ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.5, May 2009, pp.1241−1253
doi: 10.3724/SP.J.1001.2009.03409

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于同步树替换文法的统计机器翻译模型*

蒋宏飞[1+], 李 生[1], 付国宏[2], 赵铁军[1], 张 民[3]

[1](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

[2](黑龙江大学 计算机科学技术学院,黑龙江 哈尔滨 150080)

[3](Institute for Infocomm Research, Singapore)

Statistical Machine Translation Model Based on a Synchronous Tree-Substitution Grammar

JIANG Hong-Fei[1+], LI Sheng[1], FU Guo-Hong[2], ZHAO Tie-Jun[1], ZHANG Min[3]

[1](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](School of Computer Science and Technology, Heilongjiang University, Harbin 150080, China)

[3](Institute for Infocomm Research, Singapore)

+ Corresponding author: E-mail: hf.jiang@gmail.com

Jiang HF, Li S, Fu GH, Zhao TJ, Zhang M. Statistical machine translation model based on a synchronous tree-substitution grammar. *Journal of Software*, 2009,20(5):1241−1253. http://www.jos.org.cn/1000-9825/3409.htm

Abstract: A translation model based on synchronous tree-substitution-grammar is presented in this paper. It can elegantly model the global reordering and discontinuous phrases. Furthermore, it can learn non-isomorphic tree-to-tree mappings. Experimental results on two different data sets show that the proposed model significantly outperforms the phrase-based model and the model based on synchronous context-free grammar.

Key words: machine translation; synchronous tree-substitution-grammar; tree-to-tree model; global reordering; non-isomorphic tree-to-tree mapping

摘 要: 提出一种基于同步树替换文法的机器翻译模型.相对于基于短语的模型,此模型可以对远距离结构性调序和非连续短语翻译进行建模;相对于基于同步上下文无关文法模型,此模型可以对任何层次上的树节点调序进行建模.因此,该模型可以为处理语言结构间的异构对应问题提供有效的解决途径. 在两组风格差异较大的数据集上进行的实验均验证了基于同步树替换文法的模型相对于基于短语模型和基于同步上下文无关文法模型的稳定优势.

关键词: 机器翻译;同步树替换文法;树到树模型;全局调序;异构对应

中图法分类号: H085 **文献标识码**: A

机器翻译就是应用计算机实现从一种自然语言文本到另一种自然语言文本的翻译[1].随着计算机计算能力的不断提高以及硬件存储容量的不断增大,基于语料库的机器翻译技术受到越来越多的研究者的重视.其中

* Supported by the National Natural Science Foundation of China under Grant No.60736014 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2006AA010108 (国家高技术研究发展计划(863))

Received 2008-04-08; Accepted 2008-07-02

基于统计的机器翻译(statistical machine translation,简称 SMT)技术已经成为目前机器翻译研究领域的主流.

SMT 的第一个具有代表性的模型是由 IBM 的 Brown 等人在 1993 年提出来的[2].文献[2]将机器翻译看成是一个信息传输的过程,并用一种噪声信道模型对机器翻译进行解释.在此基础上,将翻译过程参数化为翻译模型和语言模型两个模型.但是,IBM 提出的模型仅仅考虑了词汇到词汇的对齐关系和一些简单的调序模型,获得的译文文法结构和准确率较差.随后出现的基于短语的机器翻译系统[3−6]把短语作为基本翻译单元,极大地提高了译文的准确度.但是,与 IBM 的基于词的模型相同,两者都是基于有限状态转录机(finite state transducer,简称 FST)的线性模型[7],不能有效地对翻译过程中的结构性词序调整进行建模.比如英语的 SVO(subject-verb-object)子句在翻译成日语时,经常变为 SOV 的句型.基于 FST 的模型难以对这种翻译现象进行有效的建模.另外,这些模型也无法对翻译过程中的非连续短语对应现象进行处理.比如,中文的“打了[某人]一顿”翻译为英语就成了“hit[sb.]”,其中“打了…一顿”与英语中的“hit”相对应.再比如,法语中的“nes…pas”在英语中的对应词汇为“not”.

本质上讲,这些基于短语的模型所存在的问题都与语言本身的结构性有关.基于此,机器翻译领域的研究者相继开展了基于句法翻译方法的研究,并已提出了一些模型.大体上,这些句法模型可以分为两类:基于语言学句法结构的模型[8−10]和非语言学(纯形式化)句法结构的模型[11,12].第 1 类模型是基于语言学的句法结构(短语句法树或者是依存句法树)来建立翻译模型.其中绝大多数模型仅考虑一端句法树结构.第 2 类模型则基于纯形式文法,文法规则中的非终结符并不是语言学意义上的句法标记,而仅仅是一些抽象的变量符号.这两类模型各有优、缺点.前者更加接近人类的思维习惯,使得人类易于对翻译模型进行分析和改进,但是同时会遇到对译句法结构异构、句法分析引入的错误等问题.后者不受限于具体语言的句法规范,但同时也不具有语言句法结构所包含的丰富特征.从形式上讲,目前已有的大部分句法模型(包括基于语言学的和纯形式化的)都可被看成是基于同步上下文无关文法(synchronous context-free grammar,简称 SCFG)的.此类模型的优点是易于实现,解码过程复杂度也相对较低.如文献[11]中所实现的 BTG(bracketing transduction grammar)文法,在实际实现时可以通过限制文法规则中的非终结节点数最多为 2,从而使得解码可以通过动态规划过程高效地实现.然而,基于 SCFG 文法的模型只允许处于同一层次中的兄弟节点之间进行调序,因而此文法在形式上就要求两种互译语言之间存在结构性的同构.一般情况下,语言之间,特别是不同语系的语言之间在结构上存在大量的非同构对应现象.因此,基于 SCFG 文法的模型无法模拟复杂的结构对应问题,故而远远不足以对语言翻译现象进行建模[13].

针对上述模型所存在的问题,本文提出一种基于同步树替换文法(synchronous tree-substitution grammar,简称 STSG)的模型.相对于基于短语的模型,此模型可以对远距离结构性调序和非连续短语翻译进行建模;相对于基于 SCFG 模型,此模型可以对任何层次上的树节点调序进行建模.因此,本模型可以为处理语言结构之间的异构对应问题提供有效的解决途径.本文首先给出一个 STSG 的形式化定义;然后给出基于 STSG 的机器翻译模型的数学模型;接着,本文详细给出文法规则的自动抽取算法和解码算法,并对其中的关键问题进行深入分析;最后,在实验部分,本文在两个数据集上对基于短语的模型、基于 SCFG 的模型和基于 STSG 的模型做了较为全面的比较分析.为了形象化地对 3 种模型的能力进行比较,本文还给出一个在真实测试集中的例子来进行实例分析比较.

