音译方法存在的以下两个主要问题，提出挖掘英语单语网页辅助统计音译的方法。

Problem I：正向音译过程中源语言人名的不发音音节的丢失问题，我们需要利用的一定方法把丢失的信息找回来。

Problem II：统计音译方法总是根据概率来选择正确答案，但是很多情况下，正确的答案往往具有小的概率。因此，在统计音译模型之外，我们需要借助于别的手段。

本文的出发点是：对于任意一个需要反向音译的中文人名，其英文源名一定在Web的某个地方出现，我们需要做的就是利用某种方法把它从海量网页中找出来。这种方法有两个优势：（1）在Web上，英文单语网页的数量比汉英混合语言网页的数量要多得多；（2）统计音译方法得到的翻译候选往往存在音节错误或者音节丢失现象，不是一个正确的英语单词，我们的方法可以纠正这些问题，把不正确的英语单词映射到正确的英语单词上去。

本文的方法由两个阶段组成。第一个阶段中，统计音译结果被划分成音节，然后将这些音节组成查询，利用基于音节的搜索过程从一个大规模 Web 词典中搜索与音译候选相似的单词，从而提高召回率；第二个阶段中，将矫正过的音译候选作为查询在 Web 中提取其上下文信息和点击率信息，然后利用 AdaBoost 判断其是否是一个正确的音译。这个阶段可以调整每个音译候选的得分，使之更合理，从而提高音译的精确率。表 1 描述了将对中文人名"阿/a 加/jia 西/xi"反向音译为"Agassi"的过程。

表 1. 音译流示例

中文名	统计音译结果	矫正的候选	重排序结果
阿加西	aggasi	agasi	**agassi**
a jia xi	agahi	agathi	agasi
Agassi	agacy	agathe	agache

	agasie	agassi	agga
	…	…	…

实验结果显示，我们的方法可以将 top-100 召回率从 41.73% 提高到 59.28%，top-5 精确率可以从 19.69% 提高到 52.19%。

2 基于单语网页挖掘辅助汉英人名反向音译的系统框架

基于单语网页挖掘辅助汉英人名反向音译的系统框架如图 1 所示，它包括三个主要模块。

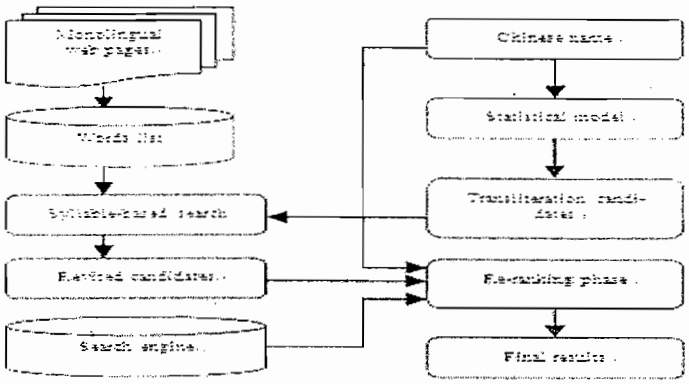

图 1. 系统框架

（1）统计音译模块：该模块接收中文拼音序列，输出 N 个统计音译结果作为音译候选。

（2）基于音节搜索的音译候选错误矫正模块：在该模块中，一个音译候选被转化为以音节为单元的查询，然后基于音节的搜索策略从一个大规模 Web 词汇表中挑选出一些相关的英语单词作为音译候选的矫正形式。其中，词汇表中的每一个单词都基于音节进行索引，搜索过程通过计算单词和查询之间的相似度来挑选相关的英语单词对音译候选进行矫正。该模块可以保证经过矫正的音译候选都是正确的英语单词。

（3）音译候选的重排序模块：在该模块中，我们在 WWW 上搜索经过矫正的音译候选并得到它们的上下文信息以及点击率信息，并利用这些信息对每个音译候选构成正确翻译的概率进行估计，从而对音译候选集合进行重排序，使正确的答案提前。

