Razdalje urejanja 2. del

Jarova in Jaro-Winklerjeva razdalja

Mladen Borovič, Jani Dugonik

Uvod

V prejšnjem delu smo podrobneje predstavili dve razdalji urejanja – Hammingovo in Levenshteinovo razdaljo. Na primerih smo predstavili glavne operacije: vstavljanje (*ang. insertion*), brisanje (*ang. deletion*) in zamenjava (*ang. substitution*). Omenili smo tudi posebno obliko zamenjave, ki jo imenujemo transpozicija (*ang. transposition*) in poudarili razliko med zamenjavo in transpozicijo. Pri navadni zamenjavi gre za zamenjavo znaka s poljubnim znakom, tudi takšnim, ki se ne pojavi v nizu, pri transpoziciji pa smo omejeni na zamenjavo znakov, ki se nahajajo v samem nizu. Ker niti Hammingova, niti Levenshteinova razdalja ne dovoljujeta operacije transpozicije, se bomo v tem prispevku omejili na dve razdalji urejanja, ki dovoljujeta transpozicijo; to sta Jarova in Jaro-Winklerjeva razdalja.

Jarova in Jaro-Winklerjeva razdalja se na področju procesiranja naravnega jezika, podobno kot vse ostale razdalje urejanja, uporabljata za primerjavo dveh nizov in ugotavljanje podobnosti med njima. Ker sta razdalji nastali v razmaku enega leta (Jarova je nastala leta 1989, Jaro-Winklerjeva pa leta 1990), sta v literaturi večkrat definirani kot ena razdalja – Jaro-Winklerjeva razdalja. V resnici Jarova razdalja predstavlja osnovno razdaljo, ki upošteva operacijo transpozicije, Jaro-Winklerjeva razdalja pa je razširitev, ki k delovanju doda višjo oceno podobnosti, če se vhodna niza ujemata v začetnih znakih. Za izračun Jaro-Winklerjeve razdalje torej potrebujemo Jarovo razdaljo. Da bi bolje razumeli uporabo transpozicije, si v nadaljevanju podrobneje poglejmo njuno delovanje na primerih.

Jarova razdalja

Leta 1989 je Matthew Jaro, ameriški znanstvenik na področju računalništva, predstavil del rešitve za združevanje podatkov pridobljenih pri popisu prebivalstva. Sodeloval je pri razvoju sistema, ki je združeval podatke iz več podatkovnih baz na podlagi imen in priimkov. Pri tem je bila potrebna posebna metrika podobnosti za krajša besedila, ki je danes znana kot Jarova razdalja [1] in se uvršča med razdalje urejanja, najpogosteje pa se uporablja za izračun podobnosti med dvema nizoma pri preverjanju črkovanja [2]. Ta razdalja za razliko od večine ostalih

a	**r**	e	z	u	l	t	a	t
b	**k**	**o**	**n**	**z**	u	l	a	t

SLIKA 1.
Premikajoče okno za znak »r«.

a	r	**e**	z	u	l	t	a	t
b	**k**	**o**	**n**	**z**	u	l	a	t

SLIKA 2.
Premikajoče okno za znak »e«

a	r	e	z	u	**l**	t	a	t
b	**k**	**o**	**n**	**z**	u	l	a	t

SLIKA 3.
Premikajoče okno za znak »l«

a	r	e	z	u	l	t	a	**t**
b	**k**	**o**	**n**	**z**	u	l	a	t

SLIKA 4.
Premikajoče okno za znak »t«.

razdalj urejanja primerja ujemajoče znake znotraj premikajočega okna, nato pa nad ujemajočimi znaki išče zamenjave znakov oziroma transpozicije.

Prvi korak izračuna Jarove razdalje je določitev velikosti premikajočega okna. Velikost premikajočega okna W pri uporabi nizov a in b definiramo z enačbama:

- $$c = \left\lfloor \frac{\max(|a|, |b|)}{2} \right\rfloor - 1 \qquad (1)$$

- $$W = 2 \cdot c + 1, \qquad (2)$$

kjer sta $|a|$ in $|b|$ dolžini nizov a in b. Vrednost parametra c nam določa, koliko znakov bomo preverjali v drugem nizu levo in desno od istoležnega znaka.

