Levenshtein-Distanz

Quelle: `https://mathe.zone/ausarbeitungen`
Version vom 25. Oktober 2020

Einleitung

In vielen Bereichen der Softwareentwicklung müssen vom Benutzer angegebene Daten ausgewertet werden. Bei vielen Daten wie etwa Geschlecht, Geburtsdatum oder Nationalität handelt es sich um endliche Mengen, aus welchen der Benutzer einfach das richtige Element auswählt. Andere Daten, wie beispielsweise die Sozialversicherungsnummer, enthalten Prüfziffern, wodurch fehlerhafte Eingaben entdeckt werden können.

Schwierig hingegen ist der Umgang mit Texten (sogenannten Zeichenketten), da hier einerseits das Fehlerpotential bei der Eingabe größer ist als bei Zahlen und genormten Daten und es andererseits für den Computer nicht möglich ist, diesen Fehler zu erkennen. Ein Beispiel: Wenn ein Mensch das falsch geschriebene Wort *Matemathik* liest, wird er trotz des Fehlers sofort wissen, was gemeint ist. Für einen Computer handelt es sich bei *Matemathik* und *Mathematik* jedoch um zwei völlig verschiedene Zeichenketten.

Für viele unserer alltäglichen Anwendungen ist es jedoch notwendig, ähnliche Zeichenketten zu erkennen. Beispiele dafür sind Suchmaschinen, Suchfunktionen bei Onlineversandhändlern, Rechtschreibkorrekturen oder Plagiatsprüfungen. Daher muss eine Methode entwickelt werden, mit welcher der Computer feststellen kann, wie ähnlich sich zwei Zeichenketten sind. Ein heute verbreitetes Instrument ist die Levenshtein-Distanz, welche nach dem russischen Mathematiker Wladimir Lewenstein (1935 – 2017) benannt ist, der sie 1965 entwickelte.

Erklärung

Gegeben sind zwei Zeichenketten X und Y, deren Längen mit $m = |X|$ und $n = |Y|$ definiert werden. Die einzelnen Zeichen von X und Y werden mit X_i bzw. Y_j bezeichnet, wobei natürlich $1 \leq i \leq m$ und $1 \leq j \leq n$ gilt.

Im nächsten Schritt wird eine Matrix A mit $m+1$ Zeilen und $n+1$ Spalten erstellt. Die einzelnen Elemente der Matrix werden mit $A_{i,j}$ bezeichnet, wobei jeweils bei 0 begonnen wird zu zählen. Der erste Index steht für die Zeile und der zweite für die Spalte. Es gilt nun $A_{0,0} = 0$, $A_{i,0} = i$ für $1 \leq i \leq m$ und $A_{0,j} = j$ für $1 \leq j \leq n$. Die Matrix hat daher vorerst folgende Gestalt:

$$A = \begin{pmatrix} 0 & 1 & 2 & \dots & n \\ 1 & * & * & \dots & * \\ 2 & * & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m & * & * & \dots & * \end{pmatrix}$$

Die weiteren Elemente der Matrix werden nun schrittweise (beispielsweise Zeile für Zeile) bestimmt. Bei der Reihenfolge der Elemente ist nur wichtig, dass alle Elemente links und oberhalb des zu bestimmenden Elements bereits bekannt sein müssen. Dabei wird für $A_{i,j}$ die kleinste der folgenden vier Zahlen verwendet, wobei die erste nur verwendet werden darf, wenn X_i und Y_j gleich sind und somit an dieser Stelle keine Änderung nötig ist:

- $A_{i-1,j-1}$ (keine Änderung nötig)
- $1 + A_{i-1,j-1}$ (ein Zeichen wird ersetzt)
- $1 + A_{i,j-1}$ (ein Zeichen wird eingefügt)
- $1 + A_{i-1,j}$ (ein Zeichen wird entfernt)

Die Levenshtein-Distanz der beiden Zeichenketten steht am Ende rechts unten in der Matrix, also auf Position $A_{m,n}$. Sie beschreibt die minimale Anzahl an Einfügungen, Löschungen und Ersetzungen, um aus der Zeichenkette X die Zeichenkette Y zu erhalten (oder umgekehrt).

