ELEKTROTEHNIŠKI VESTNIK 80(4): 141–146, 2013
IZVIRNI ZNANSTVENI ČLANEK

Graditev in analiza grafov slovenskih besed

Uroš Čibej

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: uros.cibej@fri.uni-lj.si

Povzetek. Organizacija različnih objektov v grafe je v zadnjem času postala temeljno sredstvo za pridobivanje novega znanja na različnih področjih. Najbolj poznana so dognanja na področju socialnih omrežij, omrežja interakcij med geni, omrežja solastništev v mednarodnih podjetjih in še številnih drugih. Glavna ideja tega članka je podati vpogled v slovenski jezik s pomočjo organizacije besed v grafe na podlagi različnih metrik. Besede bodo pomenile vozlišča grafa, vsaka beseda pa bo povezana z besedami, ki so v dani metriki njej najbližje. Predstavili bomo grafe na podlagi Hammingove in Levenshteinove razdalje. Iz zgrajenih grafov bomo izluščili nekatere osnovne značilke. Grafi in značilke le-teh so osnova za nadaljnje primerjave med jeziki, ki bi lahko dale nov vpogled v povezave med jeziki in njihov razvoj.

Ključne besede: grafi, analiza grafov, značilke grafov, slovenske besede, Levenshteinova razdalja, Hammingova razdalja, Burkhard-Kellerjeva drevesa

Constructing and analyzing graphs of the Slovene words

Organizing various sets of objects into graphs has recently become a fundamental tool for gaining new insights into various fields. The most notable achievements have been done in social, biological, financial, and several other networks. In this paper we construct graphs of words in the Slovene language based on two metrics connecting two words when their distance in the respective metric is small. From these graphs we extract several standard features presenting a basis for further study and comparison among different languages.

1 UVOD

Organizacija velikih količin informacije v obliki grafov je v zadnjih letih doživela velik razmah tako v raziskovalnih krogih kot tudi v drugih segmentih družbe: od ekonomije [16], politike [6], genetike [4], pa vse do zabavne industrije [8]. Najbolj odmevne pa so gotovo raziskave različnih socialnih omrežij [5], ki so z vseprisotnostjo medmrežnih aplikacij za socialno mreženje preprost vpogled v strukturo medosebnih odnosov.

Računalniki v zadnjih letih prodirajo tudi na področje naravnih jezikov. (Pol)avtomatsko že prevajajo besedila med različnimi jeziki, z nami že komunicirajo v naravnem jeziku ter v le-tem tudi sprejemajo in obdelujejo ukaze. Večina teh tehnologij temelji na statističnih pristopih, kar pomeni, da ne uporabljajo globljega vpogleda v samo strukturo jezika, ampak z "grobo silo"in veliko količino obstoječih besedil izluščijo bistvo novega besedila.

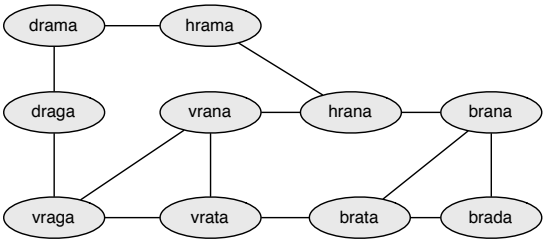
Že nekaj desetletij pa se poskuša tudi z uporabo pristopov, ki z razumevanjem strukture jezikov lažje (z manj grobe računske moči) podajo razumevanje danega besedila. V tem članku bomo z orodji za analizo velikih omrežij poskusili izboljšati ravno tovrstno razumevanje naravnih jezikov.

Besede naravnega jezika lahko organiziramo v grafe na podlagi različnih lastnosti. Naš pristop temelji na ideji Donalda Knutha, ki je za svojo knjižnico Stanford GraphBase [9] zgeneriral graf iz angleških besed dolžine 5. Graf je zgradil takole:

- vsaka beseda je vozlišče grafa
- dve besedi sta povezani, ko se razlikujeta zgolj v eni črki – tj. Hammingova razdalja med njima je enaka 1.

Primer takega grafa s slovenskimi besedami je prikazan na sliki 1.

Knuthova konstrukcija nas je motivirala, da izdelamo podobne grafe za besede slovenskega jezika, pogledamo nekatere značilnosti teh grafov in predstavimo uporabo pridobljenih informacij.



Slika 1: Primer grafa na podlagi Hammingove razdalje.