在这个框架下，基于音节搜索的音译候选错误矫正模块可以解决统计音译方法存在的以下两个问题，从而提高召回率。

（1）在基于音节的搜索中，不发音的音节被赋予很小的权重，进而通过选择最相似度英语单词，使得那些丢失的信息被恢复出来。

（2）查询扩展技术通过扩展“同义”音节，可以纠正统计音译模块中的一些音节翻译错误，从而统计音译模块丢失的正确音译结果在这个模块中被扩展出来。

而重排序模块将利用网络资源上的多种信息对音译候选进行重排序，使得正确音译结果排在候选集的前面位置，提高了精确率。

3 统计音译模型

我们使用音节作为翻译单元建立汉英人名反向音译的统计音译模型。

3.1 传统的统计翻译模型

[P. Brown et al., 1993] 提出了统计机器翻译的IBM噪声信道模型。当观察到噪声信道的输出为f= $f_1,f_2 \ldots f_n$时，可以利用公式（1）找出具有最好后验概率的句子e=$e_1,e_2 \ldots e_n$作为源语言。

$$e' = \arg\max_e P(e \mid f) = \arg\max_e P(f \mid e)P(e) \quad (1)$$

其中翻译模型 P(f|e)从汉英双语对齐的语料库中估计得到，语言模型 P(e)从英语文本中训练得到。

3.2 本文的音译模型

统计音译模型的基础是对齐方法，目前主要有两种对齐方法：基于发音单元的对齐方法（phoneme-based alignment）[Knight and Graehl, 1998; Virga and Khudanpur, 2003]和基于字母的对齐方法（grapheme-based alignment）[Long Jiang, 2007]。我们的系统采用的是从汉语拼音到英文音节的基于音节的对齐方法，使用的是[Long Jiang et al., 2007] 使用的音节划分规则。例如：中文人名“希/xi 尔/er 顿/dun”及其反向音译 “Hilton” 的对齐方案如下。“Hilton”被划分为如下音节序列“hi/l/ton”，对齐结果是“xi-hi”, “er-l”, “dun-ton”。

基于以上对齐方法可以得到以下的统计汉英反向音译模型。

$$E = \arg\max_E p(PY \mid ES)p(ES) \quad (2)$$

其中，PY 是中文拼音序列，ES 是英文音节序列，P(PY|ES)是将 ES 翻译为 PY 的概率，P(ES)是英文音节语言模型。

3.3 反向音译和翻译模型之间的差别

汉英反向音译和传统翻译之间的区别如下：（1）音译过程不需要调整音节顺序；（2）反向音译中，语言模型描述的是词语内部音节之间的关系，而传统翻译中的语言模型描述的是句子内部词语之间的关系。与后者相比，前者的作用非常不明显。我们认为，反向音译中最困难的任务是挑选正确的音节，单纯依靠统计音译很难完成这个任务。因此，本文尝试利用网络挖掘的方法改善音译的性能。

4 利用英语单语网页挖掘辅助汉英人名反向音译的方法

利用英语单语网页挖掘辅助汉英人名反向音译的方法包含两个过程：“矫正”和“重排序”。在“矫正”阶段，一个音译候选被转化为以音节为单元的查询，然后基于音节的搜索策略从一个大规模词汇表中挑选出一些相关的英语单词作为音译候选的矫正形式。其中，词汇表中的每一个单词都基于音节进行索引，搜索过程通过计算单词和查询之间的相似度来挑选相关的英语单词对音译候选进行矫正。该模块可以保证经过矫正的音译候选都是正确的英语单词。因为命名实体识别过程可能丢失一些人名，因此 Web 词表保留了 Web 语料库中的所有单词。在这个阶段，统计模型输出的音译候选中存在的错误可以在很大程度上得到矫正，从而提高了音译的召回率。在“重排序”

阶段，我们在 WWW 上搜索经过矫正的音译候选得到它们的上下文信息以及点击率信息，并利用这些信息估计每个音译候选构成正确翻译的概率，对音译候选集合进行重排序，从而使正确答案提前，提高了音译的准确率。

