Uvod u obradu prirodnog jezika

3.1. Minimalna udaljenost dva niza znakova (Minimum edit distance)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Koliko su slična dva niza znakova?

- Ispravak pravopisnih grešaka
  - Korisnik je unio
    - "žrafa"
  - Što bi bila ispravka
    - grafa
    - rafa
    - žirafa
- Računalna biologija
  - poravnaj nizove nukleotida

    AGGCTATCACCTGACCTCCAGGCCGATGCCC

    TAGCTATCACGACCGCGGTCGATTTGCCCGAC
  - poravnanje

    –AGG**CTATCAC**C**TGACC**T**C**C**A**G**G**C**C**GA**–**–TGCCC**–––

    T**AG**–**CTATCAC**–**–GACC**G**C**–**–GG**T**CGA**TT**TGCCC**GAC
- Koristi se kod strojnog prevođenja, ekstrakcije informacija, prepoznavanja govora…

Udaljenost nizova znakova

- Minimalna udaljenost između dva niza znakova
- je minimalni broj operacija
  - ubacivanja
  - brisanja
  - zamjene
- potrebnih da se jedan niz znakova transformira u drugi

Rezultat povratnog praćenja

- Dva niza znakova i njihovo poravnanje

```
T R A D * I C I J A
| | | | | | | | | |
* D E D U K C I J A
```

Minimalna udaljenost nizova znakova

- Dva niza znakova i njihovo poravnanje
  - b – brisanje
  - z – zamjena
  - u – ubacivanje

```
T R A D * I C I J A
| | | | | | | | | |
* D E D U K C I J A
b z z   u z
```

- Ako svaka operacija vrijedi 1 bod
  - onda je udaljenost 5
- ako zamjena iznosi 2 boda (Levenshtein)
  - onda je udaljenost 8

Poravnanje u računalnoj biologiji

- Za dani niz dušikovih baza

  AGGCTATCACCTGACCTCCAGGCCGATGCCC

  TAGCTATCACGACCGCGGTCGATTTGCCCGAC

- Poravnanje

  –AGG**CTATCAC**C**T****GACC**T**C**C**A****GG**C**CGA****--TGCCC**---

  T**AG****–CTATCAC****--GACC**G**C****--GG**T**CGA**TT**TGCCC**GAC

- Za dva dana niza, poravnaj svaki znak u drugi znak ili razmak

Druge upotrebe udaljenosti nizova znakova

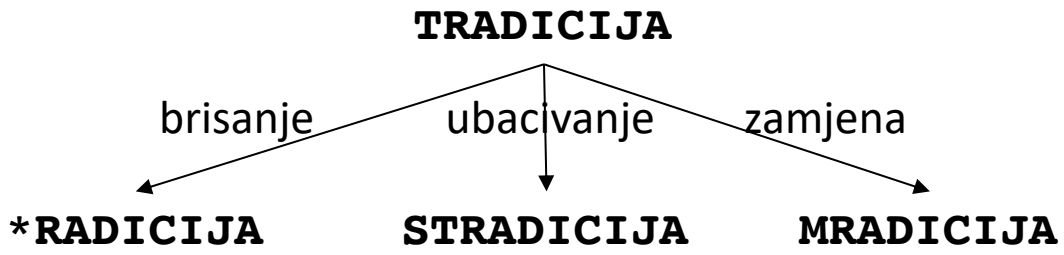
- Evaluacija strojnog prevođenja i prepoznavanja govora

```
Govornik je potvrdio da   je utakmica počela.
Govornik    kaže     kako je utakmica počela maloprije.
          B   Z        Z                        U
```

- Ekstrakcija imenovanih entiteta i njihove koreference
  - Podravka d.o.o. je najavila danas
  - Podravka je profitirala
  - Predsjednik Podravke Ante Antić je jučer najavio
  - za predsjednika uprave Podravke Antu Antića

Kako odrediti minimalnu udaljenost?