Prva naloga pri izdelavi grafa je izbira vira besed slovenskega jezika. Vsi podatki, uporabljeni v tem članku, so pridobljeni iz seznama besed, ki je bil zbran v podjetju Amebis v sodelovanju z Inštitutom Jožef Stefan in je zdaj prosto dostopen [12]. Seznam vsebuje okrog

Prejet 19. september, 2013
Odobren 10. november, 2013

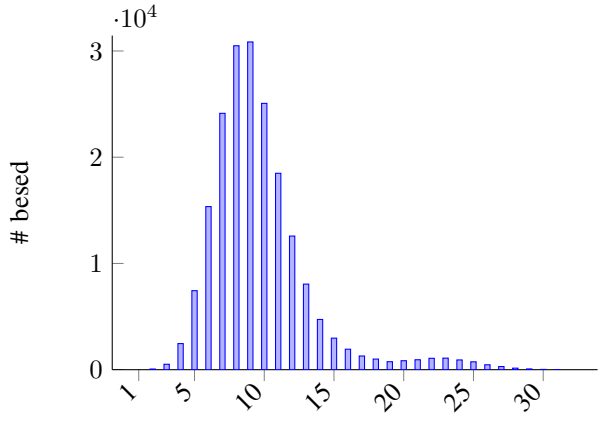
200.000 slovenskih besed (vse oblike: spregatve, sklanjatve, itn.), uporabljal pa se je v črkovalniku BesAna.

V naslednjem razdelku si bomo ogledali grafe iz teh besed na podlagi Hammingove razdalje. Ta razdalja primerja zgolj nize enakih dolžin, zato smo pri tej metriki dobili množico disjunktnih grafov in jih analizirali ločeno. V razdelku 3 bomo zgradili graf še na podlagi drugačne metrike, t. i. Levenshteinove razdalje [10]. Ta metrika je definirana tudi nad besedami različnih dolžin, kar bo posledično pomenilo tudi enoten graf za vse besede. Levenshteinova razdalja je bila izbrana, ker se pogosto uporablja v črkovalnikih oz. v algoritmih, ki za napačno črkovane besede predlaga morebitne pravilne besede. Poleg tega je v zadnjih nekaj letih postala ta metrika zanimiva tudi v primerjalnem jezikoslovju, ker zelo dobro modelira podobnosti med jeziki. Več informacij o teh primerjavah bomo navedli v sklepu.

2 Grafi na podlagi Hammingove razdalje

Hammingova razdalja zaobjame majhne spremembe nad nizi in je standardna metrika v teoriji informacij. Kot smo omenili že zgoraj, se Hammingova razdalja načeloma uporablja nad nizi iste dolžine, zato smo izmed vseh besed najprej izluščili vse enako dolge besede. Slika 2 prikazuje histogram frekvenc dolžin.

Za dolžine besed med 3 in 15 (s tem smo zajeli veliko večino besed) smo zgradili grafe neposredno po Knuthovem postopku. Tabela 1 prikazuje osnovne značilke za dobljene grafe. Vidimo lahko, da se povprečna stopnja vozlišč drastično zmanjšuje, ko imamo opravka z daljšimi besedami. Pri besedah dolžine 3 s spremembo ene črke v povprečju dobimo kar 9 novih veljavnih besed, pri besedah dolžine 15 pa v povprečju le 1,4 nove besede



Slika 2: Frekvenca dolžin slovenskih besed

Druga (bolj globalna) lastnost, ki smo si jo ogledali, je število povezanih komponent v teh grafih in njihovo velikost. Pri krajših besedah (3, 4 in 5) je večina besed povezanih v eno veliko komponento, nato pa se velikosti največjih komponent začnejo manjšati. Pri velikosti besed 6 se npr. pojavita dve veliki komponenti približno iste velikosti, pri 7 so te komponente (približno iste velikosti) že 4, pri dolžini 8 že 8 itn.

dolžina	#besed	#komp.	najv. komp.	stopnja
3	511	13	496	9.0
4	2457	106	2248	8.6
5	7437	553	6061	5.2
6	15352	2138	6076	4.2
7	24132	5044	2478	3.3
8	30493	7890	1263	3.0
9	30850	9328	1075	2.6
10	25068	9350	50	2.1
11	18487	7670	41	1.9
12	12578	5762	31	1.7
13	8055	4107	24	1.5
14	4729	2467	16	1.4
15	2968	1567	14	1.4

Tabela 1: Osnovne značilke grafov besed različnih dolžin

3 Levenshteinova razdalja in drevesa BK

Levenshteinova razdalja je še ena izmed klasičnih metričnih razdalj za primerjavo med nizi. Je ena izmed tako imenovanih "edit-distance"metrik, tj. metrik, ki opišejo, koliko transformacij je potrebnih, da pridemo iz enega niza v drugega. Bolj natančno, transformacije, ki jih dovoljuje Levenshteinova razdalja, so:

- vstavljanje novega znaka (miza $\rightarrow$ mizar),
- sprememba obstoječega znaka (miza $\rightarrow$ muza) in
- brisanje obstoječega znaka (vlak $\rightarrow$ lak).