4.1 利用基于音节的检索进行音译候选的矫正

以下将介绍两种方法，分别解决第一章中介绍的统计音译模型存在的两个问题。

4.1.1 基于音节的检索

当我们在 Web 词汇表中搜索一个音译候选 tc_i 时，首先将它划分为音节集合 $\{es_1,es_2,\ldots..es_n\}$.，然后将这个音节集合作为查询进行基于音节的搜索。首先介绍相关术语定义。

- 词汇表 $T=\{t_1,t_2\ldots t_k\}$ 是音节的有序集合，其中每个音节被看作是词汇项。
- 拼音集合 $P=\{py_1,py_2\ldots py_k\}$ 是所有拼音的有序集合。
- 输入单词定义为一个音节向量 $\{es_1,es_2,\ldots..es_n\}$。

我们通过计算一个音译候选和 Web 词汇表中的每个词汇项的相似度来选择最相似的英语词语作为矫正的音译候选。输入单词 $\{es_1,es_2,\ldots..,es_n\}$ 被转化为一个向量 $V_{query}=\{t_1,t_2\ldots t_k\}$，其中 t_i 是 T 中的第 i 个词汇项。如果第 i 个词汇项在查询中没有出现，则 t_i 的值为 0。同样地，Web 词汇表也可以转化为向量表示形式。因此，查询和词汇表之间的相似度可以利用两个向量的内积来计算。

与传统信息检索模型使用 *tf* 和 *idf* 计算每个词汇项的权重不同，本文将第 i 个词汇项的权重 t_i 表示为第 i 个词汇项是否发音的概率。如果该词汇项具有很小的发音概率，则它的权重很小。这样，因为不发音的音节对于相似度的计算影响很小，统计音译模型丢失的一些不发音音节可以在检索过程中被恢复出来。相似度计算公式为：

$$Sim(query,word)=\frac{V_{query}\times V_{word}}{L_{word}/L_{py}} \qquad (3)$$

其中分子是两个向量的内积，分母是输入词的长度 L_{word} 与汉语拼音序列 L_{py} 的商。可以看出，长度短、音节匹配程度高的词语将得到大的相似度得分。

与传统信息检索问题的另外一个区别之处在于：传统的信息检索是不考虑词语顺序的，而音译任务必须考虑每个查询中的音节项的顺序。因为目前的信息检索方法不能解决这个问题，我们采用了类似计算两个单词的编辑距离的方法。在这种计算方法中，英语单词看作由音节构成，两个英语单词的编辑距离定义为将一个单词转化为另外一个单词所需要的最少编辑操作（删除/置换/增加）的次数。如果我们计算一个查询和每个单词之间的编辑距离，则时间复杂度太高。在具体实施时，我们首先在不考虑音节顺序的条件下，找出一部分和输入音译候选相似度高的 Web 单词项，然后在这部分单词项中利用编辑距离的计算方法，从中找出最相似的单词来。

4.1.2 利用音节同义项扩展挖掘“发音相似”的单词项

针对问题 2，我们利用查询扩展技术来解决。本文用到的查询扩展技术包括：

（1） 基于发音相似度的音节扩展策略

可以对应于同一个汉语拼音的音节称为发音同义项。例如：英语音节“din” 和“tin” 都可以对应于汉语拼音项 “ding”。

给定一个汉语拼音序列$\{py_1,py_2,\ldots py_n\}$作为统计音译模型的输入，每个$py_i$可以对应于一组音节$\{es_1, es_2 \ldots es_k\}$作为它的翻译。统计模型将选择最可能的一个音节作为$py_i$的翻译，而其他选择被丢弃。很多情况下，丢弃的是正确的翻译，因此由统计音译结果得到的查询往往与正确的音译差距很大，检索过程很难找到正确的音译结果。为了解决这个问题，我们对查询进行了同义项扩展，一个音节es_i是否构成拼音py_j的扩展项由它们之间的对齐概率$P(es_i|py_j)$决定。

（2） 基于音节相似度的音节扩展策略

如果对于每一个拼音，两个音节都具有相似的对齐概率，则我们把这两个音节看作是同义项。因此，如果一个音节可以构成查询，则它的同义项也可以构成查询。例如：“fea”和“fe”可以相互置换。