- Potraga za putanjom (nizom operacija) koja transformira početnu riječ u krajnju riječ
  - inicijalno stanje: riječ koja se transformira
  - operacije: ubaci, izbriši, zamjeni
  - ciljno stanje: riječ u koju se transformira
  - vrijednost putanje: ono što želimo minimizirati: broj operacija



Minimalna udaljenost kao traženje putanje

- Prostor svih putanja je ogroman!
  - ne možemo naivno pristupiti problemu pretrage
  - mnogo različitih putanja se generira od nekog stanja
    - ne moramo pratiti svaku od njih
    - samo najkraću putanju iz svih ponovno posjećenih stanja

Definicija minimalne udaljenosti

- Za dva niza znakova
  - S veličine m
  - T veličine n
- Definiramo $D[i, j]$
  - udaljenost između $S[1 \ldots i]$ i $T[1 \ldots j]$
    - npr. prvih i znakova od S i prvih j znakova od T
  - udaljenost između S i T je onda $D[m, n]$

Uvod u obradu prirodnog jezika

3.2. Izračun minimalne udaljenosti (Computing minimum edit distance)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Dinamičko programiranje za minimalnu udaljenost

- **Dinamičko programiranje**: tablični izračun od $D[m, n]$
- Rješavanje problema kombiniranjem rješenja podproblema
- Pristup odozdo prema dolje (Bottom-up)
  - Izračuna se $D[i, j]$ za male i, j
  - Izračuna se veliki $D[i, j]$ temeljem prethodno izračunatih manjih vrijednosti
  - npr. izračuna se $D[i, j]$ za sve
    - $i\ (0 < i < m)$ i
    - $j\ (0 < j < n)$

Definicija minimalne udaljenosti (Levensthein)

- Inicijalizacija:

  $D[i,0] = i$

  $D[0,j] = j$

- Relacija povratka:

  `za svaki` $i = 1 \ldots m$

  `za svaki` $j = 1 \ldots n$

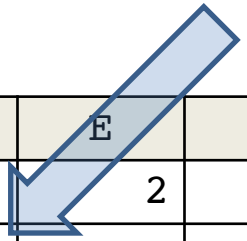
$$D[i,j] = min \begin{cases} D[i-1,j] + 1 \\ D[i,j-1] + 1 \\ D[i-1,j-1] + \begin{cases} 2: \text{ako } S[i] \neq T[j] \\ 0: \text{ako } S[i] = T[j] \end{cases} \end{cases}$$

- Zaustavljanje:

  $D[m,n]$ je udaljenost

Tablica minimalne udaljenosti

$$D[i,j] = min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 2: \text{ako } S[i] \neq T[j] \\ 0: \text{ako } S[i] = T[j] \end{cases} \end{cases}$$



S\T	#	D	E	D	U	K	C	I	J	A
#	0	1	2	3	4	5	6	7	8	9
T	1									
R	2									
A	3									
D	4									
I	5									
C	6									
I	7									
J	8									
A	9									

Tablica minimalne udaljenosti

$$D[i,j] = min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 2: \text{ako } S[i] \neq T[j] \\ 0: \text{ako } S[i] = T[j] \end{cases} \end{cases}$$

S\T	#	D	E	D	U	K	C	I	J	A
#	0	1	2	3	4	5	6	7	8	9
T	1	2	3	4	5	6	7	8	9	10
R	2	3	4	5	6	7	8	9	10	11
A	3	4	5	6	7	8	9	10	11	10
D	4	3	4	5	6	7	8	9	10	11
I	5	4	5	6	7	8	9	8	9	10
C	6	5	6	7	8	9	8	9	10	11
I	7	6	7	8	9	10	9	8	9	10
J	8	7	8	9	10	11	10	9	8	9
A	9	8	9	10	11	12	11	10	9	8

Uvod u obradu prirodnog jezika

3.3. Povratno praćenje za izračun poravnavanja (Backtrace in computing alignment)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Izračun poravnavanja

- Minimalna udaljenost nije dovoljna
- Često je potrebno **uskladiti** svaki znak od dva niza znakova
- Ovo se vrši korištenjem **povratnim praćenjem**
- Svakim ulaskom u ćeliju tablice trebamo znati odakle smo došli
- Kada dođemo do kraja
  - pratimo trag od gornjeg desnog kuta kako bi pročitali poravnanje

