基于实例的日汉机器翻译方法中的句子相似度计算研究

陈利人　陈群秀

清华大学计算机科学与技术系

摘要　基于实例的翻译方法中的核心是进行句子的相似度计算，论文作者对日汉机器翻译系统中源语句子分析时与实例库中实例句子相似度的计算进行了深入细致的研究，提出了结构相似度和语义相似度的概念，总结了句子中的词的对应关系的特性，设计了基于打分策略的带约束条件的动态规划算法，从而得到了句中的词的对应关系，进一步设计了一个原型的基于实例的日汉翻译系统分析器。

关键词　结构相似度　语义相似度　动态规划　递归分析　语义分类

Determining the Degree of Similarity between Sentences in Example_based Approach to Japanese_Chinsese Machine Translation

Chen Liren　Chen Qunxiu

Department of Computer Science & Technology ,Tsinghua University

Abstract　Determining the degree of similarity between sentences is the key problem in example_based machine translation method. On the basis of an at-depth research into the problem, this paper has proposed the concepts of Structure Similarity and Semantic Similarity and discuss the mapping of words between sentences. Moreover, it designs an algorithm of restricted dynamic programming and implements a parser of primitive translation system using example_based approach.

Keyword　Structure Similarity ; Semantic Similarity ; Dynamic Programming ; Recursive Analysis ; Semantic classification

一、引言

近年来，国际计算语言学界越来越重视对大规模真实文本的处理[1]，而提高自然语言分析器的鲁棒性和开放性是实现这一目标的关键。为了增强日汉机器翻译系统的鲁棒性和开放性，我们在基于动词配价语法、格语法和语义分类的日文分析的基础上[2][3]，再采用了基于实例的机器翻译方法。

基于实例的翻译方法有一系列的技术问题，例如实例的选择、存储、组合、查询、待翻句子与实例库中实例相似度计算等等。其中，句子的相似度计算是核心技术之一。

基于实例的翻译方法是Nagao首先提出来的[4][5][6]，它的基本思想是根据过去句子翻译经验来解决新的句子翻译问题；对于基于实例的方法来说，就是用已经翻译过的实例，通过仿效人类的类比思维来获得当前要翻译句子的结果。这就是我们常说的触类旁通。

例如，一个中国人第一次学会了翻译日语句子“私 は 学生 です 。”：“我是学生”，那么他以后遇到相似的句子：“彼 は 教授 です 。”他就能根据以前的经验理解该句子应翻译为“他是教授”，同时，他也能用类比的方法，自己造出许多类似的复杂结构的句子：“昨日 中国 へ 行った 人 は 社员 です 。”“彼 は 有名な 北京大学 の 学生 です 。”等等。因为以上句子具有共同的地方，也就是说具有很高的相似度。那么我们如何衡量这种相似性呢？我们认为句子的相似度包括两个方面：结构相似度和语义相似度。下面，本文详细论述了词的相似度，动态规划算法得到句子的相似度和词的对应关系，然后是语义相似度，最后进行了讨论和给出了实验结果。

二、词的相似度计算

首先，要解决词的分类问题，包括词性分类和语义分类。语义分类采用了计算名词或代词之间的相似度而设计的一个语义分类体系[7]。句子相似度包括结构相似度和语义相似度。因此，对词来说，也包括结构相似度和语义相似度。对于结构相似度，我们采用给分数值的策略。我们的分数范围为－２０到４０，这些分数值是一些实验结果。

表一

	は	の	P助	……
は	40	-20	-20	……
の	-20	40	-20	……
P助	-20	-20	40	……
……	……	……	……	……

打分的标准如下：１．助词、动词、形容词、形容动词（作谓语）完全相同（同一词），给40分；２．其它的词完全相同（除名词和代词），给１４分；３．对于形容词、形容动词、数词、量词、量长词、副词，词性相同给8分；４．对于名词和代词，由语义分类树计算：设W_1的语义分类标号为：$N_1N_2.....N_{w1}$，W_2的语义分类标号为：$M_1M_2.....M_{w2}$，则由右到左找它们的共同祖先(上位)，即比较N_i和M_i，直到$N_{i+1} \neq M_{i+1}$，则共同的祖先为i所处的层次，然后加分为$2 \times i$。5．对于助词，查找助词打分表(见表