Slike 1–4 prikazujejo, katere znake vključimo v iskanje ujemanja na različnih pozicijah za primer nizov a = rezultat in b = konzulat pri izračunu Jarove razdalje (velikost okna je $W = 7$). S temnejšo modro barvo je označen trenutni znak v nizu a, ki ga primerjamo z znaki znotraj premikajočega okna v nizu b; ti so označeni s svetlejšo modro barvo.

Premikajoče okno premikamo po nizih a in b od začetka do konca in beležimo število ujemajočih znakov. Pri tem znake, ki smo jih v nizu b že zabeležili kot ujemajoče, ne beležimo ponovno. Naslednji korak je ugotavljanje transpozicij. Ujemajoče znake iz nizov a in b zapišemo po originalnem vrstnem redu, kot se pojavljajo v nizih a in b. Nato preštejemo, koliko istoležnih znakov ni ujemajočih. To število delimo z 2 in dobimo število transpozicij. V zadnjem koraku določimo vrednost Jarove razdalje po enačbi:

- $$\mathrm{sim}_j(a, b) = \begin{cases} 0, & \text{če je } m = 0 \\ \frac{1}{3} \cdot \left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m} \right), & \text{sicer} \end{cases} \qquad (3)$$

kjer je m število ujemanj, t število transpozicij, $|a|$ in $|b|$ pa dolžini nizov a in b. Vidimo, da bo Jarova razdalja enaka 0, če niza nimata ujemajočih znakov. Na zgledu si poglejmo delovanje Jarove razdalje.

Zgled

Imejmo vhodna niza a = telefon in b = lepota. Najprej določimo daljši niz. Ker je $|a| = 7$ in $|b| = 6$, je niz a daljši. Naslednji korak je določitev premikajočega okna:

- $$c = \left\lfloor \frac{\max(|a|, |b|)}{2} \right\rfloor - 1 = \left\lfloor \frac{\max(7, 6)}{2} \right\rfloor - 1$$
$$= \left\lfloor \frac{7}{2} \right\rfloor - 1 = 3 - 1 = 2$$

- $W = 2 \cdot c + 1 = 2 \cdot 2 + 1 = \underline{\mathbf{5}}$.

Izračunali smo, da bo pri poravnavi nizov a in b naše premikajoče okno obsegalo dva znaka levo in desno od istoležnega znaka. Od tod sledi, da bo velikost premikajočega okna enaka 5. Sledi prehod skozi niz a in iskanje ujemajočih znakov znotraj premikajočega okna v nizu b. Na spodnjih slikah bomo na levi strani spremljali prehod skozi niz a in pozicijo premikajočega okna, na desni strani pa vsebino seznama ujemajočih znakov M, ki je na začetku prazna množica. Ujemanja bodo označena z zeleno barvo.

Prvi znak iz niza a, ki ga preverjamo za ujemanje, je znak »t«. V nizu b s premikajočim oknom prekrijemo znake »l«, »e« in »p« ter ugotovimo, da nimamo ujemanja. Seznam ujemajočih znakov M je torej prazna množica. Naslednji znak iz niza a, ki ga preverjamo za ujemanje, je znak »e«. V nizu b s premikajočim oknom prekrijemo znake »l«, »e«, »p« in »o« ter ugotovimo, da imamo ujemanje. V seznam ujemajočih znakov M dodamo znak »e«.

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

$M = \emptyset$

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

$M = \{e\}$

SLIKA 5.

Iskanje ujemanja za znaka »t« in »e«.

Naslednji znak iz niza a, ki ga preverjamo za ujemanje, je znak »l«. V nizu b s premikajočim oknom prekrijemo znake »l«, »e«, »p«, »o« in »t« ter ugotovimo, da imamo ujemanje. V seznam ujemajočih

znakov M dodamo znak »l«. Naslednji znak iz niza a, ki ga preverjamo za ujemanje, je ponovno znak »e«. V nizu b s premikajočim oknom prekrijemo znake »e«, »p«, »o«, »t« in »a« ter ugotovimo, da imamo ujemanje. Znak »e« iz niza b smo že označili kot ujemajočega v drugem koraku prehoda, zato ga ne dodamo v seznam ujemajočih znakov M.

