

Szerkeztési távolság, Edit Distance, Levenshtein távolság

Kertész-Farkas Attila

Kivonat

Gyakori probléma különböző hosszúságú szekvenciák, sztringek, dokumentumok hasonlóságának ill. távolságának mérésére. Többféle módszerek léteznek erre a problémára, mindegyik egyfajta megközelítésből méri a távolságot. Itt az Szerkeztési távolságon belül a Levenshtein távolság kerül ismertetésre, amelyet 1965-ben definiált Vladimir Levenshtein. Ennek körülbelül az a lényege, hogy a két szekvenciában a lehető legjobban szeretnénk párosítani az azonos karaktereket egymással de úgy, hogy a párosításban a karakterek sorrendje ne változzon. Egy másik megfogalmazásban, a szerkeztési távolság azon minimális beszúrások, törlések és betűk cseréjének a száma, amellyel megkapható az egyik sztring a másiktól.

Ennek a dokumentumnak az elérhetősége www.inf.u-szeged.hu/~kfa/ed.zip. Itt a forrás is megtalálható, bárki módosíthatja terjesztheti.

1. Definíciók

Jelöljön két szekvenciát (sztringet) x, y egy rögzített ábécé felett, azaz a szekvenciák karakterei ugyanabból az ábécéből valók. x és y illesztésén, ill. alignmentjére a következő megszorításoknak kell egyszerre teljesülnie:

- Minden szimbólum x -ben és y -ban illeszteve kell hogy legyenek, és ugyanabban a sorrendben, mint ahogy x -ben és y -ben előfordulnak.
- Egy szimbólummal az egyik sztringből pontosan egy másikkal lehet párban a másik sztringből.
- Egy szimbólum párban állhat egy réssel (gap) jele: $_$
- Kettő rész nem állhat párban.

Példa. Legyen $x=aababab$ és $y=aabbababb$. E két szekvencia egy lehetséges illesztése:

aab_abab_

aabbababb

Illetve ugyanezen két szekvencia másik illesztése:

aa_bab_ab

aabbababb

Vagy egy harmadik:

--aababab

aabbababb

Ha egy betűpár (amely egymás alatt van) megegyezik akkor *illeszkedésről* beszélünk, ellenben *helyettesítésről* vagy *cseréről*. Ha egy betű pár egy réssel illeszkedik, akkor azt *beszúrásnak* vagy *törlésnek* nevezzük (Nézőpont kérdése, az egyik sztring szempontjából törlés, akkor a másik szempontjából beszúrás).

A kérdés, hogy melyik illesztés jobb? Ennek megválaszolására súlyfüggvényeket ill. büntetések fogunk definiálni. Büntetést szabunk ki rés beszúrásakor és helyettesítéskor is, illetve jutalmat adunk ha illeszkedést kapunk. A cél az illeszkedés értékek minimalizálása. Például, ha az illeszkedés jutalma 0 a helyettesítés és a rés beszúrásáé 1, akkor az első illesztés értéke 7, a másodiké 6, viszont a harmadiké 1.

Ennek kiszámítása dinamikus programozással történik egy $n+1 \times m+1$ méretű táblázat kitöltésével, ahol n és m a két sztring hossza.

Edit_distance(string x, string y)

```
1:  m = length(x);
2:  n = length(y);
3:  d = matrix(m+1,n+1);
4:  for i = 0 to m
5:      d[i,0] = i*p;
6:  for j = 0 to n
7:      d[0,j] = j*p;
8:  for j = 0 to m
9:      for j = 0 to n
10:         d[i,j] = min{d[i-1,j]+p, d[i,j-1]+p, d[i-1,j-1]+c(x[i],y[j]) };
11:  return d[m,n];
```

Itt a rés beszúrásának költségét p jelöli valamint a helyettesítés és az illesztés költségét $c(.,.)$ függvény adja. A minta illesztés értékét a dinamikus tábla jobb alsó sarkában lévő $d[m,n]$ érték adja.

1.1. Példa

Tekintsük az **aabab** és **abab** sztringeknek az illesztését a $p = 1$ és az alábbi helyettesítési függvényvel.

$$c(a, b) = \begin{cases} 0 & a = b \\ 1 & a \neq b \end{cases}$$

Ekkor a számítási tábla az 1. ábrán látható.

| | | A | A | B | A | B |
|---|---|-------|-------|-------|-------|-------|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| A | 1 | ↓ ↘ 0 | ↓ ↘ 1 | ↓ ↘ 2 | ↓ ↘ 3 | ↓ ↘ 4 |
| B | 2 | ↓ ↘ 1 | ↓ ↘ 1 | ↓ ↘ 1 | ↓ ↘ 2 | ↓ ↘ 3 |
| A | 3 | ↓ ↘ 2 | ↓ ↘ 1 | ↓ ↘ 2 | ↓ ↘ 1 | ↓ ↘ 2 |
| B | 4 | ↓ ↘ 3 | ↓ ↘ 2 | ↓ ↘ 1 | ↓ ↘ 2 | ↓ ↘ 1 |

1. ábra. Az **aabab** és **abab** sztringek illesztése.

Itt a példában a nyilak jelölik, hogy a minimális értékek mely cellákból jöhetnek. A tábla jobb alsó sarkában lévő 1-es érték jelöli a szerkesztési távolság értékét, amely jelen esetben 1.

A két sztring illesztését a jobb alsó $d[m, n]$ sarokból és az ebbe a cellába vezető úton a $d[0, 0]$ cellába visszafelé haladva olvashatjuk ki. Ha egy cellába kettő nyíl vezet, akkor mehetünk mindkét irányba, és ez eggyel növeli a helyes illesztések számát. Ha egy cellába három nyíl vezet, akkor ez kétfőve növeli a helyes illesztések számát. Tehát egy illesztési értékhez több illesztés is tartozhat. Az illesztés leolvasása a következőképpen történik. Az illesztés visszafelé halad. Ha egy cellába átlós irányban érteztünk, akkor az illesztésben *illesztés* vagy *helyettesítés* van és a cella oszlopához és sorához tartozó szimbólumokat leírjuk az eddigiek elé. Ha a cellába jobb oldalról érteztünk, akkor a felső sztringbe szűrünk egy részt, ha fentről, akkor a bal oldali sztringbe szűrünk egy részt.

Tehát a példa illesztései:

aabab

a_bab

és

aabab

_abab

Mindkettőhöz ugyanaz az illesztési (távolság) érték tartozik.

Megjegyzem, hogy teljesen más illesztések kaphatók más költségfüggvényekkel. Ha például a rés beszúrásának költsége sokkal magasabb a helyettesítés költségéhez képest, akkor a kapott illesztésben várhatóan kevesebb lesz a rés és több lesz a helyettesítés. Fordítva: ha a rés beszúrásának költsége alacsony a helyettesítéshez képest, akkor az algoritmus inkább rést szűr be, minthogy helyettesítést alkalmazzon.

2. Magyar irodalom

Tikk Domokos: Szövegbányászat, Typotex kiadó, 2007