1 同步树替换文法

本文提出的模型基于同步树替换文法.本文中的 STSG 是对文献[14]给出的 STSG 在机器翻译领域的一个应用实例.Shieber 在文献[14]中对同步树替换文法给出如下定义:一个 STSG 就是一个五元组 $G=\langle \Sigma_{in},\Sigma_{out},P,S_{in},S_{out}\rangle$,其中:

- Σ_{in} 和 Σ_{out} 分别是输入、输出的有序字符集.
- $S_{in}\in\Sigma_{in},S_{out}\in\Sigma_{out}$ 分别是输入、输出的起始符号.
- P 是一个产生式规则集合.元树(elementary tree)对集.其中每个元树对中的两个元树存在一定的连接关系.

本文将 STSG 引入统计机器翻译,并给出一个更为具体的定义.在我们所要研究的统计机器翻译课题范围内,一个 STSG 是一个七元组:$G=\langle \Sigma_s,\Sigma_t,N_s,N_t,P,S_s,S_t\rangle$,其中:

- Σ_s 和 Σ_t 分别是源语言端和目的语言端的终结符(词语,单词)字符集.
- N_s 和 N_t 分别是源语言端和目的语言端的非终结符(词性,句法标记等)字符集.
- $S_s\in\Sigma_s$ 和 $S_t\in\Sigma_t$ 是源语言端和目的语言端的起始符号(相对应于源语言、目的语言的句法树根节点).
- P 是一个产生式规则集合.

这里将源语言端和目的语言端的句法标记集分别加以考虑,其原因是不同语言的句法标记体系不尽相同.在给出 STSG 文法中的产生式规则的详细描述之前,需要先给出几个定义.在一般意义上,树是一个有根的、连通的无环图.本研究中的树特指句法树.为了下文叙述方便,这里给出一个树的组成式定义.

定义 1(树 tree). 假设 Σ 是一个终结符表(对应语言中的词表),N 是一个非终结符表(对应句法标记集),那么一个三元组 $T=\langle V,E,V_1\rangle$是一个树,如果:

- V 是节点集,E 是边集;
- $V_1\subset V$ 是叶子节点集,并且有 $V_1\subset\Sigma$,$(V\backslash V_1)\subset N$;
- T 是一个无向无环连通图.

基于定义 1,我们可以给出元树的定义:

定义 2. 假设 T 是一棵树,那么元组 $\xi=\langle V,E,V_1\rangle$是 T 的一棵元树,如果:

- V 是节点集,E 是边集, $V_1\subset V$ 是叶子节点集;
- $V_1\subset(\Sigma\cup N)$, $V\subset V(T)$,$E\subset E(T)$;
- $\forall$node $w\in(V\backslash V_1)$,$\forall$node $v\in\{n|$ node n is a direct son of w in tree $T\}\Rightarrow v\in V$.

图 1 给出了一个带有词对齐关系的句法树对的例子.而 VP(VBA NP(DT(the) NN(pen)) PP)和 PP(TO(to) PRP(me))是两棵来自图 1 英语句法树的元树.其中,前者是一棵叶子节点有非终结符的元树,后者是一棵一般的子树.从此也可以看出,子树一定是一棵元树,但反之则不然.

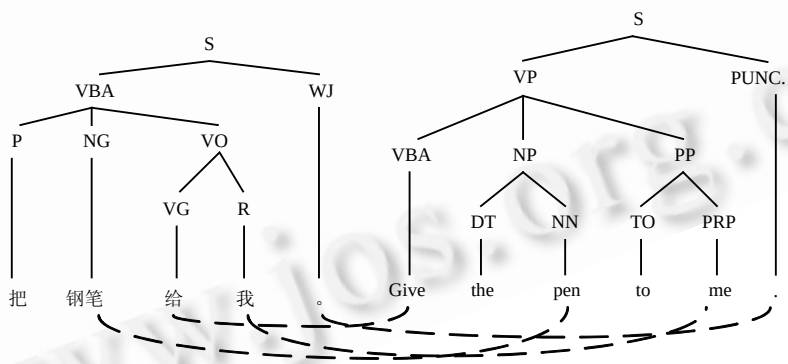


Fig.1 An example of syntax pair with word alignment

图 1 带词对齐关系的句法树对例子

接下来对 STSG 中产生式规则给出解释.其一般形式为

$$\langle\alpha,\beta,A\rangle,$$

其中,α 为源语言端元树,β 为目的语言端元树,A 为两者节点间对应关系.另外,实际中,STSG 文法的每个产生式规则都带有一组特征值,以便对规则质量进行衡量.

表 1 中给出了几个可以从图 1 所示的句法树对抽取出来的规则示例.本文中将元树中的非终结符叶子节点称为抽象节点.其中,R1~R5 是子树对,没有抽象节点,不具备泛化能力;R6~R9 都有抽象节点,具有泛化能力,两端抽象节点是一一对应的.

Table 1 Examples of the rules extracted from the syntax pair in Fig.1

表 1 从图 1 所示句法树对中抽取出来的部分规则

R1: NG(钢笔) ||| NN(pen)
R2: VG(给) ||| VBP(Give)
R3: R(我) ||| PRP(me)
R4: WJ(。) ||| PUNC.(.)
R5: VBA(P(把) NG(钢笔) VO(VG(给) R(我))) ||| VP(VBA(Give) NP(DT(the) NN(pen)) PP(TO(to) PRP(me)))
R6: VBA(P(把),NG[0],VO(VG(给),R(我))) ||| P(VBP(give),NP(DT(the),NN[0]),PP(TO(to),PRP(me)))
R7: VBA(P(把),NG(钢笔),VO(VG[0],R(我)))|||P(VBP[0],NP(DT(the),NN(pen)),PP(TO(to),PRP(me)))
R8: VBA(P(把),NG[0],VO(VG[1],R(我))) ||| VP(VBP[1],NP(DT(the),NN[0]),PP(TO(to),PRP(me)))
R9: S(VBA[0],WJ[1]) ||| S(VP[0],PUNC.[1])