Beispiele

Als Einstieg wird die Levenshtein-Distanz der beiden Zeichenketten $X = \text{TOR}$ und $Y = \text{TÜR}$ ermittelt. Diese beträgt offensichtlich 1, da hier nur der zweite Buchstabe ersetzt werden muss.

Es ist sinnvoll, die Spalten und Zeilen der Matrix mit dem entsprechenden Buchstaben zu beschriften, da man so leichter feststellen kann, ob sie gleich sind. Vertikal werden die Buchstaben der ersten Zeichenkette X aufgetragen und horizontal jene der zweiten Zeichenkette Y.

Nun muss für jedes unbekannte Matrix-Element das Minimum der vier oben genannten Terme bestimmt werden. Die letzten drei Zeilen wirken viel komplizierter, als sie in der Praxis tatsächlich sind: Es handelt sich dabei um jene Zahlen links, oberhalb bzw. links oberhalb des gesuchten Matrix-Elements. Man sucht lediglich die kleinste davon und addiert 1. Falls X_i und Y_j gleich sind, muss diese Zahl noch mit dem Element links oberhalb verglichen werden.

$$\begin{array}{c} \\ \text{T} \\ \text{O} \\ \text{R} \end{array} \begin{array}{c} \begin{array}{cccc} \text{T} & \text{Ü} & \text{R} \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 2 \\ 3 & 2 & 2 & 1 \end{pmatrix} \end{array}$$

Da rechts unten die Zahl 1 steht, beträgt die Levenshtein-Distanz der vorgegebenen Zeichenketten wie erwartet 1.

Als zweites Beispiel wird der Name CHRISTIAN mit der polnischen Variante KRYSTIAN verglichen. Hier ist die minimale Anzahl an nötigen Veränderungen nicht mehr so trivial wie zuvor. Man erhält folgende Matrix:

$$\begin{array}{c} \\ \text{C} \\ \text{H} \\ \text{R} \\ \text{I} \\ \text{S} \\ \text{T} \\ \text{I} \\ \text{A} \\ \text{N} \end{array} \begin{array}{c} \begin{array}{ccccccccc} \text{K} & \text{R} & \text{Y} & \text{S} & \text{T} & \text{I} & \text{A} & \text{N} \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 3 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 4 & 3 & 3 & 4 & 5 & 5 & 6 & 7 \\ 5 & 5 & 4 & 4 & 3 & 4 & 5 & 6 & 7 \\ 6 & 6 & 5 & 5 & 4 & 3 & 4 & 5 & 6 \\ 7 & 7 & 6 & 6 & 5 & 4 & 3 & 4 & 5 \\ 8 & 8 & 7 & 7 & 6 & 5 & 4 & 3 & 4 \\ 9 & 9 & 8 & 8 & 7 & 6 & 5 & 4 & 3 \end{pmatrix} \end{array}$$

Die Levenshtein-Distanz dieser Namen beträgt daher 3. Eine Möglichkeit wäre, das C durch ein K zu ersetzten, das H zu entfernen und das erste I durch ein Y zu ersetzen.

Mit dem folgenden Online-Tool kann die Levenshtein-Distanz berechnet werden:

`https://mathe.zone/tools/levenshtein-distanz`

Eigenschaften

Die Levenshtein-Distanz erfüllt folgende Eigenschaften:

- Sie ist genau dann 0, wenn die beiden Zeichenketten gleich sind.
- Ihr Wert entspricht mindestens dem Längenunterschied der beiden Zeichenketten.
- Der Maximalwert entspricht der Länge der längeren Zeichenkette.

Praxis-Tipps

In der Praxis sind eventuell folgende Vorgehensweisen sinnvoll:

- Falls es nur darum geht, eine Benutzereingabe mit einer Datenbank abzugleichen, so ist es sinnvoll, zuerst alle Zeichen der Eingabe und der Datenbank in Klein- oder Großbuchstaben umzuwandeln. Ansonsten vergrößern Unterschiede in der Groß- und Kleinschreibung die Levenshtein-Distanz.
- Bei Suchfunktionen kann es sinnvoll sein, anstelle der Levenshtein-Distanz die Levenshtein-Distanz pro Zeichen heranzuziehen. Denn bei einer Zeichenkette mit nur fünf Zeichen ist eine Levenshtein-Distanz von 3 relativ betrachtet weitaus größer als bei einer Zeichenkette mit 20 Zeichen.