Dodavanje povratnog praćenja

- Inicijalizacija:
  $D[i,0] = i$
  $D[0,j] = j$
- Relacija povratka:

za svaki $i = 1 \ldots m$

  za svaki $j = 1 \ldots n$

$$D[i,j] = min \begin{cases} D[i-1,j] + 1 & \text{brisanje} \\ D[i,j-1] + 1 & \text{ubacivanje} \\ D[i-1,j-1] + \begin{cases} 2: \text{ako } S[i] \neq T[j] \\ 0: \text{ako } S[i] = T[j] \end{cases} & \text{zamjena} \end{cases}$$

$$Ptr[i.j] = \begin{cases} \uparrow & \text{brisanje} \\ \leftarrow & \text{ubacivanje} \\ \nwarrow & \text{zamjena} \end{cases}$$

- Zaustavljanje:
  $D[m,n]$ je udaljenost

Tablica minimalne udaljenosti

$$D[i,j] = min \begin{cases} D[i-1,j]+1 & \text{brisanje} \\ D[i,j-1]+1 & \text{ubacivanje} \\ D[i-1,j-1] + \begin{cases} 2: \text{ako } S[i] \neq T[j] \\ 0: \text{ako } S[i] = T[j] \end{cases} & \text{zamjena} \end{cases}$$

S\T	#	D	E	D	U	K	C	I	J	A
#	0	1	2	3	4	5	6	7	8	9
T	1	2	3	4	5	6	7	8	9	10
R	2	3	4	5	6	7	8	9	10	11
A	3	4	5	6	7	8	9	10	11	10
D	4	3	4	5	6	7	8	9	10	11
I	5	4	5	6	7	8	9	8	9	10
C	6	5	6	7	8	9	8	9	10	11
I	7	6	7	8	9	10	9	8	9	10
J	8	7	8	9	10	11	10	9	8	9
A	9	8	9	10	11	12	11	10	9	8

Tablica minimalne udaljenosti

$$Ptr[i.j] = \begin{cases} \uparrow \\ \leftarrow \\ \nwarrow \end{cases}$$

- ↑ brisanje
- ← ubacivanje
- ↖ zamjena

S\T	#	D	E	D	U	K	C	I	J	A
#	0	← 1	← 2	← 3	← 4	← 5	← 6	← 7	← 8	← 9
T	↑ 1	↖↑← 2	↖↑← 3	↖↑← 4	↖↑← 5	↖↑← 6	↖↑← 7	↖↑← 8	↖↑← 9	↖↑← 10
R	↑ 2	↖↑← 3	↖↑← 4	↖↑← 5	↖↑← 6	↖↑← 7	↖↑← 8	↖↑← 9	↖↑← 10	↖↑← 11
A	↑ 3	↖↑← 4	↖↑← 5	↖↑← 6	↖↑← 7	↖↑← 8	↖↑← 9	↖↑← 10	↖↑← 11	↖ 10
D	↑ 4	↖ 3	↖← 4	↖← 5	← 6	← 7	← 8	← 9	← 10	↑← 11
I	↑ 5	↑ 4	↖↑← 5	↖↑← 6	↖↑← 7	↖↑← 8	↖↑← 9	↖ 8	← 9	← 10
C	↑ 6	↑ 5	↖↑← 6	↖↑← 7	↖↑← 8	↖↑← 9	↖ 8	↑← 9	↖↑← 10	↖↑← 11
I	↑ 7	↑ 6	↖↑← 7	↖↑← 8	↖↑← 9	↖↑← 10	↑ 9	↖ 8	← 9	← 10
J	↑ 8	↑ 7	↖↑← 8	↖↑← 9	↖↑← 10	↖↑← 11	↑ 10	↑ 9	↖ 8	← 9
A	↑ 9	↑ 8	↖↑← 9	↖↑← 10	↖↑← 11	↖↑← 12	↑ 11	↑ 10	↑ 9	↖ 8

