五阶段训练生成定向猜测算法

变换操作

五阶段训练生成算法支持的变换操作有删除段、交换段位置、插入段、删除字符、插入字符、大小写变换、口令复写，变换操作汇总见表。

表 1 五阶段训练生成算法口令变换操作

结构级变换	删除段 DelSeg(m,n) 交换段 SwapSeg(n) 插入段 AddSeg(m,n)	删除长度为 n 的 BS 结构的第 m 个段 当前段与其后第 n 个段交换位置 长度为 n 的 BS 结构的第 m 个间隙插入新段
字段级变换	删除字符 DelChar(m,n) 插入字符 AddChar(m,n)	删除长度为 n 的段的第 m 个字符 长度为 n 的段的第 m 个间隙插入新字符
特殊变换	大小写变换 C 口令复写 Repeat	大小写变换 重复一遍口令

训练过程

训练模型时，对于训练集中的每一对口令，会模拟从旧口令逐阶段变换为新口令的过程，这个过程主要分为五个阶段，每个阶段是可循环的，可以用图 1表示流程。

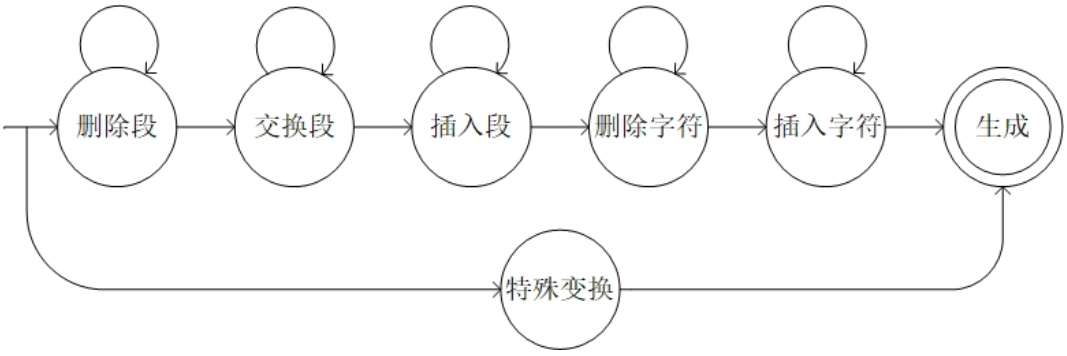

图 1 五阶段流程图

首先，算法会计算旧口令与新口令的编辑距离 (也称 Levenshtein 距离)d，可按以下方式计算两个口令的相似度 s:

$$s = d/max(len(oldpsw), len(newpsw))$$

相似度 $s(0 \leq s \leq 1)$ 越小，代表两个口令越相似，当 $s = 0$ 时，则代表两个口令完全相同。算法会抛弃 $s \geq 0.5$ 的口令对，避免相差很大的口令对影响训练结果。

在删除段阶段，训练目标是某个长度的 BS 结构删除某个位置的段的概率。为了获知删除旧口令中的哪些段，算法会先解析出两个口令的 BS 结构 (L、D、S 三种段)，随后计算两个口令中相同类型段两两之间的相似度，计算方式与口令相似度计算类似，对于 $s < 0.4$ 的段结构对，认为两个段结构是相似的，为旧口令中的每个段从新口令中找到它能匹配的最相似段，匹配完成后，旧口令中的每个段会匹配到 0 或 1 个新口令中的段，同样新口令中的每个段会匹配到 0 或 1 个旧口令中的段，匹配示例见图 2。旧口令中匹配度为 0 的段会被删除，并且旧口令 BS 长度与被删除口令所在位置对应的统计数组会被更新用于生成训练模型。

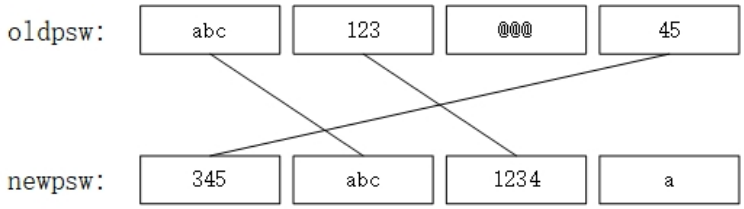

图 2 段结构匹配

旧口令经过删除段后，进入交换段阶段，训练目标是当前段与其之后段交换位置的概率。本阶段会利用在删除段阶段获得的匹配信息，循环选中目标位置最靠前的段，与当前段交换位置，并根据位置之间的距离更新 swapseg 数组，图 2的旧口令在该阶段的处理过程如图 3。