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

$M = \{e, l\}$

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

$M = \{e, l\}$

SLIKA 6.
Iskanje ujemanja za znaka »l« in »e«.

Naslednji znak iz niza a, ki ga preverjamo za ujemanje, je znak »f«. V nizu b s premikajočim oknom prekrijemo znake »p«, »o«, »t« in »a« ter ugotovimo, da nimamo ujemanja. Seznam ujemajočih znakov M se ne spremeni. Naslednji znak iz niza a, ki ga preverjamo za ujemanje, je znak »o«. V nizu b s premikajočim oknom prekrijemo znake »o«, »t« in »a« ter ugotovimo, da imamo ujemanje. V seznam ujemajočih znakov M dodamo znak »o«.

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

$M = \{e, l\}$

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

$M = \{e, l, o\}$

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

$M = \{e, l, o\}$

SLIKA 7.
Iskanje ujemanja za znake »f«, »o« in »n«.

Zadnji znak iz niza a, ki ga preverjamo za ujemanje, je znak »n«. V nizu b s premikajočim oknom prekrijemo znaka »t« in »a« ter ugotovimo, da nimamo ujemanja. Seznam ujemajočih znakov M se ne spremeni.

Po končanem prehodu skozi niz a preštejemo število ujemajočih znakov s seznama M. Ugotovimo, da imamo tri ujemajoče znake in dobimo vrednost $m = 3$. Sledi ugotavljanje transpozicij. Za niza a in b po vrstnem redu pojavitve v nizu zapišemo pojavitev znakov s seznama ujemajočih znakov M. Ujemajoči znaki so označeni z oranžno barvo.

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

→

a	e	l	o
b	l	e	o

SLIKA 8.
Ugotavljanje transpozicij za niza a = telefon in b = lepota.

Iz dobljenega zapisa preštejemo, kolikokrat se na istoležnih mestih zgodi neujemanje, in to število delimo z 2. Dobljena vrednost predstavlja število transpozicij t. Ugotovimo, da imamo dve neujemanji na istoležnih mestih. Gre za neujemanji v znakih »e« in »l«. Neujemanji sta označeni z vijolično barvo. Gre torej za transpozicijo teh dveh znakov, saj opazimo, da se znaka v nizih a in b pojavita v drugačnem vrstnem redu. Iz tega izračunamo $t = \frac{2}{2} = 1$. Zdaj, ko imamo število ujemanj ($m = 3$) in število transpozicij ($t = 1$), lahko izračunamo Jarovo razdaljo po enačbi 3:

- $$\mathrm{sim}_j(a,b) = \frac{1}{3} \cdot \left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m} \right)$$
$$= \frac{1}{3} \cdot \left(\frac{3}{7} + \frac{3}{6} + \frac{3-1}{3} \right)$$
$$\mathrm{sim}_j(a,b) = \frac{1}{3} \cdot (1{,}595) = \underline{\mathbf{0{,}532}}.$$

Izračunali smo, da je Jarova razdalja med nizoma a = telefon in b = lepota enaka 0,532, kar interpretiramo kot 53,2% podobnost med nizoma a in b.

Jaro-Winklerjeva razdalja

Kmalu zatem, ko se je pojavila Jarova razdalja, se je začela tudi širše uporabljati v ameriških uradih za popis prebivalstva. Leta 1990 je bil William Winkler, ameriški znanstvenik na področju računalništva, zaposlen na enem izmed državnih uradov za popis prebivalstva. Pri svojem delu je prišel do spoznanja, da pri popisovanju osebnih podatkov prebivalstva največkrat prihaja do napak v črkovanju. Ugotovil je tudi, da se pri teh podatkih ujemanja zgodijo prav na začetku nizov oziroma da so si največkrat nizi zelo podobni na začetku, do razlik pa prihaja proti koncu nizov. Iz teh dognanj je Jarovo razdaljo razširil z dvema parametroma, ki dajeta večjo utež ujemanju na začetku nizov, novo razdaljo pa je poimenoval Jaro-Winklerjeva razdalja [5]. Prvi parameter je parameter ℓ, ki predstavlja dolžino predpone na začetku niza in ima v izvorni različici Jaro-Winklerjeve vrednost največ 4 ($0 \leq \ell \leq 4$). Drugi parameter je parameter p, ki je skalirni faktor za spremembo dodatka h končni vrednosti razdalje, ki nastopi zaradi ujemanja v začetnih ℓ znakih. Pri tem naj vrednost p za izvorno različico Jaro-Winklerjeve ne bi presegla 0,25, saj se v nasprotnem primeru lahko zgodi, da je vrednost razdalje več kot 1. To seveda ni zaželeno, saj ne želimo imeti vrednosti razdalje na intervalu $[0, 1]$. Winkler je pri svojem delu uporabljal vrednost $p = 0{,}1$, kar je tudi danes privzeta vrednost za ta parameter.