Tablica minimalne udaljenosti

```
TRAD*ICIJA
*DEDUKCIJA
bzz uz
```

S\T	#	D	E	D	U	K	C	I	J	A
#	0	← 1	← 2	← 3	← 4	← 5	← 6	← 7	← 8	← 9
T	↑ 1	↖↑ ← 2	↖↑ ← 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10
R	↑ 2	↖↑ ← 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11
A	↑ 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↖ 10
D	↑ 4	↖ 3	↖ ← 4	↖ ← 5	← 6	← 7	← 8	← 9	← 10	↑ ← 11
I	↑ 5	↑ 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖ 8	← 9	← 10
C	↑ 6	↑ 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖ 8	↑ ← 9	↖↑ ← 10	↖↑ ← 11
I	↑ 7	↑ 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↑ 9	↖ 8	← 9	← 10
J	↑ 8	↑ 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↑ 10	↑ 9	↖ 8	← 9
A	↑ 9	↑ 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↖↑ ← 12	↑ 11	↑ 10	↑ 9	↖ 8

Tablica minimalne udaljenosti

```
TRADI**CIJA
*DED*UKCIJA
bzz buu
```

S\T	#	D	E	D	U	K	C	I	J	A
#	0	← 1	← 2	← 3	← 4	← 5	← 6	← 7	← 8	← 9
T	↑ 1	↖↑ ← 2	↖↑ ← 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10
R	↑ 2	↖↑ ← 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11
A	↑ 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↖ 10
D	↑ 4	↖ 3	↖ ← 4	↖ ← 5	← 6	← 7	← 8	← 9	← 10	↑ ← 11
I	↑ 5	↑ 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖ 8	← 9	← 10
C	↑ 6	↑ 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖ 8	↑ ← 9	↖↑ ← 10	↖↑ ← 11
I	↑ 7	↑ 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↑ 9	↖ 8	← 9	← 10
J	↑ 8	↑ 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↑ 10	↑ 9	↖ 8	← 9
A	↑ 9	↑ 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↖↑ ← 12	↑ 11	↑ 10	↑ 9	↖ 8

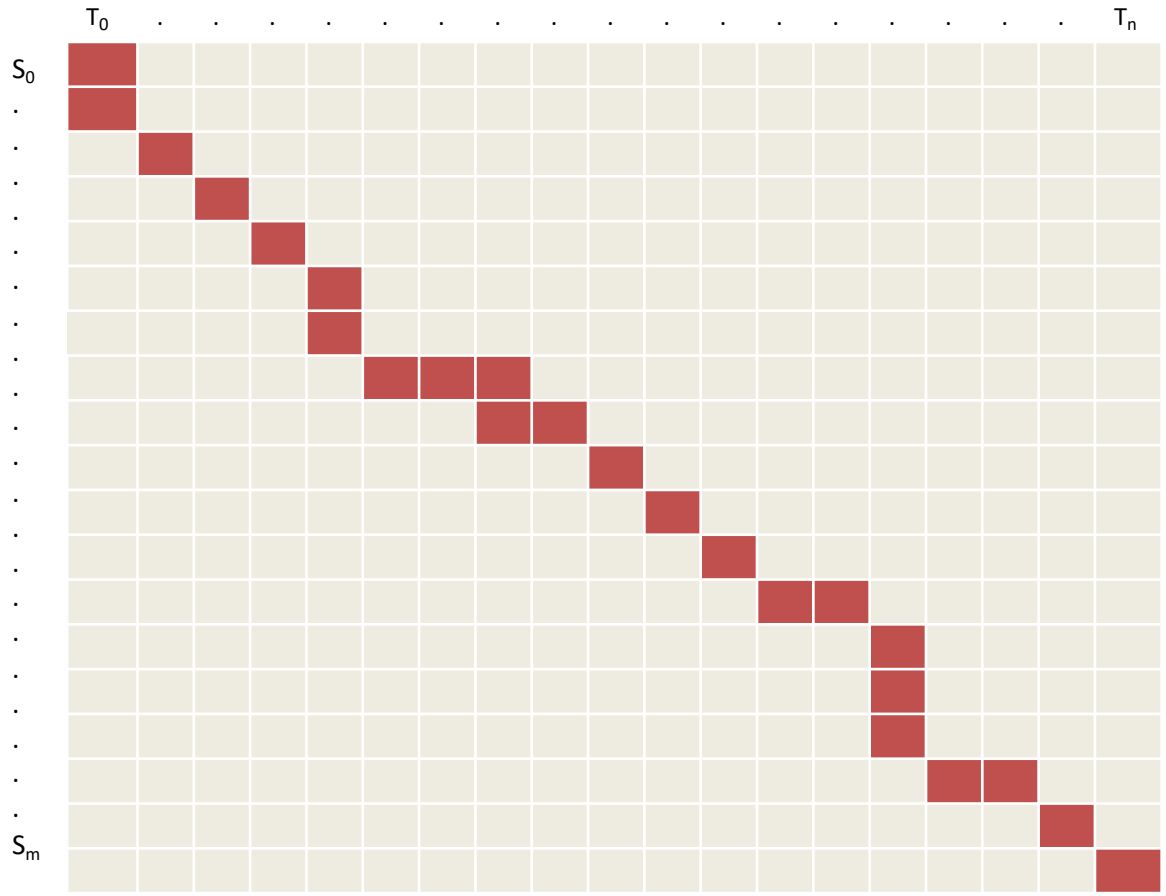
Tablica minimalne udaljenosti

```
TRADI****CIJA
***D*EDUKCIJA
bbb buuuu
```

