HSK 动态作文语料库偏误标注方法研究

王洁，宋柔
北京语言大学语言信息处理研究所
<wangjie;songrou>@blcu.edu.cn

摘要：HSK 动态作文语料库是一个大规模的汉语中介语语料库，人工对其中各类偏误进行了标注。针对人工标注的缺陷，本文提出了偏误自动标注的方法。基于编辑距离算法，以汉语的词为单位通过进一步求解编辑路径发现修正原句所需要的基本编辑操作，从而实现了原句和修正句的自动比对。这样，能够较好地弥补当前中介语语料库标注方法的缺陷，体现了人机的优势互补。

关键词：偏误，编辑距离，编辑路径，自动标注

A STUDY ON AUTO TAGGING OF ERRORS IN HSK ESSAY CORPUS

Wang Jie, Song Rou
Center for Language Information Processing
Beijing Language and Culture University
<wangjie;songrou>@blcu.edu.cn

Abstract: HSK Essay Corpus is a large Chinese interlanguage corpus in which each error type has been tagged manually. In order to make up for the deficiency of the manual work, this paper proposed an error auto tagging method. With the use of the famouse edit distance algorithom and taking the Chinese "word" as the unit, computer can follow the edit path to work out the foundmental operations required to correct the original sentence. This proposal can improve the interlanguage corpus tagging method, and shows a good cooperation between human and computer.

Key words: error, edit distance, edit path, auto tagging

1HSK 动态作文语料库

"**HSK 动态作文语料库**"是母语非汉语的外国人参加高等汉语水平考试（HSK 高等）作文考试的答卷语料库，收集了 1992-2005 年的部分外国考生的作文答卷，共计 10740 篇，约 400 万字。语料加工方式为人工标注字、词、句、篇、标点各类偏误。

2 人工标注的缺陷

人工标注的缺陷体现在标注质量和标注速度两方面：

首先，质量方面。

（1）标注结果缺乏一致性。这主要由归类本身存在歧义性而导致。具体而言，有两种情况：一种情况是某一偏误只有一个改法，但存在多种归类方式，比如句子都少了"方面"一词，例 1)标注的是缺中心语，而例 2)标注的是缺词；另一种情况是某一偏误有多个改法，也存在多种归类方式，比如例 3)4)5)的改法可以是去掉"有"，也可以是去掉"感"，3)标注的是多词"有"，

4)标注的是多述语“感”，5)标注的是“有”字句病句。

1) 可[C]是社会在提高人们的文化水平、道德水平{**CJ-zxy 方面**}起了很大的作用。
2) 在公众利益{**CQ 方面**}，吸烟更是影响深切。

3) 我对流行歌曲{**CD 有**}感兴趣。
4) 她们对我有{**CJ+sy 感**}兴趣，
5) 我对贵公司的这一次招聘特有感兴趣{**CJy**}，

（2）误归类和遗漏。这主要是因为标注者未能想清楚或一时疏忽造成的。如例6)标注的是补语多余，实为状语多余；例7)黑体部分有毛病，但没有给出标记。

6) “绿色食品”出场以后{CJ+buy 被人们}很受欢迎。
7) 也可以说年青人对看电视听音乐这些方面很**有感兴趣**。

（3）标记不合规范。虽然目前有专门开发的用于作文库标注的工具软件[1]，但仍然不能杜绝此类问题。见例8)。

8) 但我父亲从没**埋怨**{**CC 叹**}过一声辛苦，[BC、]
注：{CC}代表错词，按照标注规范的要求，原句字数和改后字数不一致的，需在括号中CC之后且紧靠CC处加一个阿拉伯数字，表明改后的字数。此句的正确标注方法是“但我父亲从没**埋怨**{**CC2 叹**}过一声辛苦，[BC、]”。

（4）个别标记设计不合理。比如：语序偏误的标注方式是在顺序颠倒的两个部分之间插入标记{CJX}，但前后两部分各自的边界并没有标记，见例9)。

9) **十五岁**{**CJX**}**当我**上高中时我又{CD 再次}失去我的的父亲[F 親]，

其次，速度方面。
标注者从观察原文到给出标记的过程可分为如下步骤：(1)判断对错；(2)如果错了——(2.1)头脑中映射出正确的表达；（2.2）对偏误进行归类；（2.3）回忆甚至查阅标注规范；（2.4）添加标记。其中（2.2）（2.3）（2.4）平添了标注者的负担，势必影响标注速度。