Sam izračun Jaro-Winklerjeve razdalje se ne razlikuje preveč od izračuna Jarove razdalje. Najprej med nizoma a in b izračunamo Jarovo razdaljo $\mathrm{sim}_j(a, b)$, nato pa uporabimo parametra ℓ in p. Vrednost Jaro-Winklerje razdalje $\mathrm{sim}_{jw}(a, b)$ izračunamo z enačbo

- $$\mathrm{sim}_{jw}(a, b) = \mathrm{sim}_j(a, b) + \ell \cdot p \cdot (1 - \mathrm{sim}_j(a, b)), \quad (4)$$

kjer je $\mathrm{sim}_j(a, b)$ Jarova razdalja med nizoma a in b, ℓ je dolžina predpone na začetku niza, p pa skalirni faktor za spremembo dodatka h končni vrednosti razdalje. Na zgledu si poglejmo izračun Jaro-Winklerjeve razdalje.

Zgled

Imejmo niza a = telefon in b = lepota. Najprej izračunamo Jarovo razdaljo $\mathrm{sim}_j(a, b)$. To smo za niza a = telefon in b = lepota storili že v zgledu za Jarovo razdaljo in dobili rezultat $\mathrm{sim}_j(a, b) = 0{,}532$. Nato izračunamo vrednost Jaro-Winklerjeve razdalje, za katero moramo definirati vrednosti parametrov ℓ in p. Kot predpono bomo upoštevali prve tri znake v nizu ($\ell = 3$), za skalirni faktor pa bomo uporabili vrednost iz izvorne različice Jaro-Winklerjeve razdalje ($p = 0{,}1$). Sledi izračun Jaro-Winklerjeve razdalje po enačbi 4:

- $\mathrm{sim}_{jw}(a, b) = \mathrm{sim}_j(a, b) + \ell \cdot p \cdot (1 - \mathrm{sim}_j(a, b))$
  $\mathrm{sim}_{jw}(a, b) = 0{,}532 + (3 \cdot 0{,}1 \cdot (1 - 0{,}532)$
  $\mathrm{sim}_{jw}(a, b) = 0{,}532 + 0{,}140 = \underline{\mathbf{0{,}672}}$

a	t	e	l	e	f	o	n
b	l	e	p	o	t	a	

SLIKA 9.
Ujemanje v prvih treh znakih nizov a in b. Območje predpone ($\ell = 3$) je označeno s svetlo modro barvo, ujemanja znotraj predpone pa s temno modro barvo.

Izračunali smo, da je Jaro-Winklerjeva razdalja med nizoma a = telefon in b = lepota enaka 0,672, kar interpretiramo kot 67,2 % podobnost. Če primerjamo rezultat z vrednostjo Jarove razdalje nad nizoma a in b, opazimo, da smo z Jaro-Winklerjevo razdaljo povečali podobnost med nizoma. To je zaradi dodatka h končni vrednosti razdalje zaradi ujemanja v začetnih ℓ znakih nizov a in b. Ker smo uporabili parameter $\ell = 3$, smo dodatno utežili ujemanje v prvih treh znakih nizov a in b, po skaliranju s faktorjem $p = 0{,}1$ pa smo izračunali, da je dodatek h končni vrednosti razdalje enak 0,140. Če preverimo, zaradi katerih znakov smo povečali vrednost Jarove razdalje, ugotovimo, da sta za to zaslužna znaka »e« in »l«, ki sta na sliki 9 označena s temno modro barvo.