在计算相似度时，我们首先得到每个音节的对齐概率$P(py_j|es_k)$，然后利用公式（4）计算任意两个音节之间的距离：

$$Sim(es_j,es_k)=\frac{1}{N}\sum_{i=1}^{N}P(py_i\mid es_j)P(py_i\mid es_k) \qquad (4)$$

根据公式（4），我们可以选择最相似的N个音节作为第i个音节的扩展项。

（3） 基于音节编辑距离的音节扩展策略

以上两种音节扩展方法的不足之处是：它们完全依赖于训练集。如果某个音节在训练集中没有出现，则它不可能被扩展。为了解决这个问题，我们使用了基于编辑距离的音节扩展方法，该方法用编辑距离来衡量两个音节的相似度，其中一个音节在训练集中出现，另外一个不出现。因为编辑距离扩展与发音不是很相关，我们给这种扩展一个小的权值。在出现新的音节的情况下，这种策略将发挥作用。

（4） 以上三种策略的综合应用

我们利用线性插值方法将以上三种音节扩展策略进行综合应用。

$$S=(1-\alpha)S_{pre}+\alpha S_{sy}+\beta S_{ed} \qquad (5)$$

$$S=(1-\alpha)S_{pre}+\alpha S_{py}+\beta S_{ed} \qquad (6)$$

其中Spre是完全匹配的得分，Ssy是基于音节相似度的扩展得分，Spy是基于发音相似度的扩展得分。实验部分对其中的参数进行调节，以获得最优的性能。

4.2 利用单语Web资源对矫正后的音译候选进行重排序

在“矫正”阶段，我们已经利用统计音译结果作为线索，从大规模Web词汇项中找出了矫正的音译候选集合$\{rc_1,rc_2,\ldots,rc_n\}$，其目标是提高召回率。“重排序”阶段的目标是提升其精确率，即我们希望对音译候选集合进行重新排序，使得正确的音译排在最前面的位置。

[Al-Onaizan et al., 2002]提出了对音译候选进行重排序的方法，他的方法的不足之处是：只能对现有的候选进行重排序，不能对其中的错误进行纠正。如果候选中不存在正确的答案，则重排序过程无法得到正确的音译，即只能提升Top-5精确率，不能提升Top-100召回率。本文的工作就是希望通过矫正过程使召回率也得到提高。

我们利用 AdaBoost 框架，综合使用多种特征对音译候选集合进行重排序。AdaBoost 分类器计算一个候选构成一个人名的概率，然后我们根据这个概率对音译候选集合进行重排序。AdaBoost 使用的特征包括：

是否是人名： 首先使用 rc_i 作为查询来搜索单语言英语网页集合，得到其上下文集合$\{T_{i1}, T_{i2}\ldots\ldots T_{in}\}$，然后在每一个上下文环境 T_{ik} 中，利用命名实体识别工具来判断 rc_i 是否一个人名。如果 rc_i 在某个上下文环境 T_{ik} 被识别为人名，则赋予 rc_i 一定的分值。如果 rc_i 在任何上下文环境中都不构成人名，则被裁减掉。

点击率： 从搜索引擎上获取经过矫正的音译候选 rc_i 的点击率信息，并用它来评价 rc_i 的重要程度。与[Al-Onaizan et al., 2002]不一样，本文直接使用点击率作为特征，而[Al-Onaizan et al., 2002]是用它来过滤掉不合法的翻译结果。

复合人名之间的限制信息： 在对一个复合人名进行音译时，我们首先把它分成几个单独的人名，然后把每个人名的音译结果组合起来作为复合人名的音译结果。但是，在音译过程中，每个单独的人名之间可以互相提供一些限制信息。例如："希拉里·克林顿"是一个复合人名，"希拉里"可以被音译为"Hilary"或"Hilaly"，"克林顿"可以被音译为 "Clinton"或"Klinton"。但是，如果它们构成一个复合人名，则"Hilary·Clinton"将是最常出现的形式，从而被挑选出来。因此，复合查询的点击率信息也被抽取出来作为分类器特征。