2 基于STSG的统计机器翻译系统数学模型

机器翻译系统的基本任务是将一个输入的源语言句子 f 翻译成适当的目标语言句子 e.这样,机器翻译模型的基本任务就是对概率 $\Pr(e|f)$进行建模.本模型是通过树到树的转化来对翻译过程进行建模.因此,首先需要引入隐含变量 $T(f)$和 $T(e)$来分别表示源语言句子和目的语言句子的句法树.于是,$\Pr(e|f)$可以得到如下推导:

$$\Pr(e|f)=\sum_{\langle T(f),T(e)\rangle}\Pr(e,T(e),T(f)|f) \tag{1}$$

接着,我们对公式(1)右边和式中的每项进行进一步推导:

$$\Pr(e,T(e),T(f)|f)=\Pr(T(f)|f)\times\Pr(T(e)|T(f),f)\times\Pr(e|T(e),T(f),f) \tag{2}$$

公式(2)将公式(1)中等号右边和式中的每项分解为 3 个因式,每个因式对应一个子模型.其中,$\Pr(T(f)|f)$是句法分析模型.因为在我们目前的研究实现中,对每个输入句子 f,仅仅考虑句法分析器的最优输出 $T(f)$.所以,句法分析的模型可以略去.第 3 个子模型 $\Pr(e|T(e),T(f),f)$对应从生成的目的语言句法树中得到目的语言句子的过程.也即对句法树叶子节点依次进行扫描输出为目的语言句子.对于一个既定的句法树,有且仅有 1 个句子与之对应.所以,这个子模型也同样可以不予考虑.本研究最关键的是第 2 个子模型 $\Pr(T(e)|T(f),f)$,它对应源语言句法树到目的语言句法树的转换过程.接下来重点讨论如何基于 STSG 文法对这个子模型进行建模.

本文将树到树的转换模型建立在 STSG 文法上,因此每个树到树的转换过程就等同于 STSG 文法的一个推导过程.图 2 演示了一个用表 1 给出的 STSG 规则对应于图 1 所示的句法树对的一个推导过程.假设 D 是文法 G 的一个推导,$f(D)$和 $e(D)$分别是 D 所生成的源语言端和目的语言端的句子.我们可以将 D 表示为一个三元组 $\langle r,i,j\rangle$的集合.每个$\langle r,i,j\rangle$表示一步推导:用规则 r 重写一个覆盖 $f(D)_j^i$ 的非终结符.$\Pr(T(e)|T(f),f)$值由对应的推导过程 D 的权重 $w(D)$来模拟.$w(D)$用 D 中包含的规则的权重的乘积来计算:

$$w(D)=\prod_{\langle r,i,j\rangle\in D}w(r).$$

每个规则 r 的权重用所对应的特征值的乘积来计算:

$$w(r)=\prod_i\phi_i(r)^{\lambda_i},$$

其中,ϕ_i 是特征函数,λ_i 是对应的特征权重.最终,本文将模型建立在一个对数线性构架下,如公式(3)所示.

$$\begin{aligned}\Pr(e,T(e),T(f)|f)&=\Pr(T(e)|T(f),f)\\&=\frac{\exp[\sum_{m=1}^{M}\lambda_m h_m(e,T(e),T(f),f)]}{\sum_{e'}\exp[\sum_{m=1}^{M}\lambda_m h_m(e',T(e'),T(f),f)]}\end{aligned} \tag{3}$$

〈S,S〉

$\overset{R9}{\Rightarrow}$ 〈**S(VBA[1] WJ[2]),S(VP[1] PUNC.[2])** 〉

$\overset{R8}{\Rightarrow}$ 〈S(**VBA(P(把) NG[3] VO(VG[4] R(我)))** WJ[2]), S(**VP(VBP[4] NP(DT(the) NN[3]))** PUNC.[2])〉

$\overset{R4}{\Rightarrow}$ 〈S(VBA(P(把) NG[3] VO(VG[4] R(我))) **WJ(。)**), S(VP(VBP[4] NP(DT(the) NN[3])) **PUNC.(.)**)〉

$\overset{R1}{\Rightarrow}$ 〈S(VBA(P(把) **NG(钢笔)** VO(VG[4] R(我))) WJ(。)), S(VP(VBP[4] NP(DT(the) **NN(pen)**)) PUNC.(.))〉

$\overset{R2}{\Rightarrow}$ 〈S(VBA(P(把) NG(钢笔) VO(**VG(给)** R(我))) WJ(。)), S(VP(**VBP(Give)** NP(DT(the) NN(pen))) PUNC.(.))〉

Fig.2 A derivation of the syntax pair in Fig.1 using the rules listed in Table 1

图 2 用表 1 中列出的部分规则对图 1 所示句法树对的一个推导

在本文实现的系统中,采用了以下特征:

(1) 双向的元树映射概率:

a) $\phi(e \mid f) = \log \prod_{k=1}^{K} \frac{N(\xi_e^k, \xi_f^k)}{N(\xi_f^k)}$

b) $\phi(f \mid e) = \log \prod_{k=1}^{K} \frac{N(\xi_e^k, \xi_f^k)}{N(\xi_e^k)}$

(2) 双向的词汇翻译概率:$lex(f \mid e)$和 $lex(e \mid f)$

(3) 语言模型得分:lm

(4) 翻译推导过程中所用到的规则个数:K

(5) 生成译文的单词数:l

3 规则自动抽取

STSG 文法的最重要组成部分就是产生式规则集合.本节给出一种基于双语并行语料的 STSG 文法规则自动抽取方法.规则自动抽取模块的输入是一个含有词对齐信息的双语并行句法树对集合.每个句对是一个三元组$\langle T(f), T(e), A\rangle$,其中 $T(f)$和 $T(e)$分别是源语言端和目的语言端句子的句法树,A 是两端的词对齐关系.