一)，得到相应的加分或是罚分（分数为负）。6．P助与除助词以外的词均给予罚分：－20分(P助是系统假定的一个“虚”词)。7．其它的情形分数为0分。

一旦得到了句子中词与词之间的对应关系以后，我们可以进行词的语义相似度的计算了，词的语义相似度的计算如下：1．对于形容词(形容动词)作定语、副词、数词、量词、量长词，词性相同，则语义相似度为1.0；2．对于动词、形容词、形容动词作谓语，只要词的原形相同，则语义相似度为1.0；3．对于助词进行分类，同类中的的语义相似度为1.0，不同类中的语义相似度为0.0；4．对于名词、代词，根据语义分类树，按如下公式计算：待翻译句子中词的语义分类标号为$A_1A_2\cdots\cdots A_n$，实例中的词的语义分类标号为$B_1B_2\cdots\cdots B_m$，若$A_1=B_1,A_2=B_2,\cdots\cdots,A_i=B_i(i\le\min(m,n))$，则词的语义相似度为 $\text{WordSimilarity}(A_1A_2\cdots\cdots A_n,B_1B_2\cdots\cdots B_m)=\frac{i}{m}$

意义即是两个词的共同祖先(上位)所在的层次除以待翻译句子中的词所在树中的层次。

三. 句子的相似度计算

句子相似度包括：结构相似度和语义相似度。因此句子相似度计算包括两个步骤：

1．经过词的结构相似度计算后，得到一个由各词之间的分数值构成的矩阵，矩阵的行为待翻译句中的单词，列为某一实例中的单词。那么，问题就是在矩阵中寻找一条从左上角（起点）到右下角（终点）的路径，在满足路径的特性的情况下，使路径上各点的分数值的和最大，该值就定义为句子的结构相似度。

首先，我们讨论路径的特性：①单调性：由于句子的语法结构的限制，词与词之间的先后顺序对相似的句子而言应该是确定的，也就是说一个句子靠前的词与另一个句子靠前的词对应；靠后的词与靠后的词对应。因此，要求路径必须是从起点往下或是往右延伸的，即具有单调性(图一)，不会出现起伏(图二)。也就是说(i,j)点的下一步只能是$(i+1,j)$或$(i+1,j+1)$或$(i,j+1)$三种情形。②无直角性：一个句子与另一个句子相似时，只是句子中的一个词对应于另一个句子中的一个词或是多个词(词组或是扩展成分)；或是句子中的多个词(词组或是扩展成分)对应于另一个句子的一个词；不可能一个句子的多个词对应于另一个句子的多个词，若出现这种情况，则这种关系表现在路径上就是出现了直角。而路径中应无直角．也就是说，如果有部分路径$(i,j)\Rightarrow(i,j+1)$，则下一步只能是$(i,j+2)$或是$(i+1,j+2)$，不能是$(i+1,j+1)$；或是有部分路径$(i,j)\Rightarrow(i+1,j)$，则下一步只能是$(i+2,j)$或是$(i+2,j+1)$，不能是$(i+1,j+1)$。直角有两种情形(图三、图四)。③竖直路径远离终点（就近特性）：首先介绍三种路径：Ⅰ．竖直路径：待翻译句子中的多个词对应于实例中的一个词，如图五所示。Ⅱ．水平路径：待翻译句子中的一个词对应于实例中的多个词，如图六所示。Ⅲ．倾斜路径：待翻译句子中的词与实例中的词为一一对应关系，图中表现显而易见。因为一个助词词组或是一个由修饰成分和主词构成的词组一般有相邻的几个词构成，不会跨越几个无关的词，也就是就近特性，体现在图中即是竖直路径远离终点。④倾斜路径优先于竖直和水平路径：两个句子中的词一一对应时结构相似度高于多对一和一对多。

下面，我们采用带约束条件的动态规划来得到结构相似度，同时得到句子间词与词的对应关系。动态规划的核心是Bellman所提出的最优化原理[8]，它涉及多级决策过程最优化。设求解问题的数学模型为:

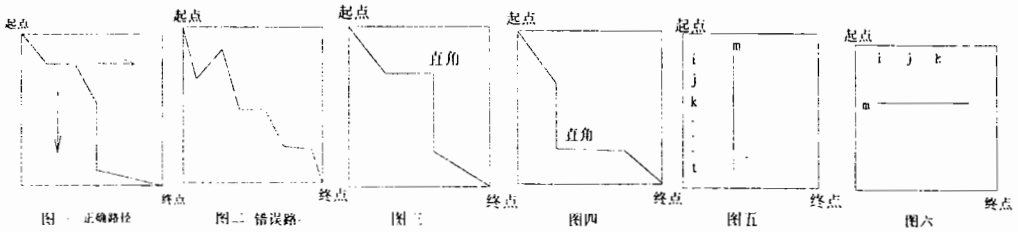
目标函数: $\max \quad (J) = \sum_{k=0}^{N} L[x(k), u(k), k)]$

约束条件: $X(k+1) = G[x(k), u(k), k]$ 即状态转移方程，表示状态$x(k)$，决策$u(k)$及下一段的状态$x(k+1)$的对应关系。若给定初始状态和一个决策过程$\{u(1), u(2)\ldots\ldots, u(N)\}$，则全过程就确定了。

令$I(x, k)$表示从k阶段的X状态到最后阶段的综合最优目标函数值，则动态规划的阶段最优递推公式为（后向动态规划）:

$$I(x,k) = \max_{u \in U} \{L(x,u,k) + I(G(x,u,k), k+1)\}$$
$$(k = 0, 1, \ldots\ldots N)$$

路径的特性为约束条件，采用以下的罚分机制来满足路径的特性: 最后得到的最优路径的值为两个句子的结构相似度。



图一 正确路径　图二 错误路径　图三　图四　图五　图六

计算路径的值和罚分机制: ①路径的值为路径上各点的分数值之和(图七)（除了以下其它机制的要求）: 则路径的值为$a_{i-i,j-1} + a_{i,j}$。②对于水平路径上的点，除末点外，其它点不计入路径的值(图八): 则路径的值为$a_{i-1,j-2} + a_{i,j}$，而$a_{i,j-1}$不计入路径的值。③对于竖直路径上的点，除末点外，其它点不计入路径的值(图九): 则路径的值为$a_{i-2,j-1} + a_{i,j}$，而$a_{i-1,j}$不计如路径的值（①、②、③说明了只有对应的主词才对句子的结构相似度有贡献）。④对于水平路径的罚分(图十): 则路径的值为$(a_{i-1,p_{j-1}} + a_{i,p_{j+1}}) - (p_{j+1} - p_j) \times 2$。⑤对于竖直路径的罚分(图十一): 则路径的值为$(a_{p_{i-1},j-1} + a_{p_{i+1},j}) - (p_{i+1} - p_i) \times j$（④、⑤为了满足倾斜路径优先和就近特性）。⑥对于直角路径的超重罚分(图十二): 则路径的值为$a_{i,j} - 100$（⑥为了满足路径的无直角特性）。⑦为了满足路径的单调性，(i, j) 的后一点只能是(i + 1, j) 或(i + 1, j + 1) 或(i, j + 1) 这三点。如图十三所示。

计算结构相似度的动态规划算法: 设矩阵为$m \times n$，我们要得到(1,1) 点(起点)到(m,n)点(终点)的结构相似度最大的最优路径。

(一)计算$\begin{bmatrix} (m-1,n-1)\{3\} & (m-1,n)\{2\} \\ (m,n-1)\{1\} & (m,n) \end{bmatrix}$中(m - 1, n - 1) 到$(m,n)$的最优路径，按$\{1\},\{2\},\{3\}$的顺序计算，因为$\{3\}$要用到$\{1\},\{2\}$的路径值。