人名周围的线索词： 人名周围的线索词也可以被加入到查询中，从而为过滤掉噪音提供一些有用的信息。例如："总统 (president)"可以作为"克林顿 (Clinton)"的线索词。为了找到这些线索词，首先在中文网页中搜索中文人名，得到的网页中的高频词将作为线索词，并利用双语词典把它们翻译为英文形式。这些线索词和矫正过的音译候选组合成新的查询，搜索英文网页，这样得到的点击率信息作为有一个特征。

AdaBoost 分类器函数如下：

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \qquad (7)$$

其中 α_t 是第 i 个弱分类器 $h_t(x)$ 的权重，我们基于该分类器的准确率确定 α_t 的大小。

5 实验分析

本章通过一系列实验分析出"矫正"过程和"重排序"过程在多大程度上提升了统计音译基准系统的性能，同时分析网页挖掘方法可以在多大程度上解决第一章提出的两个问题。

5.1 实验数据

统计音译模型的训练语料来源于 Chinese-English Name Entity Lists v 1.0 (LDC2005T34)，该语料库包括 565,935 音译对。我们从中过滤掉一些不适合于汉英反向音译研究的音译对，例如汉日音译对，得到 14,443 对汉语-欧美人名对作为训练集，其中随机挑选 1,344 对作为封闭测试集，在训练集之外再找 1,294 对作为开放测试集。Web 词汇表是从规模为 2GB 的网页资源中得到的。因为测试集中 7.42% 的人名不出现在 Web 词汇表中，我们利用 *Google* 获取了包含这部分人名的网页资源补充进来，最终

得到的 Web 词汇表包含 672,533 个词。

5.2 矫正过程 vs. 统计音译方法

将统计音译模型得到的结果作为基准，我们评测了加入矫正过程后音译的召回率。统计音译模型分以下四部分进行：（1）中文人名转化为拼音表示，英文人名分解为音节表示；（2）利用 *GIZA++*[1]工具进行拼音和音节对齐，得到对齐概率 $P(py|es)$；（3）高频的音节序列组合为音节短语，例如："be/r/g"→"berg"，"s/ky"→"sky"；(4)利用 *Camel*[2] 解码器为每个输入的中文人名生成 100 个最有的音译候选。

表 2. 统计音译模型 vs. 矫正方法

	统计音译结果		矫正结果	
	close	open	close	Open
Top1	33.64%	9.41%	27.15%	11.04%
Top5	40.37%	13.38%	42.83%	19.69%
Top10	47.79%	17.56%	56.98%	26.52%
Top20	61.88%	25.44%	71.05%	37.81%
Top50	66.49%	36.19%	82.16%	46.22%
Top100	**72.52%**	**41.73%**	**85.78%**	**59.28%**

表 2 给出了统计音译基准系统以及附加了矫正过程的性能比较，可以看出，在附加矫正过程以后，封闭测试的 top-100 召回率可以提高 13.26%，开放测试的 top-100 召回率可以提高 17.55%。这说明矫正过程可以有效地纠正统计音译模型中的错误。

为了评测矫正过程可以在多大程度上解决统计音译模型存在的两个问题：不发音音节的丢失问题以及小概率答案的丢失问题，我们定义了新统计量"矫正次数"。对于一个中文词，如果它的正确音译只有在经过矫正过程以后才出现在 top-100 音译候选中，则统计一次"矫正次数"。例如： 当 "A-ga -hi" 被矫正为"A-ga-s-si"时，对于 Problem II 的"矫正次数"为 1，因为"hi"→ "si" 是一个音节扩展；对于 Problem I 的"矫正次数"也为 1，因为不发音音节"s"被扩展出来。

表 3. 平均矫正次数

	Close test	Open test
Problem I	0.6931	0.7853
Problem II	0.9264	1.1672

统计音译模型的召回率在某种程度上依赖于英文人名的长度。对于长的人名可能包含更多的不发音音节或者容易混淆的音节，统计模型很难得到完全正确的音译结果。但是，矫正过程可以有效地解决这个问题。图 2 比较了统计模型和矫正过程的召回率随英文人名长度变化的情况，曲线显示矫正过程有效解决了召回率随人名长度增长而降低的情况。