3 偏误的自动标注

对于绝大多数偏误，人可以进行纠正。对于任意两个字符串，计算机都可以进行比对，并通过一定的形式标记将比对结果表示出来。据此，我们提出偏误自动标注的方法，让人来完成对意

[1] 该软件的工作方式是：标注者选定错误部分，单击鼠标右键，从弹出菜单中选择错误类型，然后计算机自动添加标记。这主要是为了减轻标注负担、加快标注速度以及防止漏掉半个括号、写错字母等情况发生。

义的理解，针对原句给出修正句；计算机来发现原句和修正句之间形式上的差异，从而实现人机的优势互补。

整个标注过程分为三个步骤：

首先，人工对偏误直接修改，而不是添加标记。

人对偏误的第一反应是正确的表达是什么而非偏误的类型是什么。因此，采用对偏误直接修改的方式可以大大减轻标注者的负担，同时能够避免标记误用的情况。例 10)给出了某一偏误所在的原句、人工直接修改的修正句。当存在多种改法时，可以对应给出标注者想到的所有修改结果。

10)

原句：我希望当广告设计师工作。

修正句：我希望做广告设计工作。

　　我希望当广告设计师。

其次，计算机对原句和修正句进行比对自动后添加标记。

计算机通过比较原句和修正句，可以自动添加标记。在此步骤中，只需关注构成比较双方的语言符号在形式上的差别，而不涉及意义，这正是计算机擅长处理的字符串比较问题，属典型计算问题，计算机可以很好地完成。字符串比较涉及的基本操作有三种：插入、删除、替换，本文根据需要又增加了两种操作：移位、交换，在 4.3 中将具体介绍。例 10)经计算机自动比对的结果如下，{rep 当 with 做}表示将“当”替换成“做”，{del 工作}表示删除“工作”。

10)

原句：我希望当广告设计师工作。

修正句：我希望做广告设计工作。

　　我希望当广告设计师。

比对结果：我希望{rep 当 with 做}广告设计{del 师}工作。

　　我希望当广告设计师{del 工作}。

再次，集中起来人工归纳偏误类型。

全部如此处理完毕后，人工在此基础上再进行偏误类型归纳，这样便于从宏观上把握，从而可以保证较好的一致性。人工标注时一致性较难把握的方面主要有：字和词的区分；词和句子成分的区分；词和句的区分。

字和词有时不太好区分，比如例 10) 的修正句之一是“我希望做广告设计工作。”，原文中“广告设计师”的“师”多余，“师”到底算字还是算词，不同的标注者可能会给出不同的判断。采用本文建议的方法，就可以将所有出现{del 师}的句子集中起来，统一决定全部算作多字，还是全部算作多词。

标注规范对词和句子成分加以区分，原则是看多余、缺失或误用的部分是否能充当句子成分，能者按句子成分处理，不能的则按词处理。然而，从当前已有的标注结果看，这一标准对人来说也是很难把握的，因此本文建议在人工归纳时，“句子成分”的类别还是慎用为好。

标注规范也对如何区分词一级偏误和句一级偏误作了说明，但从已有的标注结果看，标注者同样很难把握。如果让计算机介入进来，根据自动比对的结果，人工再归类时一致性就容易把握

了。比如对所有出现标记{del 有}的句子，统一决定算作“有”字句病句，还是算作多词“有”，这样就可以避免人工直接标注时的随意性。

4 基于编辑距离算法的中文句子自动比对

4.1 编辑距离算法

编辑距离(Levenshtein Distance 或 Edit Distance)是由俄国科学家 Vladimir Levenshtein 于 1965 年提出的，用于计算字符串的相似度。从源串 s1 转换成目标串 s2，只用删除、插入、替换三种操作，需要的最少操作次数就是 s1 到 s2 的编辑距离。

两字符串之间的编辑距离 d(s1,s2)有如下规定：

规定 1：d (", ") = 0　　// "代表空串，两个空串的编辑距离为 0
规定 2：d (s, ") = d (", s) = |s|　　//字符串 s 和空串的编辑距离为 s 的长度|s|
规定 3：d (s1+ch1, s2+ch2)　　// s1、s2 代表字符串，ch1、ch2 代表字符
= min (d (s1, s2) + if ch1=ch2 then 0 else 1,　//函数 min (a,b,c)取最小值
d (s1+ch1, s2) + 1,
d (s1, s2+ch2) + 1)