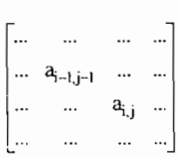
㈡如果行或列值大于1，行减少1，列减少1，否则保持不变，即计算

$\begin{bmatrix} (m-2,n-2)\{5\} & (m-2,n-1)\{4\} & (m-2,n)\{3\} \\ (m-1,n-2)\{2\} & (m-1,n-1) & (m-1,n) \\ (m,n-2)\{1\} & (m,n-1) & (m,n) \end{bmatrix}$中$(m-2,n-2)$到$(m,n)$的最优路径，按$\{1\},\{2\},\{3\},\{4\},\{5\}$的顺序计算。

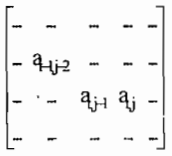
㈢重复㈡，直到行和列分别到达1(即起点)，计算完成。由于计算过程中每个点均保留最优路径上前一点的位置,因此可以找出从起点到终点的最优路径来，同时得到结构相似度。概括来说，计算方向如图十四所示。

㈣对最优路径上的点从右往左投影，得到$m\times 1$维矩阵(矩阵中的元素为投影后所见到的元素)，此时得到待翻译句子中的词与实例的词的对应关系。

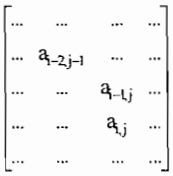
例如：待翻译的句子为：田中さんは电车で会社へ行きます。实例为：私は行く（动词原形）。得到分数矩阵为(见表二)。经过计算相似度的动态规划算法得到起点到终点的最优路径为(0,0)→(1,0)→(2,1)→(3,2)→(4,2)→(5,2)→(6,2)→(7,2)，路径的值，即句子的结构相似度为84 。表示成矩阵的形式为(1表示路径上的点，否则为0)(矩阵二)。 然后，按⇐从右往左投影得到词与词的对应矩阵(矩阵一)，最后得到待翻译句子与实例中的词的对应关系如图十五所示。



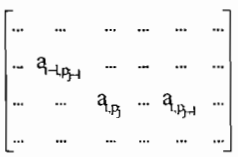
图七



图八

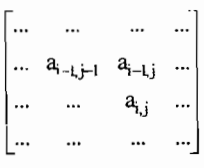


图九

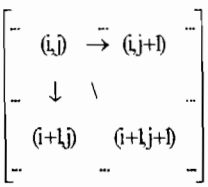


图十

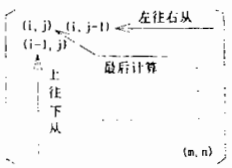
图十一



图十二

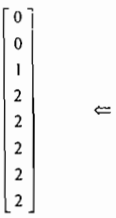


图十三



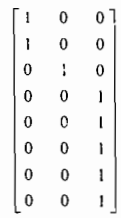
图十四

矩阵一



⇐

矩阵二



2．将句子的词的对应关系建立以后，就可进行句子语义相似度的计算了。计算句子的语义相似度时，只进行主词的相似度的计算，如（田中さん）中的主词为さん。

句子的语义相似度定义为:

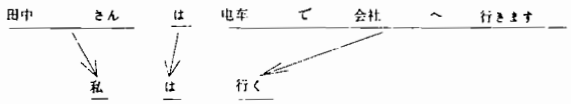
$$\text{SentenceSimilarity} = \frac{\sum \text{WordSimilarity (Mapping KeyWord)}}{n\,(\text{Number of Words in Example Sentence})}$$

因此，我们寻找最相似的实例时，先以结构相似度为准，找出结构最相似的实例，然后计算语义相似度，只有语义相似度大于一定的阈值(设定为70%)，才认为该待翻译的句子在实例库中有相似的实例。

最后，补充一下路径的校正问题：例如有多个名词在一起$N_1N_2N_3N_4$，此时对应的主词可能是N_2，其实应该是N_4，所以应该调整路径，使N_4成为主词。同时对于数词+量词，名词+の等连续出现的情形也应该进行校正。实际上是捆绑的思想。

表二

	私 0	は 1	行く 2
田中 0	10	0	0
さん 1	12	0	0
は 2	0	40	0
电车 3	4	0	0
で 4	0	0	0
会社 5	8	0	0
へ 6	0	0	0
行きます 7	0	0	40