图 1 给出了一个训练句对的例子.本文给出的规则抽取算法不考虑训练句对之间的相互影响,即规则的抽取是对每个句对独立进行的.那么给定一个如图 1 所示的训练句对,如何抽取出有用的产生式规则呢?更具体地,如何才能衡量两棵分别来自源端和目的端的元树是否可以构成一个产生式规则呢?给定一个训练句对三元组$\langle T(f), T(e), A\rangle$,并假设 $T(f_{j_1}^{j_2}), T(e_{i_1}^{i_2})$ 分别是 $T(f)$和 $T(e)$的任意元树,如果没有任何约束,$T(f_{j_1}^{j_2}), T(e_{i_1}^{i_2})$ 就可以组成一个规则.但是,显然这样得到的规则集总体质量不能保证,而且数量上会非常巨大.在本研究中,利用了两方面约束对其进行限制.其中一个是词对齐约束.形式上,词对齐约束可如下表示:

- C1: $\exists (i,j) \in A, i_1 \le i \le i_2, j_1 \le j \le j_2$ 并且 $\forall (i,j) \in A : i_1 \le i \le i_2 \leftrightarrow j_1 \le j \le j_2$

另外一个是句法限制,实际上已经被产生式规则的定义所限定(这里,为了便于下文给出抽取基本规则,将树进一步限制为子树),形式上有:

- C2: $T(f_{j_1}^{j_2})$ 是 $T(f)$的一棵子树,$T(e_{i_1}^{i_2})$ 是 $T(e)$的一棵子树.

按照是否具有泛化能力,我们将产生式规则分为两类:

1) 基本规则

元树的叶子节点都是终结符(单词)的规则不具有泛化能力.如表 1 中的 R1~R5.

2) 泛化规则

元树存在非终结符的叶子节点的规则具有泛化能力.如表 1 中的 R6~R9.

在抽取过程中,我们先对基本规则进行抽取,然后基于基本规则再进一步生成泛化规则.下面分别给出抽取基本规则和泛化规则的算法.

算法 1. 基本规则抽取算法.

输入:句法树对$\langle T(f),T(e),A\rangle$,$A$ 为对齐关系.

1. 建立一个空的基本规则集合 S_B.
2. For $\forall n(n\in T(f))$
3. $t(n)$是以 n 为根节点的 $T(f)$的子树;
4. For $\forall m(m\in T(e))$
5. $t(m)$是以 m 为根节点的 $T(e)$的子树;
6. $A(t(n),t(m))$是 A 中与 $t(n)$和 $t(m)$相关的词对齐关系
7. If $\langle t(n),t(m),A(t(n),t(m))\rangle$满足词对齐限制 C1 和句法限制 C2
8. Then 将$\langle t(n),t(m),A(t(n),t(m))\rangle$加入规则集合 S_B

输出:基本规则集合 S_B.

算法 2. 泛化规则抽取算法.

输入:句法树对$\langle T(f),T(e),A\rangle$,$A$ 为对齐关系;基本规则集合 S_B.

1. 建立一个空的规则队列 Q
2. $Q\leftarrow S_B$ //用基本规则集合初始化队列 Q
3. If 扫描到 Q 的队尾
4. Then 结束
5. $p=\langle\xi_f,\xi_e,a\rangle\leftarrow$Next($Q$) //取 Q 的下一个元素并赋予 p
6. For $\forall n(n\in\xi_f)$
7. $C:\{m\mid m\in\xi_e\wedge\exists\langle t(m),t(n),a'\rangle\rangle\in S_B\}$
8. For $\forall k(k\in C)$
9. 将 p 中的$\langle n,k\rangle$部分进行抽象规约,形成新的泛化规则 p';
10. 将 p'加入 Q 的队尾;
11. Goto 2.

输出:全部规则集合 Q.

抽取基本规则的算法(算法 1)是一个很直观的过程.算法扫描任意一个元树对,如果它们满足约束条件 C1 和 C2,就可以用它们构造一个基本规则.而抽取泛化规则的算法(算法 2)则相对复杂.首先,建立一个规则队列,并将前面抽取得到的基本规则装入队列.接着,每次弹出队首规则$p=\langle\xi_f,\xi_e,a\rangle$,并判断是否在基本规则集 S_B 中存在 ξ_f 的子元树 $t(n)$和 ξ_e 子元树 $t(m)$所组成的规则.如果存在,就可以把 $t(m)$,$t(n)$分别用它们的根节点 m,n 替换,从而形成两个抽象节点.这样形成的新规则就是一个新的泛化规则.比如,在图 1 给出的句对中,假设已经抽取出如下几条基本规则:如表 1 中的 R1,R2,R5.在 R5 的基础上,把 R1 部分进行抽象规约,即可得到一个泛化规则 R6.在 R6 的基础上再进一步将 R2 对应部分进行抽象规约,即可得到另一个泛化规则 R7.

如果不采用适当控制措施,则以上算法会产生数量巨大的规则集.这样,训练过程和解码过程将会变得非常低效.为了对整体的模型复杂度(规则抽取以及解码)进行控制,具体实现中,我们采用以下控制参数对规则数量进行控制:

1) c,任一规则中抽象节点个数的上限;

2) h,任一规则中元树高度的上限;

3) ω,从任一树节点对抽取出的抽象规则个数上限.

4 参数估计

在本文第 3 节给出的对数线性模型中,部分特征值以及模型中各个特征的权重值需要在解码前进行估计.需要估计的参数可以分为 3 种(假设每个规则的形式为$\langle\alpha,\beta,A\rangle$):1) 两端元树的互译概率 $P(\alpha|\beta)$,$P(\beta|\alpha)$;2) 词汇化

翻译分数 $lex(\alpha|\beta)$,$lex(\beta|\alpha)$;3) 对数线性模型中的特征权重参数$\{\lambda_i\}$.

4.1 元树翻译概率的估计

为了估计 $P(\alpha|\beta)$和 $P(\beta|\alpha)$,需要对抽取出来的规则进行统计.一般情况下,给定一个训练句子对 P,存在多个推导过程可以用从 P 中抽取出来的规则推导出 P.实际上,在参数估计的过程中,我们并不能确切地知道每个推导的出现次数.同样,每个规则的出现次数也不能准确得到.

这样,一个直观的近似做法就是直接在抽取出的规则集中统计 α,β 的出现次数 $count(\alpha)$,$count(\beta)$以及$count(\langle\alpha,\beta\rangle)$然后用相对频度来估计:$P(\alpha|\beta)= count(\langle\alpha,\beta\rangle)/count(\beta)$,$P(\beta|\alpha)=count(\langle\alpha,\beta\rangle)/count(\alpha)$.