S\T	#	D	E	D	U	K	C	I	J	A
#	0	← 1	← 2	← 3	← 4	← 5	← 6	← 7	← 8	← 9
T	↑ 1	↖↑ ← 2	↖↑ ← 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10
R	↑ 2	↖↑ ← 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11
A	↑ 3	↖↑ ← 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↖ 10
D	↑ 4	↖ 3	↖ ← 4	↖ ← 5	← 6	← 7	← 8	← 9	← 10	↑ ← 11
I	↑ 5	↑ 4	↖↑ ← 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖ 8	← 9	← 10
C	↑ 6	↑ 5	↖↑ ← 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖ 8	↑ ← 9	↖↑ ← 10	↖↑ ← 11
I	↑ 7	↑ 6	↖↑ ← 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↑ 9	↖ 8	← 9	← 10
J	↑ 8	↑ 7	↖↑ ← 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↑ 10	↑ 9	↖ 8	← 9
A	↑ 9	↑ 8	↖↑ ← 9	↖↑ ← 10	↖↑ ← 11	↖↑ ← 12	↑ 11	↑ 10	↑ 9	↖ 8

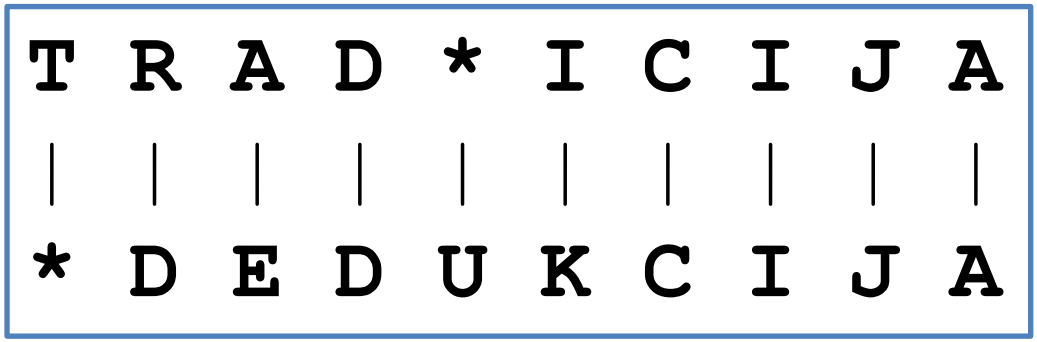
Matrica udaljenosti

- Svaka ne opadajuća putanja od (m,n) do (0,0) odgovara poravnanju dvaju niza znakova
- Optimalno poravnanje se sastoji od optimalnih podporavnanja



Minimalna udaljenost nizova znakova

- Dva niza znakova i njihovo poravnanje



Performanse

- vrijeme O(mn)
- prostor O(mn)
- povratno praćenje O(m+n)