规定 1、规定 2 容易理解，在此对规定 3 做出解释。从字符串 s1+ch1 到字符串 s2+ch2，对字符 ch1 和 ch2 的操作有四种情况：一是当 ch1=ch2 时，不进行任何操作，此时 d(s1+ch1, s2+ch2)= d(s1, s2)；二是当 ch1≠ch2 时，将 ch1 替换成 ch2，此时 d(s1+ch1, s2+ch2)= d(s1, s2)+1；三是将 s1+ch1 转换成 s2，再插入 ch2，此时 d(s1+ch1, s2+ch2)= d(s1+ch1, s2)+1；四是将 s1 转换成 s2+ch2，再删除 ch1，此时 d(s1+ch1, s2+ch2)= d(s1, s2+ch2) + 1。从所有情况中取最小值作为 d(s1+ch1, s2+ch2)的最终取值。

我们用动态规划算法来求解编辑距离。建立二维矩阵 m[0...|s1|,0...|s2|]，m[i,j]用来存放源串 s1 的前 i 个字符构成的子串和目标串 s2 的前 j 个字符构成的子串的编辑距离，即 m[i,j] = d(s1[1..i], s2[1..j])，m[|s1|,|s2|]的值即为 s1 和 s2 的编辑距离。

4.2 编辑路径的求解

编辑距离反映的只是两个字符串的相似度，并没有反映从源串到目标串的转换过程。相对于编辑距离，我们将完成转换所需的最短操作序列称为编辑路径。编辑路径反映的是字符串间的转换过程。本文工作要求显式地给出转换过程，即表示出源串的哪些位置发生了哪些操作，因此我们需要通过求解编辑距离来进一步求解编辑路径。算法补充如下：

建立二维数组 p[0...|s1|, 0...|s2|]，p[i, j]用来存放路径方向，从 p[|s1|, |s2|]沿路径方向回退到 p[0, 0]就得到了逆向的编辑路径。

编辑路径可以有多条，编辑路径的选择取决于求解过程中替换↖、插入←、删除↑三种操作的优先顺序。三种操作共有六种排列情况：↖←↑，↖↑←，←↖↑，←↑↖，↑↖←，↑←↖。只要在算法实现时设定一种优先顺序，得到的编辑路径就是唯一的。

当存在多条编辑路径时，为了使求得的编辑路径更符合人的编辑习惯，如何设定三种操作的优先顺序也值得研究。例如字符串 s1“sport”转换成 s2“resort”有三条编辑路径，见图 1。显然图 1 a 更符合人的编辑习惯，因为人在编辑时倾向于保留源串和目标串中相同的字符。为了尽量符合人的编辑习惯，我们将↖设为最后，这时三种操作只剩两种排列情况：←↑↖，↑←↖。就实例 s1“sport”向 s2“resort”的转换而言，要实现图 1 a 应设定的优先顺序为↑←↖，但就实例 s2“resort”向 s1“sport”的转换而言，则设定的优先顺序为←↑↖时才符合人的习惯。因此，←和↑哪个优先还需视具体情况而定。我们采取的策略是：按←↑↖和↑←↖两种优先顺序分别求解编辑路径，设定其中一种为默认的优先顺序，如果两种优先顺序求得的编辑路径中替换操作数量相同，则采用按默认优先顺序求得的编辑路径；否则采用替换操作较少的编辑路径。在本文实验中，将默认的优先顺序设为←↑↖。

图 1

		s	p	o	r	t
r	e	s		o	r	t
插入	插入		删除			

a

s	p		o	r	t
r	e	s	o	r	t
替换	替换	插入			

b

	s	p	o	r	t
r	e	s	o	r	t
插入	替换	替换			

c

4.3 中文句子自动比对

我们基于编辑距离和编辑路径的求解算法，让计算机对汉语的句子进行自动比对。三种基本操作的标记及含义为：1.{insX}：插入 X；2.{delX}：删除 X；3.{repXwithY}：把 X 替换成 Y。

自动比对分为以下三个步骤：

步骤 1，自动分词。

中文是用汉字来书写的，一个中文句子可以看作一个汉字串。但是，以汉字为单位进行句子间的比对往往会不尽如人意，见例 11)，<s></s>之间的部分为原句，<t></t>之间的部分为修正句，<res></res>为计算机自动比对的结果。

11)

<s>既然我得感冒，</s>

<t>即使我得感冒，</t>
<res>{rep 既 with 即}{rep 然 with 使}我得感冒，</res>

为了避免此类的情况，我们先对句子进行自动分词，然后以词为单位进行比对，这时 11)中的比对结果就能如人所愿了，即“<res>{rep 既然 with 即使}我得感冒，</res>”。