文献[12]使用了一种较为细致的考虑方法.在这种方法中,先对规则集中的每个基本规则赋予一个特定的计数.然后,每个基本规则所抽取出来的抽象规则平分这个计数.基于这种计数分配策略,再使用相对频度的方法得到 $P(\alpha|\beta)$和 $P(\beta|\alpha)$.在本文的研究中采用了与文献[12]一致的方法.

4.2 词汇化翻译分数的估计

给定规则$\langle\alpha,\beta,A\rangle$,本文使用类似于文献[3]所用的方法来对词汇化翻译分数 $lex(\alpha|\beta)$和 $lex(\beta|\alpha)$进行计算.假设 α 中包含的源端单词为$\{f_1\ldots f_n\}$,则 $lex(\alpha|\beta)$的计算如公式(4)所示,其中,$w(f|e)$表示词汇翻译概率.词汇翻译概率可以在获取词对齐信息时用 GIZA++工具自动生成.$lex(\beta|\alpha)$的计算方法类似.

$$lex(\alpha\mid\beta)=\prod_{i=1}^{n}\frac{1}{|\{j\mid(i,j)\in A\}|}\sum_{\forall(i,j)\in A}w(f_i\mid e_j) \tag{4}$$

4.3 特征权重的估计

公式(3)中所包含的特征权重$\{\lambda_i\}$由最小错误率训练过程决定[15].最小错误率训练以翻译性能得分为评判准则.目标是通过迭代调整权重参数值,使得翻译性能得分在开发集上最优化.

5 解 码

因为在统计机器翻译奠基性的工作[2]中,利用信源信道模型对翻译过程进行建模,所以,在随后统计机器翻译的研究中都约定俗成地将翻译过程称为解码.一个机器翻译系统的任务就是把输入的源语言句子 f 翻译成为目的语言句子 e.换言之,一个机器翻译过程就是在所有的目的语言候选$\{e\}$中找出最优的译文 $\hat{e}$.公式(5)给出了本模型中的解码公式:

$$\hat{e}=\arg\max_{e,T(e)}\left\{\sum_{m=1}^{M}\lambda_m h_m(e,T(e),T(f),f)\right\} \tag{5}$$

对于输入的源语言句子 f,首先用句法分析器得到对应的句法树 $T(f)$,解码过程的输入就是 $T(f)$.翻译过程可以分为如图 3 所示的 5 个步骤.第 1 步,加载翻译模型(文法规则集)和语言模型;第 2 步,读入源语言句法树 $T(f)$,并对每个树节点进行后续编号;第 3 步,获取可用的规则集;第 4 步,进行从底向上的树到树转化过程,这个过程称为栈搜索;第 5 步,将最优译文输出.

其中,解码过程的核心是第 4 步,下文简称为栈搜索.在这一步中,本文采用了一种从底向上、逐步进行节点扩展的过程.在这个过程中,对每个后序编号为 i 的节点,均存在一个翻译选项栈(translation option stack) $TransOption[i]$和一个假设栈(hypothesis stack)$Hypo[i]$与之对应.$TransOption[i]$中存放的是翻译节点 i 对应的子树时所有可用的规则,$Hypo[i]$中存放的是以节点 i 为根的子树的翻译候选.算法 3 给出了此步骤的较为细化的过程.为了控制解码过程的复杂度,设置了一个 $BeamThreshold$ 阈值来对 $Hypo[i]$中包含的翻译候选数量进行控制.

首先,用 $TransOption[0]$对 $Hypo[0]$进行初始化.因为后序编号为 0 的树节点一定是叶子节点,故 $TransOption[0]$中存放的一定是基本规则,无须进一步翻译,可直接作为译文候选.接着,算法按后续编号 i 从小到大的顺序依次扫描 $TransOption[i]$.对其中每个元素 $TransOption[i,j]$,如果是基本规则,则直接作为翻译候选;若是泛化规则,则利用 $Hypo[0]$~$Hypo[i-1]$中所有已获得的译文候选对 $TransOption[i,j]$中所含抽象节点进行替换翻译.当全部抽象节点被翻译完以后,将获得的新译文候选置入 $Hypo[i]$中.最后,对任意一个 $Hypo[i]$,需要按照

BeamThreshold 进行剪枝.

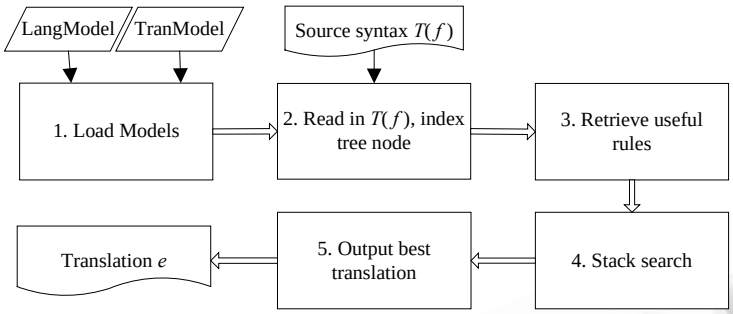


Fig.3 Decoding flowchart of the STSG based SMT model

图 3 基于 STSG 文法 SMT 模型的解码过程流程图

6 对比实验分析

这一节给出本文提出的基于 STSG 的翻译模型与基于短语的模型以及基于 SCFG 的模型的实验对比分析.对于基于短语的系统,我们采用由 Koehn 等人开发的 Pharaoh 系统[4]和 Moses 系统[16].前者是影响最大的、可公开免费使用的基于短语的统计机器翻译系统,后者可看成是前者的升级版本,可利用更加丰富的特征,本文中主要利用其可灵活设置语言模型元数的功能,其余设置和 Pharaoh 完全一致.对于另外两个基于句法系统,我们实现了一个基于 STSG 翻译模型的原型系统,并通过设置元树高度上限 h 为 2 来模拟 SCFG 系统(参见第 3 节结尾部分的限制参数说明).以下实验在两个差异较大的数据集上进行.因为本文所提出的模型是一种树到树的翻译模型,即训练双语语料的源语言端和目的语言端都需要标注有句法树,为了尽量避免自动句法分析结果的错误对机器翻译性能带来的影响,故第 1 组实验采用的数据集有人工校正过的句法树信息.由于人工校正句法语料的稀缺性,故此数据集仅含有 10 000 句对.另外,此数据集语料的来源大都为对话题材的文体,故可看成是口语语料.第 2 组实验中采用了目前机器翻译领域常用的 FBIS 数据集.它的句法信息是由目前流行的自动句法分析器自动获取的,此数据集具有 20 万句对的数据规模,而且此数据集来源为政府官方的议会会议记录,故句子较长.以下分别详细介绍两组实验的情况.