[1] http://www.fjoch.com/GIZA++.html

[2] http://www.nlp.org.cn

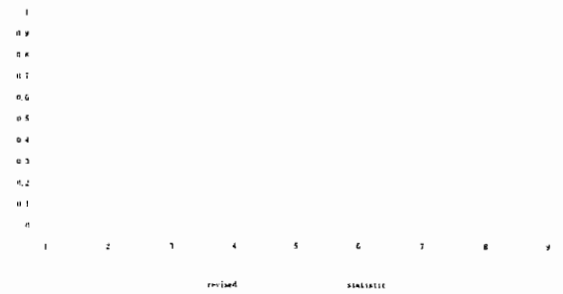

图 2. 召回率随人名长度变化情况的比较

5.3 矫正过程的参数设置

以下将分析查询扩展的不同参数设置对实验结果的影响。在基于发音的查询扩展中，对于每个汉语拼音，我们挑选 20 个音节建立扩展集。在公式（5）中设定 $\beta = 0.1$，则实验结果在表 4 中的“exp1”列给出。从这个结果中可以看出，当 $\alpha = 0.4$ 时我们得到最好的性能。这说明，当精确匹配的权重比模糊匹配的权重稍微大一些时，音译的召回率最好。如果精确匹配的权重太高，召回率将很低；而模糊匹配的权重太高，将引入噪音。

我们也评测的基于音节相似度的查询扩展方法。对于每一个音节，我们选择最多 15 个音节建立查询扩展集。设定 $\beta = 0.1$，则实验结果在表 4 中的“exp2”列给出。从实验结果我们可以看出，当 $\alpha = 0.5$ 时我们得到最好的性能。这说明，我们不能偏重任意一种匹配方法。与基于发音相似度的方法相比，基于音节相似度的方法的性能要差。这说明基于发音相似度的方法更适合于矫正统计音译候选中的错误。

表 4. 参数设置实验

	$\alpha = 0.2$		$\alpha = 0.3$		$\alpha = 0.4$		$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	exp1	exp2	exp1	exp2	exp1	exp2	exp1	exp2	exp1	exp2	exp1	exp2	exp1	exp2
Top1	13.46	13.32	13.79	13.61	**11.04**	12.70	11.65	**10.93**	10.83	11.25	9.62	10.63	8.73	10.18
Top5	21.58	19.59	23.27	20.17	**19.69**	18.28	21.07	**17.25**	22.05	16.84	17.90	16.26	17.38	15.34
Top10	27.39	22.71	28.41	24.73	**26.52**	22.93	26.83	**21.81**	27.26	20.39	24.38	21.20	25.42	18.20
Top20	35.23	34.88	35.94	29.49	**37.81**	31.57	38.59	**33.04**	36.52	31.72	35.25	**29.75**	34.65	27.62
Top50	43.91	40.63	43.75	40.85	**46.22**	41.46	48.72	**42.79**	45.48	40.49	41.57	39.94	42.81	38.07
Top100	53.76	48.47	54.38	52.04	**59.28**	53.15	57.36	**53.46**	55.19	51.83	55.63	49.52	53.41	47.15

5.4 矫正过程 vs. 重排序过程

在对统计音译结果进行矫正以后，我们将从单语言 Web 资源中挖掘相关信息对音译候选进行重排序。以下的实验将显示出重排序过程对于精确率的提升程度。

我们为 AdaBoost 分类器选择四种特征。为了判断一个音译候选在具体上下文环境中是否是人名，我们使用了软件工具 Lingpipe3。把查询送入 google，得到每个查询的点击率，并把 top-10 snippets 抽取出来作为上下文环境特征。矫正过程和重排序过程的 Top-N 精确率对比如下图所示。

表 5. 矫正过程 vs. 重排序过程

	矫正结果		重排序结果	
	close	open	close	open
Top1	27.15%	11.04%	**58.08%**	**38.63%**
Top5	42.83%	19.69%	**76.35%**	**52.19%**
Top10	56.98%	26.52%	83..92%	54.33%
Top20	71.05%	37.81%	83.92%	57.61%
Top50	82.16%	46.22%	83.92%	57.61%
Top100	**85.78%**	**59.28%**	85.78%	59.28%

