**[编辑距离（Levenshtein Distance）](http://www.cnitblog.com/ictfly/archive/2005/12/27/5828.html)**

搞自然语言处理的应该不会对这个概念感到陌生，编辑距离就是用来计算从原串（s）转换到目标串(t)所需要的最少的插入，删除和替换的数目，在NLP中应用比较广泛，如一些评测方法中就用到了（wer,mWer等），同时也常用来计算你对原文本所作的改动数。  
编辑距离的算法是首先由俄国科学家Levenshtein提出的，故又叫Levenshtein Distance。  
Levenshtein distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t. For example,

* If s is "test" and t is "test", then LD(s,t) = 0, because no transformations are needed. The strings are already identical.
* If s is "test" and t is "tent", then LD(s,t) = 1, because one substitution (change "s" to "n") is sufficient to transform s into t.

The greater the Levenshtein distance, the more different the strings are.

Levenshtein distance is named after the Russian scientist Vladimir Levenshtein, who devised the algorithm in 1965. If you can't spell or pronounce Levenshtein, the metric is also sometimes called edit distance.

The Levenshtein distance algorithm has been used in:

* Spell checking
* Speech recognition
* DNA analysis
* Plagiarism detection

**The Algorithm**

**Steps**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns. |
| 2 | Initialize the first row to 0..n. Initialize the first column to 0..m. |
| 3 | Examine each character of s (i from 1 to n). |
| 4 | Examine each character of t (j from 1 to m). |
| 5 | If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1. |
| 6 | Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately above plus 1: d[i-1,j] + 1. b. The cell immediately to the left plus 1: d[i,j-1] + 1. c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost. |
| 7 | After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m]. |

**Example**

This section shows how the Levenshtein distance is computed when the source string is "GUMBO" and the target string is "GAMBOL".

**Steps 1 and 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | G | U | M | B | O |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| G | 1 |  |  |  |  |  |
| A | 2 |  |  |  |  |  |
| M | 3 |  |  |  |  |  |
| B | 4 |  |  |  |  |  |
| O | 5 |  |  |  |  |  |
| L | 6 |  |  |  |  |  |

**Steps 3 to 6 When i = 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | G | U | M | B | O |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| G | 1 | 0 |  |  |  |  |
| A | 2 | 1 |  |  |  |  |
| M | 3 | 2 |  |  |  |  |
| B | 4 | 3 |  |  |  |  |
| O | 5 | 4 |  |  |  |  |
| L | 6 | 5 |  |  |  |  |

**Steps 3 to 6 When i = 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | G | U | M | B | O |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| G | 1 | 0 | 1 |  |  |  |
| A | 2 | 1 | 1 |  |  |  |
| M | 3 | 2 | 2 |  |  |  |
| B | 4 | 3 | 3 |  |  |  |
| O | 5 | 4 | 4 |  |  |  |
| L | 6 | 5 | 5 |  |  |  |

**Steps 3 to 6 When i = 3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | G | U | M | B | O |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| G | 1 | 0 | 1 | 2 |  |  |
| A | 2 | 1 | 1 | 2 |  |  |
| M | 3 | 2 | 2 | 1 |  |  |
| B | 4 | 3 | 3 | 2 |  |  |
| O | 5 | 4 | 4 | 3 |  |  |
| L | 6 | 5 | 5 | 4 |  |  |

**Steps 3 to 6 When i = 4**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | G | U | M | B | O |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| G | 1 | 0 | 1 | 2 | 3 |  |
| A | 2 | 1 | 1 | 2 | 3 |  |
| M | 3 | 2 | 2 | 1 | 2 |  |
| B | 4 | 3 | 3 | 2 | 1 |  |
| O | 5 | 4 | 4 | 3 | 2 |  |
| L | 6 | 5 | 5 | 4 | 3 |  |

**Steps 3 to 6 When i = 5**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | G | U | M | B | O |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| G | 1 | 0 | 1 | 2 | 3 | 4 |
| A | 2 | 1 | 1 | 2 | 3 | 4 |
| M | 3 | 2 | 2 | 1 | 2 | 3 |
| B | 4 | 3 | 3 | 2 | 1 | 2 |
| O | 5 | 4 | 4 | 3 | 2 | 1 |
| L | 6 | 5 | 5 | 4 | 3 | 2 |

**Step 7**