6.1 实验1:基于人工校正句法语料的实验

6.1.1 实验设置

本组实验所用语料是哈尔滨工业大学机器智能与翻译实验室经过人工校正的中英双语对齐树库的一部分.训练集共 9 000 句(75 026 个中文词汇,78 223 个英文词汇).开发集共 528 句(4 432 个中文词汇,4 630 个英文词汇).测试集共 1 000 句(8 334 个中文词汇,8 614 个英文词汇).本数据集中对每个中文句子,仅有 1 个英文参考译文与之对应.

算法 3. 栈搜索(stack search).

输入:可用规则存放栈 *TransOption*,其中,*TransOption*[*i*]中存放那些在翻译以树节点 i 为根节点的子树 $t(i)$ 时可用的规则集合;输入句子句法树节点个数 N.

输出:假设(hypothesis)存放栈 *Hypo*,其中,*Hypo*[*i*]中存放与 $t(i)$对应的译文候选集.

1. *Hypo*[0]←*TransOption*[0]
2. For *i*←1...*N*−1
3. For *j*←0...sizeof(*TransOption*[*i*])−1
4. If *TransOption*[*i*,*j*]是一个基本规则
5. Then 将 *TransOption*[*i*,*j*]加入 *Hypo*[*i*]中;

6. Else
7. ζ^0={*TransOption*[*i*,*j*]}
8. For *k*=0...*TransOption*[*i*,*j*]中抽象节点数 *M*
// node(*k*)表示第 *k* 个抽象节点对应的树节点编号
9. ζ^{k+1}=用 *Hypo*[node(*k*)]来替换 ζ^k 中的第 *k* 个抽象节点获得的中间结果;
10. 将 ζ^M 加入 *Hypo*[*i*];
11. If sizeof(*Hypo*[*i*])>*BeamThreshold*
12. Then 对 *Hypo*[*i*]进行剪枝,以满足 *BeamThreshold* 限制

本实验中采用了一个基于词典和统计相结合的词对齐工具[17]来获取训练句对的词对齐信息.基于训练集 9 000 句的英文句子,我们用 SRILM 工具[18]训练了一个 3 元的语言模型,采用的平滑算法为修正的 Kneser-Ney 策略[19].实验所用各模型的特征权重均采用文献[15]中提出的最小错误率训练方法进行估计.此方法在开发集上针对评测指标 BLEU(bilingual evaluation understudy)[20]进行最优化迭代来对参数权重进行调整.

本文采用 BLEU 作为译文质量的评价指标,并用 NIST(National Institute of Standards and Technology)官方网站发布的 mteval-v11.pl 来进行计算.实验结果的 BLEU 分数采用自举重抽样方法(bootstrapping resampling)[21]进行显著性测试,测试工具为 Zhang 等人的实现[22].如果不作特殊说明,则以下所报结果均具有 95%的置信度.

6.1.2 系统设置

本实验采用的作为对比的基准系统是 Pharaoh[4],它是一个广泛流行的基于短语的机器翻译系统.在本实验中,Pharaoh 采用了 8 个默认的特征:

1) 词序扭曲模型:*d*
2) 语言模型:*lm*
3) 短语翻译概率:$\Phi(e|f)$和 $\Phi(f|e)$
4) 词汇化权重:$lex(e|f)$和 $lex(f|e)$
5) 短语个数惩罚:*pp*
6) 译文词个数惩罚:*wp*

对 STSG 系统,规则中抽象节点个数的上限 *c* 设置为 5,规则中元树高度的上限 *h* 设置为 5,每个树节点对相关的抽象规则个数上限 ω 设置为 50.解码时的 *BeamThreshold* 设置为 100.SCFG 系统除了规则中元树高度的上限 *h* 设置为 2 以外,其余设置与 STSG 一致.

6.1.3 实验结果及对比分析

表 2 给出了 3 个系统在开发集上用最小错误率训练估计出的特征权重值.SCFG/STSG 系统均没有词序扭曲模型,其他特征均可类比于 Pharaoh 的特征.比如,SCFG/STSG 系统的元树映射概率类似于 Pharaoh 中的短语翻译概率;SCFG/STSG 系统的词汇互译得分特征类似于 Pharaoh 的词汇化权重;SCFG/STSG 系统的规则数类似于 Pharaoh 的短语个数惩罚 *pp*;译文词个数惩罚与语言模型得分特征 3 个系统一致.比较表 2 中的各个系统权重可以看出,STSG 系统倾向于选取利用较少规则的推导(*pp*>0),这样的推导倾向于使用较大的元数对,翻译也将更加准确.另外,SCFG/STSG 系统的词惩罚均为正值(*wp*>0),这表明含有较多词汇的规则更有利于准确的翻译.另外，3 个系统的语言模型权重均比较大,表明语言模型是一个很重要的特征.

Table 2 Feature weights obtained by MER training on the development set

表 2 在开发集上通过最小错误率训练获得的特征权重

Systems	Feature weights							
	D	$\Phi(e\|f)$	$lex(e\|f)$	*pp*	*wp*	*lm*	$\phi(f\|e)$	$lex(f\|e)$
Pharaoh	0.047	0.232	–0.025	0.126	–0.0995	0.167	0.130	0.172
SCFG	–	0.191	–0.03	0.054	0.318	0.180	0.20	0.012
STSG	–	0.209	–0.045	–0.207	0.152	0.227	0.148	0.010

表 3 给出了各个系统在训练集上抽取出来的规则(Pharaoh 的短语翻译对也可看成是翻译规则)个数以及在

测试集上进行翻译时用到的规则个数.Pharaoh 可以将所有满足词对齐约束的短语翻译对抽取出来.SCFG 模型则只能抽取出满足句法限制且元数高度为 2 的规则.虽然 SCFG 的规则中含有泛化规则,但在整体数目上仍然远低于 Pharaoh 的规则数.STSG 模型抽取规则数则大大超过了 Pharaoh 的规则数.这是因为 STSG 系统可以抽取高度小于等于 5 的元数对作为规则,而且大量的泛化规则也会随之产生.