图十五

四、复杂结构句子的递归分解为简单句子或是助词词组的相似度计算

一个复杂句子通常可以分解成几个简单的句子，因此可以进行递归分析生成。先从右到左扫描基本词典中未能查到的词，然后以该词作为索引词检索实例库；若是存在实例，则进行相似度计算，找出最相似的实例，得到两个句子之间词与词的对应关系。对于多个词对应一个词时，可以找到多个词中的主词；然后将余下部分又作为新的句子继续检索实例库的工作，直到全部生成；若是不存在实例，则认为该词是未登录词，直接输出日文原文，余下的部分作为新的句子继续进行检索实例库的工作，直到全部完成。

在此过程中，采用的是从右到左找相似实例定出待翻译句子的主干，然后根据实例的汉语译文生成部分结果；对于每个部分采用深度优先的策略直到全部生成。总结来说就是：①从右到左定主干；②每个部分深度优先；③边分析边生成。

例如：　电车で东京へ行った人は田中さんです。首先“。”号作为一个实例[。][。](为了节省实例的存储，把[。]和[か。]作为实例存储)，那么将句子分析生成为：(电车で东京へ行った人は田中さんです)。然后检索“です”的实例，得到最相似的实例[王は大学の学生です] [王是大学的学生]，分析并生成汉语译文：(电车で东京へ行った)人是田中先生。然后检索“行った”的实例，得到最相似的实例[私は日本へ行く

N定] [我到日本去的], 分析并生成: (电车で)到东京去的人是田中先生。最后检索"で"的实例,得到最相似的实例[电车で] [乘电车], 分析并生成: 乘电车到中国去的人是田中先生。到此时分析与生成全部完成, 输出译文。

五、讨论

1. 复杂度分析: 假定待翻译的句子有N个词, 每个词有M个语义项, 则与一个实例要计算相似度的次数为M^N; 若再有K个实例, 则计算相似度的次数为$K \times M^N$。可见复杂度相当高, 因此, 在设计的实例库, 使用了一些技巧, 尽量减少实例的数量。另外, 动态规划算法的复杂度为$m \times n$, m为待翻译句子中的单词数, n为实例中的单词数。所以总的复杂度为$(m \times n) \times K \times M^N$ (但是一般来说一个日文单词一般只有一个语义, 即$N = 1$)。

2. 对于有些复杂的句子与实例计算相似度以后, 由于此的对应关系错误, 不能翻译成功, 因此最好能将规则的方法和实例的方法结合使用。

3. 一个复杂结构句子分解为简单句子和助词词组过程(由于篇幅所限, 没能列出)。

六、 结束语

基于实例的翻译方法中最核心的问题是句子的相似度计算, 我们对句子的相似度提出了结构相似度和语义相似度, 并设计了相应的带约束条件的动态规划算法, 同时采用了人工智能中常用的打分策略, 并且进行了大量的实验和设计了一个简单的基于实例的翻译系统分析器, 此系统主要是为了完善实用翻译系统的学习功能。

参考文献

[1]黄昌宁 "关于处理大规模真实文本的谈话", 《语言文字应用》, 1993年第2期

[2]冯昶 《一个实用型日汉机器翻译系统的研究和初步实现》 清华大学计算机系硕士论文, 1994年6月

[3]孙勇 《日汉机器翻译系统分析器的改进与实现》 清华大学计算机系毕业设计论文, 1995年6月

[4]佐藤理史 "MBT1: 实例に基づく译语选择"

[5]佐藤理史 "MBT2: 实例に基づく翻译における复数翻译例の组合せ利用"

[6]佐藤理史、长尾真 "实例に基づいた翻译"

[7]陈群秀、张普 "信息处理用现代汉语语义分类体系(之一): 属性分类" 《计算语言学研究与应用》

[8]张润琦编 《动态规划》 北京理工大学出版社

[9]Hiroaki Kitano "A Comprehensive and Practical Model of Memory-based Machine Translation"

[10]Eiichiro SUMITA "Example-based Machine Translation on Massively Parallel Processors"

[11] "Search for Solutions Combining Classification Rules with Case-based Reasoning" PRICAI'92

[12]Sasao Kurohashi、Makoto Nagao "A Syntactic Analysis Method of Long Japanese Sentence Based on the Detection of Conjunctive Structure" *Computational Linguistics* 1994