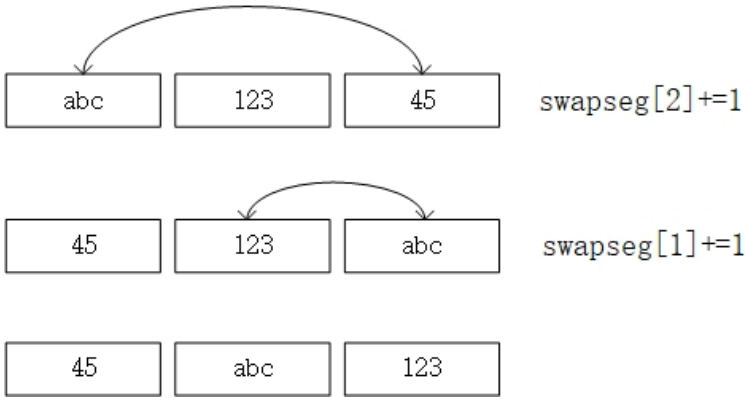

图 3 段结构交换

在插入段阶段，训练目标是某个长度的 BS 结构在某个间隙插入段的概率以及插入段类型、长度的概率，不包括插入段内容的概率，插入段具体内容的概率

由 PCFG 的训练结果确定。经过删除段、交换段，旧口令只需在对应间隙插入新口令中匹配度为 0 的段。

以上三个阶段完成了结构级变换的训练，接下来进入字段级变换的训练：删除字符阶段、插入字符阶段。在字段级变换训练中，是以段为单位的，只考虑结构级变换训练中得到的相似段对，不考虑插入的段，每一对相似段都需要经历删除字符阶段、插入字符阶段。

在删除字符阶段，训练目标是某个长度的段删除某个位置的字符的概率。为了获知删除段中哪些字符，首先利用动态规划得到两个段的最长公共子序列，旧口令段中的某个字符若在最长公共子序列中，则代表该字符需要保留，否则需要删除。

在插入字符阶段，训练目标是某个长度的段在某个间隙插入字符的概率，不包括具体插入字符内容的概率，该概率由 Markov 的训练结果确定。经过删除字符阶段，旧口令段转换成了两个段的最长公共子序列，还需要插入新口令段中不在最长公共子序列中的字符。

五阶段训练过程以外还有对特殊变换的训练，该训练相对来说简单一些，检查旧口令是否可以通过特殊变换得到新口令即可。

生成猜测

生成猜测阶段从旧口令到猜测口令的流程依然如图 1所示，与训练流程保持一致性。genguess 函数以一个旧口令为输入，以所需猜测数目的猜测口令序列为输出，函数借助优先队列完成整个生成流程，伪代码见算法 1。

Algorithm 1 生成猜测算法 genguess

输入： 旧口令 oldpsw

输出： 猜测口令序列 newpswlist

1: 初始化优先队列 Q，队列元素为 Password 类，包含口令字符串、阶段号、概率等信息，阶段号越大、概率越大的元素越靠前；
2: 初始化概率阈值为较小浮点数；
3: 将 oldpsw 初始化，阶段号设为删除段阶段号，概率设为 1.0，加入队列中，随后将阶段号设为特殊变换阶段号，再次加入队列中；
4: 取出队首元素，根据其阶段号分配到不同的处理模块；
5: 删除段模块 (阶段号为 1)：逐段检查删除概率至尾部，若删除后概率大于当前阈值，则将该段标记删除 (并非真正删除)，阶段号仍为 1 加入 Q 中。检

查步骤从最后被标记删除的段开始，以避免产生重复删除。检查结束后，将本口令带删除标记的段真正删除，修改阶段号为 2 加入 Q 中；

6: 交换段模块 (阶段号为 2)：逐段检查当前段与之后段交换位置后概率是否大于当前阈值，若大于则将交换后的口令的当前段原位置标记交换，阶段号设为 2 加入 Q 中。检查步骤从最后被标记交换的段开始，以避免重复交换。检查结束后，将本口令阶段号修改为 3 加入 Q 中；

7: 插入段模块 (阶段号为 3)：逐段间隙检查插入概率至尾部，计算插入段类型、长度、具体内容概率，若插入后概率大于当前阈值则插入新段，在新段位置标记插入，阶段号设为 3 加入 Q 中。检查步骤从最后被标记插入的段后间隙开始，以避免重复检查插入。检查结束后，将本口令阶段号修改为 4 加入 Q 中；

8: 删除字符模块 (阶段号为 4)：逐字符检查字符所在段删除该字符后的概率，若大于当前阈值则将该字符标记删除 (并非真正删除)，阶段号仍为 4 加入 Q 中。检查步骤从最后被标记删除的字符开始，以避免重复删除，并且检查时会跳过插入的段。检查结束后，将本口令带删除标记的字符真正删除，修改阶段号为 5 加入 Q 中；

9: 插入字符模块 (阶段号为 5)：逐字符间隙检查字符所在段在该间隙插入字符后的概率，字符具体内容概率由 Markov 模型训练得到，插入字符后的总概率若大于当前阈值，则插入字符并标记，阶段号设为 5 后加入 Q 中。检查步骤从最后被标记插入的字符后间隙开始，以避免重复检查插入。检查结束后，将本口令阶段号修改为 7 加入 Q 中；

10: 特殊变换模块 (阶段号为 6)：对口令进行大小写变换 (全大写、全小写、首字母大写、首字母小写、全小写后首字母大写)、口令复写后，将生成的口令更新概率，阶段号设为 7 加入 Q 中；

11: 完成猜测模块 (阶段号为 7)：该模块维护了 newpswlist 与概率阈值，当 newpswlist 长度小于目标猜测数时，将猜测口令加入 newpswlist；当 newpswlist 长度等于目标猜测数时，若 newpswlist 中概率最小的口令的概率小于当前猜测口令的概率时，替换为当前猜测口令，并更新概率阈值为 newpswlist 中的最小概率，否则抛弃当前猜测口令。替换时会检查重复，若重复则将口令概率更新为较大的概率，并相应调整概率阈值。

12: 当 Q 非空时，跳转至 4；

13: 返回 newpswlist；