步骤 2，求解编辑路径。

根据 4.1 和 4.2 中介绍的算法，以词为单位求解编辑路径。

步骤 3，对编辑路径进行后处理。

对编辑路径进行后处理，主要是为了让计算机比对的结果更加符合人的编辑习惯，因为计算机只是机械地对符号的形式进行比对，而并非理解了符号的意义，比对结果不可避免地会带有机器味。具体来说，有以下几种情况还需进一步处理：

首先，需要合并的情况。当一个替换操作紧跟一个删除操作时，将删除操作并入替换操作，即将{repXwithY}{delZ}合并为{repXZwithY}，原因是 X 和 Z 很可能是 Y 对应的一个错词，正是由于原句写错才被切碎。如例 12)中，“欠少”应该为“缺少”，在原句中被切碎，比对结果就成了{rep 欠 with 缺少}{del 少}，将两个操作合并成{rep 欠少 with 缺少}更符合人的编辑习惯。当一个替换操作紧跟一个插入操作时，即{repXwithY}{insZ}的情况不合并为{repXwithYZ}，因为 Y 和 Z 相邻出现在修正句中，多数情况下确实是两个词，见例 13)。

12)

<s>石油/成为/故事/中/的/水/一样/不可/欠/少/的/东西/，</s>
<t>石油/成为/故事/中/的/水/一样/不可/缺少/的/东西/，</t>
<res>石油成为故事中的水一样不可**{rep 欠 with 缺少}{del 少}**的东西，</res>
处理为：<res>石油成为故事中的水一样不可**{rep 欠少 with 缺少}**的东西，</res>

13)

<s>对/我/个人/来说/一/首/歌/最/重要/要/能/表达/出来/个人/的/想法/，</s>
<t>对/我/个人/来说/一/首/歌/最/重要/的/是/能/表达/出来/个人/的/想法/，</t>
<res>对我个人来说一首歌最重要**{rep 要 with 的}{ins 是}**能表达出来个人的想法，</res>

其次，移位的情况。当{delX}前面或后面的某个位置出现{insX}时，按照人的编辑习惯，应该是 X 发生了移位，即从删除的位置移到了插入的位置。由此，我们设计了“移位”操作，标记为：{mov$i_X} {mov$i}，含义为：把 X 移到{mov$i}所在处。$i 代表编号，因为从原句转换成修正句可能会发生不止一次的移位，有了编号就有了移动对象和移动终点的对应。见例 14)。

14)

<s>人/的/感情/也/把/所有/的/问题/不/能/解决/，</s>
<t>人/的/感情/也/不/能/把/所有/的/问题/解决/，</t>
<res>人的感情也**{ins 不}{ins 能}**把所有的问题**{del 不}{del 能}**解决，</res>

处理为：<res>人的感情也**{mov1}{mov2}**把所有的问题**{mov1_不}{mov2_能}**解决，</res>

再次，互换的情况。当{repXwithY}和{repY withX}共现时，按照人的编辑习惯，应该是 X 和 Y 互换。由此，我们设计了“互换”操作，标记为：{exc$i_X} {exc$i_Y}，含义为：X、Y 互换。$i 代表编号，有了编号就有了互换双方的对应。见例 15)。

15)

<s>看见/一/个/小/的/漂亮/女/孩/。</s>

<t>看见/一/个/漂亮/的/小/女/孩/。</t>

<res>看见一个**{rep 小 with 漂亮}**的**{rep 漂亮 with 小}**女孩。</res>

处理为：<res>看见一个**{exc1_小}**的**{exc1_漂亮}**女孩。</res>

5 结语

本文研究的目的是针对目前 HSK 动态作文语料库标注中存在的缺陷，提出一些改进的建议，具体的思路是分三步走：首先，人工对偏误直接修改；其次，计算机对原句和修正句进行自动比对；再次，人工参照自动比对的结果对偏误进行归类。偏误自动标注的目并非要取代人，而是为了更好地辅助人。

参考文献

[1]Michael Gilleland, Levenshtein Distance, in Three Flavors, http://www.merriampark.com/ld.htm.

[2]Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest, Clifford Stein, 2002, 算法导论（第二版，影印版），高等教育出版社。

[3]马希文，1986，计算机与思维科学，逻辑.语言.计算——马希文文选，商务印书馆，2003。

[4]张宝林，2006，“HSK 动态作文语料库”标注说明。

[5]邹旭凯，汉字/字符串编辑距离和编辑路径的有效求解技术。