Table 3 Numbers of extracted rules and of used rules in testing

表 3 抽取的规则数目和测试时所用规则数目

Systems	Numbers of extracted rules	Numbers of used rules in testing
Pharaoh	499 423	64 491
SCFG	70 000	24 302
STSG	2 629 146	98 422

表 4 给出了 3 个系统的性能比较.从中可以看出:

1) STSG 系统显著地超过了 Pharaoh.在 BLEU-4 分值上,STSG 系统比 Pharaoh 有 0.018 6(0.1394−0.1208)的绝对提升,也即相对性能提高为 15.3%(0.0186/0.1208).这些结果表明,基于短语的机器翻译模型,如 Pharaoh,只能有效地对局部调序进行建模.但语言学结构性的转化信息对机器翻译系统的全局调序作用非常明显.本文提出的基于 STSG 文法的模型能够有效地捕捉到此类信息,为翻译过程服务.

2) 同时,STSG 显示出对 SCFG 模型的绝对优势.这是因为,SCFG 规则只能模拟兄弟树节点之间的调序,而 STSG 规则却能模拟不同层次之间树节点之间的调序.这个明显的对比也证明兄弟树节点之间的调序远远不足以对中英翻译的词序调整进行建模.

3) 同时,我们还可以看出,SCFG 系统的性能也明显劣于 Pharaoh.主要原因在于,严格的句法限制使得 SCFG 系统无法使用大量的翻译片断对(从规则数对比可以清晰地看出);再有,SCFG 规则只能模拟兄弟树节点之间的调序,这一点严格限制了模型的调序能力.

Table 4 System performances

表 4 系统性能

Systems	BLUE4
Pharaoh	0.120 8±0.006 9
SCFG	0.086 7±0.004 8
STSG	0.139 4±0.007 3

6.2 实验2:基于自动标注句法语料的实验

6.2.1 实验设置

本实验在 2005 年 NIST 机器翻译评测中的中英翻译任务上进行.抽取规则集所用的训练句对来自语言学数据联盟(linguistic data consortium)提供的 FBIS 数据集(编号:LDC2003E14).该数据集经过过滤,包含 23 万 9 千多个句对,约 700 万汉语词汇和 900 万英语词汇.语言模型的训练语料是 Gigaword 语料中的新华部分,共包含 181M 英文单词.语言模型采用 SRILM 工具进行训练,设置元数为 4 元.

另外,训练数据需要作词对齐处理(基准系统和 STSG 系统均需要)和句法分析处理(只有 STSG 系统需要).在本文的实验中,先对训练数据进行两个方向的 GIZA++[23]词对齐,然后采用“grow-diag-final”启发式规则获得多对多的词对齐关系.源语言端(中文)和目的语言端(英文)的句法树均由现成的 Stanford 句法分析器[24]获得.其他实验设置与实验 1 相同.

评测准则本组实验利用了 BLEU 和 NIST[25]两种指标.其他实验设置与实验 1 相同.

6.2.2 系统设置

本实验采用的作为对比的基准系统是 Moses[16].它也是一个公开免费供研究使用的基于短语的机器翻译系统.除语言模型元数采用 4 元以外,其余设置均与 Pharaoh 一致(Pharaoh 只能使用元数为 3 的语言模型).

对 STSG 系统,规则中抽象节点个数的上限 c 设置为 6,规则中元树高度的上限 h 设置为 6,每个树节点对相关的抽象规则个数上限 ω 设置为 50.解码时的 *BeamThreshold* 设置为 100.SCFG 系统除了规则中元树高度的上

限 h 设置为 2 以外,其余设置均与 STSG 一致.

6.3 实验结果

表 5 和表 6 分别给出了 3 个系统在实验 2 数据集上的实验 BLEU 分值对比和 NIST 分值对比情况.

Table 5 Comparison BLEU results of Pharaoh, SCFG model and STSG model on 2005 NIST Chinese-English test set

表 5 在 2005 年 NIST 中英机器翻译评测集上 Pharaoh,SCFG 模型和 STSG 模型的对比结果(BLEU)

Systems	BLEU-*n*	*n*-gram precision						
	4	1	2	3	4	5	6	7
Moses	0.239 1	0.707 1	0.338 4	0.166 0	0.086 1	0.045 3	0.023 9	0.012 8
SCFG	0.219 0	0.686 3	0.307 8	0.145 4	0.074 8	0.038 5	0.020 0	0.010 2
STSG	**0.242 2**	0.699 4	0.332 4	0.166 8	0.088 7	0.046 9	0.025 2	0.013 6

Table 6 Comparison NIST results of Pharaoh, SCFG model and STSG model on 2005 NIST Chinese-English test set

表 6 在 2005 年 NIST 中英机器翻译评测集上 Pharaoh,SCFG 模型和 STSG 模型的对比结果(NIST)

Systems	NIST-*n*	*n*-gram precision						
	4	1	2	3	4	5	6	7
Moses	7.909 4	6.171 5	1.397 4	0.268 1	0.055 9	0.016 5	0.004 8	0.001 7
SCFG	7.711 5	6.091 7	1.315 8	0.238 9	0.050 1	0.015 0	0.004 3	0.001 4
STSG	**7.970 2**	6.211 5	1.415 5	0.268 7	0.057 4	0.017 1	0.005 4	0.002 1

从表 5 和表 6 中我们可以看出,与实验 1 结果一致,基于 STSG 的系统仍然稳定地超过另外两者.与实验 1 相对照,我们还可得出以下分析结论:

1) 实验 1 所用数据集为口语语料,中文平均句长小于 10 词,而实验 2 所用数据集可看成是篇章风格的长句子语料,平均句长超过 20 词.综合两组实验说明,新提出的 STSG 模型在两种风格差异较大的数据集上都能取得稳定的性能优势.

2) 实验 1 所用数据集包含人工校正过的句法标注,而实验 2 所用句法信息为自动句法分析器获得.在存在句法分析错误的情况下,基于 STSG 的系统仍能优于基于短语的系统.

3) 实验 2 为较大数据集的实验,而且词对齐采用完全基于统计的工具获得.一般情况下,在大数据集上的实验中,基于句法的系统很难取得对基于短语系统的绝对优势.但实验 2 的结果表明,STSG 的远距离调序能力以及异构对应能力有助于改善系统翻译性能.