Uvod u obradu prirodnog jezika

3.4. Težinska minimalna udaljenost (Weighted edit distance)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

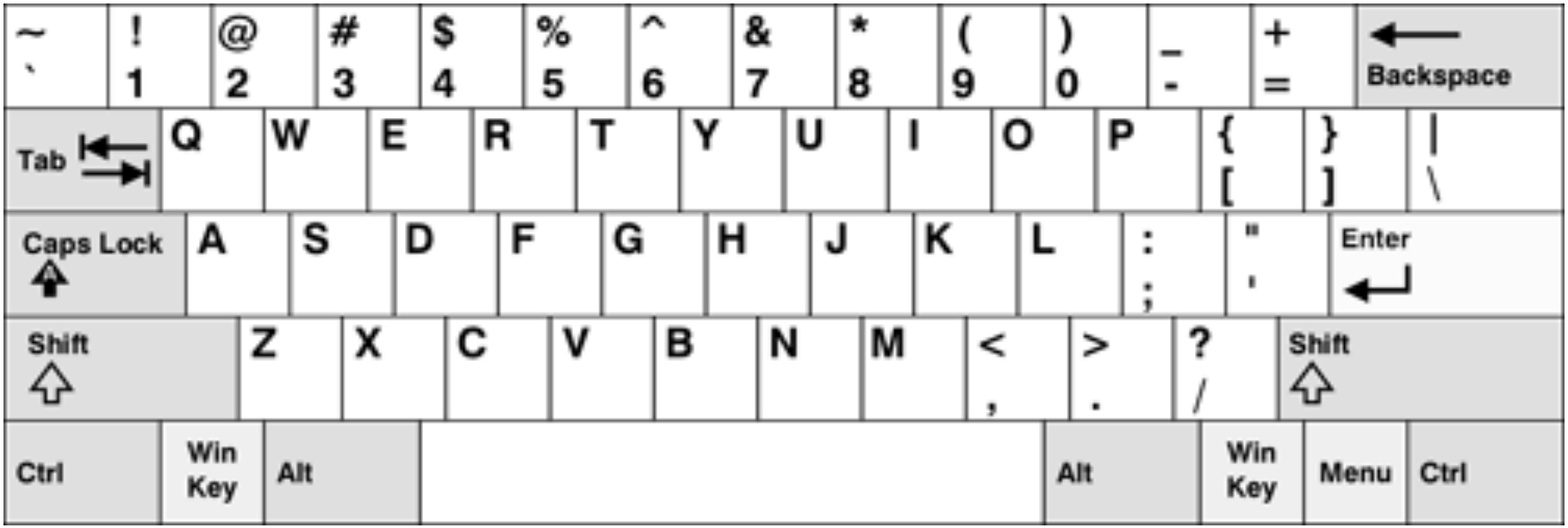
Zašto dodati težinski faktor?

- Ispravak pravopisnih grešaka
  - neka slova imaju veću vjerojatnost krivog unosa u odnosu na druga
- Biologija
  - određena brisanja i umetanja su vjerojatnija od drugih

Matrica konfuzije za pravopisne pogreške

zamjena[X,Y] = supstitucija od X(pogrešno) s Y(točno)

		Y(točno)																									
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
	b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
	c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
	d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
	e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	14	0	1	0	18	0
	f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
	g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
	h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
	i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
	j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
	k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
	l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
	m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
	n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
	o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
	p	0	11	1	2	0	6	5	0	2	0	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
	q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
	s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
	t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
	u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
	v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
	w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
	x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
	y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
	z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0


~ `
! 1
@ 2
# 3
$ 4
% 5
^ 6
& 7
* 8
(9
) 0
_ -
+ =
Backspace
Tab
Q W E R T Y U I O P
{ [
}]
| \
Caps Lock
A S D F G H J K L
: ;
" '
Enter
Shift
Z X C V B N M
< ,
> .
? /
Shift
Ctrl
Win Key
Alt
Alt
Win Key
Menu
Ctrl

Težinska minimalna udaljenost

- Inicijalizacija:

$$D[0,0] = 0$$
$$D[i,0] = D[i,0] + \text{brisanje}(S[i]); \quad 1 \le i \le m$$
$$D[0,j] = D[0,j] + \text{ubacivanje}(T[j]); \quad 1 \le j \le n$$

- Relacija povratka:

za svaki $i = 1 \ldots m$

za svaki $j = 1 \ldots n$

$$D[i,j] = min \begin{cases} D[i-1,j] + \text{brisanje}(S[i]) \\ D[i,j-1] + \text{ubacivanje}(T[j]) \\ D[i-1,j-1] + \text{zamjena}(S[i],T[j]) \end{cases}$$

- Zaustavljanje:

$D[m,n]$ je udaljenost