Zaključek

V tem prispevku smo podrobneje predstavili delovanje Jarove in Jaro-Winklerjeve razdalje urejanja. Ob primerih smo spoznali delovanje operacije transpozicije znakov, ki smo jih v prvem delu prispevka izpustili. Primera implementacije Jarove in Jaro-Winklerjeve razdalje v programskem jeziku

Python sta na voljo na javno dostopnem repozitoriju GitHub [3].

Jarova in Jaro-Winklerjeva razdalja se danes ponavadi ne ločita, saj v praksi srečujemo večinoma Jaro-Winklerjevo implementacijo. Obe sta zelo koristni pri popravljanju črkovanja in seveda pri združevanju podatkov v obliki krajšega besedila iz različnih virov, recimo imen in priimkov. Kot smo lahko spoznali v obeh prispevkih na temo razdalj urejanja, imamo na področju procesiranja naravnega jezika na izbiro kar nekaj različnih opcij, kadar govorimo o iskanju podobnosti med besedili. Kljub temu za splošno ugotavljanje podobnosti med besedili prednjači uporaba Levenshteinove razdalje, ostale razdalje urejanja pa uporabljamo v posebnih primerih. Na primeru Jaro-Winklerjeve razdalje smo lahko zelo nazorno videli smiselnost prilagoditve razdalje urejanja za posebni primer. Velja omeniti, da obstaja več različic Jaro-Winklerjeve razdalje v smislu vpeljave novih ali drugačnih parametrov [4]. To nakazuje na dejstvo, da se področje procesiranja naravnega jezika neprestano razvija in po potrebi prilagaja na različne oblike podatkov, čeprav je na voljo že kar obsežna množica metod za ugotavljanje podobnosti med besedili.

Literatura

[1] M. A. Jaro, *Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida*, Journal of the American Statistical Association **84.406** (1989), 414–420, doi: 10.1080/01621459.1989.10478785.

[2] M. A. Jaro, *Probabilistic linkage of large public health data files*, Statistics in medicine **14** (1995), 491–498.

[3] *Procesiranje naravnega jezika – GitHub*, dostopno na `github.com/procesiranje-naravnega-jezika/example-code/tree/main/1c%5C%20-%5C%20Razdalje%5C%20urejanja%5C%202.%5C%20del%5C%20-%5C%20Jarova%5C%20in%5C%20Jaro-Winklerjeva%5C%20razdalja`, ogled 5. 5. 2022.

[4] W. E. Winkler, *Overview of Record Linkage and Current Research Directions*, Bureau of the Census (2006).

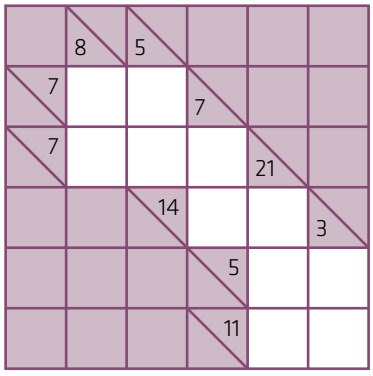
[5] W. E. Winkler, *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage*, Proceedings of the Section on Survey Research, 1990, 354–359.

× × ×

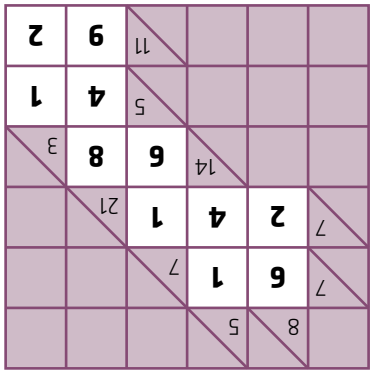
Križne vsote

↓ ↓ ↓

→ Naloga reševalca je, da izpolni bele kvadratke s števkami od 1 do 9 tako, da bo vsota števk v zaporednih belih kvadratkih po vrsticah in po stolpcih enaka številu, ki je zapisano v sivem kvadratku na začetku vrstice (stolpca) nad (pod) diagonalo. Pri tem morajo biti vse števke v posamezni vrstici (stolpcu) različne.



REŠITEV KRIŽNE VSOTE



× × ×