另外,通过仔细分析表 5 中的 n-gram 精确度,我们发现在 1-gram 和 2-gram 的精确度上,STSG 系统均明显劣于 Moses.但是在 3-gram~7-gram 的精确度上,STSG 系统均稳定优于 Moses.BLEU 指标的一个特性就是它偏向于检测语言的流利度,而流利度主要由较长的词串体现.这个细节表明,STSG 系统可以得到流利度更高的句子,这在本质上也与长距离调序以及非连续短语模拟能力相关.

那么,为什么在 1-gram 和 2-gram 的精确度上 STSG 处于明显的劣势呢?一个最有可能的原因就是 STSG 系统中存在严格的句法限制(参见第 3 节中提到的句法限制 C2).我们将在后续的研究中对此进行深入研究,相信解决这个问题能使 STSG 系统获得进一步的性能提升.

7 结 论

本文提出一种基于同步树替换文法(STSG)的统计机器翻译模型.这一模型可以对不同语言之间的异构结构性对应进行建模,具有全局调序的能力,可以对非连续短语翻译现象进行模拟.本文给出了基于 STSG 文法的翻译模型的翻译规则抽取算法,并且针对抽取泛化规则过程中出现的冗余问题给出了解决方案.对于解码问题,本文提出一种自底向上的、树节点依次扩展的集束搜索算法.在两组风格差异较大的数据集上进行的实验均验证了基于 STSG 的模型相对于基于短语模型和基于 SCFG 模型的稳定优势.

在后续的研究中,我们将深入研究句法系统的句法限制过于严格的问题.

References:

[1] Zhao TJ, *et al*. The Principle of Machine Translation. Harbin: Harbin Institute of Technology Press, 2001 (in Chinese).

[2] Brown PF, Pietra SD, Pietra, VJD, Mercer RL. The mathematics of machine translation: Parameter estimation. Computational Linguistics, 1993,19(2):263−311.

[3] Koehn P, Och FJ, Marcu D. Statistical phrase-based translation. In: Proc. of the NAACL 2003. Edmonton, 2003. 48−54.

[4] Koehn P. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In: Proc. of the AMTA 2004. Washington, 2004. 115−124

[5] Och FJ, Tillmann C, Ney H. Improved alignment models for statistical machine translation. In: Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora. College Park: University of Maryland, 1999. 20−28.

[6] Zens R, Ney H, Watanabe T, Sumita E. Reordering constraints for phrase-based statistical machine translation. In: Proc. of the COLING 2004. Geneva, 2004. 205−211.

[7] Och FJ, Tillmann C Ney H. Syntax for statistical machine translation. Technical Report, Baltimore: Johns Hopkins University, 2003.

[8] Yamada K, Knight K. A syntax-based statistical translation model. In: Proc. of the ACL 2001. Toulouse, 2001. 523−530.

[9] Quirk C, Menezes A, Cherry, C. Dependency treelet translation: Syntactically informed phrasal SMT. In: Proc. of the ACL 2005. New York, 2005. 271−179.

[10] Liu Y, Liu Q, Lin S. Tree-to-String alignment template for statistical machine translation. In: Proc. of the ACL 2006. Sydney, 2006. 609−616.

[11] Wu D. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 1997, 23(3):377−403.

[12] Chiang D. Hierarchical phrase-based translation. Computational Linguistics, 2007,33(2):201−228.

[13] Galley M, Hopkins M, Knight K, Marcu D. What's in a translation rule? In: Proc. of the HLT/NAACL 2006. Boston, 2004. 273−280.

[14] Shieber SM. Synchronous grammars as tree transducers. In: Proc. of the TAG+7. Vancouver, 2004. 88−95.

[15] Och FJ. Minimum error rate training in statistical machine translation. In: Proc. of the ACL 2003. Sapporo, 2003. 160−167.

[16] Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E. Moses: Open source toolkit for statistical machine translation. In: Proc. of the ACL 2007. Prague, 2007.

[17] Lü YJ. Research on bilingual corpus alignment and automatic translation knowledge acquisition [Ph.D. Thesis]. Harbin: Harbin Institute of Technology, 2003 (in Chinese with English abstract).

[18] Stolcke A. SRILM—An extensible language modeling toolkit. In: Proc. of the ICSLP 2002. Colorado, 2002.

[19] Kneser R, Ney H. Improved backing-off for *M*-gram language modeling. In: Proc. of the ICASSP'95. Detroit, 1995.

[20] Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: A method for automatic evaluation of machine translation. In: Proc. of the ACL 2001. Toulouse, 2001. 311−318.

[21] Koehn P. Statistical significance tests for machine translation evaluation. In: Proc. of the EMNLP 2004. Barcelona, 2004.

[22] Zhang Y, Vogel S, Waibel A. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system. In: Proc. of the LREC 2004. Lisbon, 2004.

[23] Och FJ, Ney H. Improved statistical alignment models. In: Proc. of the ACL 2000. Hong Kong, 2000. 440−447.

[24] Klein D, Manning CD. Fast exact inference with a factored model for natural language parsing. In: Proc. of the NIPS 2002. Westin, 2002. 3−10.

[25] Doddington G. Automatic evaluation of machine translation quality using *n*-gram co-occurrence statistics. In: Proc. of the HLT 2002. San Francisco, 2002. 138−145.

附中文参考文献:

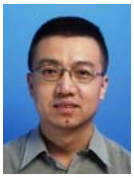
[1] 赵铁军,等.机器翻译原理.哈尔滨:哈尔滨工业大学出版社,2001.

[17] 吕雅娟.基于双语语料库对齐的翻译知识自动获取技术研究[博士学位论文].哈尔滨:哈尔滨工业大学,2003.

蒋宏飞(1982－),男,山西沁县人,博士生,主要研究领域为自然语言处理,机器翻译.

赵铁军(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为自然语言处理,机器翻译,基于内容的网络信息处理,人工智能应用.

李生(1943－),男,教授,博士生导师,主要研究领域为自然语言处理,机器翻译,基于内容的网络信息处理,人工智能应用.



张民(1970－),男,博士,高级研究员,主要研究领域为信息抽取,机器 翻译.

付国宏(1968－),男,博士,教授,CCF 高级会员,主要研究领域为自然语言处理,网络信息处理,生物信息挖掘.