从以上结果可以看出，在重排序过程以后，噪声词的排名下降了。矫正过程和重排序过程联合工作，则如果只返回 5 个结果时，召回率和精确率都得到提升。

我们利用平均排名(average rank, AR)和平均倒数排名(average rank and average reciprocal rank，ARR) [Voorhees and Tice, 2000]来评测性能的改进程度， ARR 的计算公式如下：

$$ARR = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{R(i)} \tag{8}$$

其中 $R(i)$ 是第 i 个测试词的正确翻译的排名，M 是测试集的规模。ARR 越大，性能越好。表 6 是测试结果。

从上表可以看出，矫正过程的 ARR 小于统计模型。因为矫正模型的目标是尽可能地提升召回率，因此可能会引入一些噪声词。这些噪声词将在重排序阶段被过滤掉，因此经过重排序过程后，我们得到了最高的 ARR 值。因此，我们可以得到以下结论：矫正过程提升了召回率，而重排序提升了精确率，这两个过程中很大程度上改善了统计音译模型的性能。

表 6. ARR 和 AR 评测

	统计模型		矫正过程		重排序过程	
	Close	open	close	open	close	open
AR	37.63	70.94	24.52	58.09	16.71	43.87
ARR	0.3815	0.1206	0.3783	0.1648	0.6519	0.4492

6 结论

本文提出了一种利用单语言 Web 资源挖掘技术辅助音译对方法。通过矫正过程，音译的封闭测试 top-100 召回率从 72.52%提升到 85.78%，开放测试 Top-100 召回率从

[3] http://www.alias-i.com/lingpipe/

41.73%提升到 59.28%。通过重排序过程，音译的封闭测试 top-5 精确率从 42.83%提升到 76.35%，开放测试 top-5 精确率从 19.69%提升到 52.19%。实验结果显示，这种方法适合于反向音译任务。

在后续的研究中，我们将改进矫正过程的相似度计算方法，并研究基于音译候选挖掘正确答案的更直接有效地算法。

参考文献：

YANG Fan, ZHAO Jun, ZOU Bo, LIU Kang. 2008. Chinese-English Backward Transliteration Assisted with Mining Monolingual Web Pages, In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-2008), Columbus, OH, June 15-20, 2008

Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In Proc.of ACL-02.

Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. Computational Linguistics 24(4).

Wei-Hao Lin and Hsin-His Chen. 2002 Backward Machine Transliteration by Learning Phonetic Similarity. In Proc. Of the 6th CoNLL

Donghui Feng, Yajuan Lv, and Ming Zhou. 2004. A New Approach for English-Chinese Named Entity Alignment. In Proc. of EMNLP-2004.

Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu, 2007. Named Entity Translation with Web Mining and Transliteration. In Proc. of IJCAI-2007.

Wei Gao. 2004. Phoneme-based Statistical Transliteration of Foreign Name for OOV Problem. A thesis of Master. The Chinese University of Hong Kong.

Ying Zhang, Fei Huang, Stephan Vogel. 2005. Mining translations of OOV terms from the web through cross-lingual query expansion. SIGIR 2005.

Pu-Jen Cheng, Wen-Hsiang Lu, Jer-Wen Teng, and Lee-Feng Chien. 2004 Creating Multilingual Translation Lexicons with Regional Variations Using Web Corpora. In Proc. of ACL-04

Masaaki Nagata, Teruka Saito, and Kenji Suzuki. 2001. Using the Web as a Bilingual Dictionary. In Proc. of ACL 2001 Workshop on Data-driven Methods in Machine Translation.

Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In Proc. of the ACL workshop on Multi-lingual Named Entity Recognition.

Jenq-Haur Wang, Jei-Wen Teng, Pu-Jen Cheng, Wen-Hsiang Lu, Lee-Feng Chien. 2004. Translating unknown cross-lingual queries in digital libraries using a web-based approach. In Proc. of JCDL 2004.

E.M.Voorhees and D.M.Tice. 2000. The trec-8 question answering track report. In Eighth Text Retrieval Conference (TREC-8).