Razdalja med besedama hiša in šah je npr.

$$d(\text{hiša}, \text{šah}) = 3.$$

Levenshteinova razdalja je široko uporabljena v praksi, predvsem v bioinformatiki, kjer se uporablja za primerjavo bioloških zaporedij [11], npr. zaporedij nukleotidov v DNK. V tej panogi se za izračun razdalje med zaporedji uporabljajo hevristike, ker so zaporedja zelo dolga. V našem primeru so besede kratke, zato smo za izračun uporabili natančen klasičen pristop, tj. dinamično programiranje [3].

Naš cilj je za vsako besedo a poiskati vse besede, ki so od a oddaljene za 1. Z naivnim pristopom je treba za obdelavo ene besede naračunati Levenshteinovo razdaljo do vseh drugih besed. Pri slovenskem slovarju je to

$$\binom{2 * 10^5}{2} \approx 2 * 10^{10}$$

izračunov Levenshteinove razdalje. V naši implementaciji je povprečen izračun Levenshteinove razdalje med dvema nizoma trajal $5\mu s$. To pomeni, da bi naivna gradnja grafa trajala več kot dva dneva.

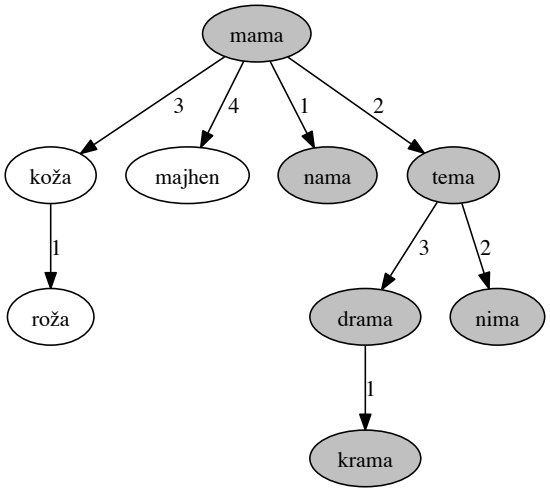
Zato smo za učinkovitejšo primerjavo poiskali boljšo podatkovno strukturo, ki zna izkoriščati dejstvo, da je Levenshteinova razdalja metrika, torej je nenegativna,

simetrična in zadošča trikotniški neenakosti. Zaradi teh lastnosti lahko s primernim predprocesiranjem organiziramo prostor besed tako, da bomo ob iskanju sosednjih besed primerjali zgolj perspektivne kandidate. Za tako organizacijo so zelo primerna Burkhard-Kellerjeva drevesa [2] oz. krajše drevesa BK.

3.1 Drevesa BK

Drevesa BK so podatkovne strukture, namenjene hranjenju objektov iz nekega metričnega prostora (najbolje prostora s celoštevilsko metriko). Podpirajo eno operacijo `query(a, k)` – to je iskanje vseh objektov oddaljenih od danega objekta `a` za največ `k`.

Vsako vozlišče v drevesu pomeni en objekt in ima več poddreves (označimo jih $1 \ldots m$). Poddrevo i vsebuje vse objekte v danem prostoru, ki so od korena oddaljeni za natanko i. Ta lastnost rekurzivno velja za vsa poddrevesa. Omenimo še, da je razdalja med dvema besedama omejena z njuno razdaljo, v najslabšem primeru namreč lahko preprosto zamenjamo vse črke. Največja stopnja vozlišč v drevesu BK je torej navzgor omejena z največjo razdaljo med dvema besedama v podanem slovarju.



Slika 3: Primer drevesa BK in eno povpraševanje `query(doma, 1)`. Osenčena so tista vozlišča, ki jih obiščemo.

Najprej si oglejmo, kako iz seznama besed zgradimo tako drevo. S postopnim dodajanjem besed iz praznega drevesa zgradimo drevo, ki vsebuje vse besede. Taka gradnja besed bi teoretično lahko pripeljala do izrojenega drevesa (npr. seznam), ki bi bilo neučinkovito za povpraševanje, vendar je bilo empirično ugotovljeno [2], da z naključnim dodajanjem besed dobimo lepo uravnoteženo drevo.

V obstoječe drevo pa takole dodajamo besede:

1) če je drevo prazno, ustvarimo novo drevo, ki ima a za koren,
2) sicer najprej izračunamo razdaljo med novim nizom a in korenom trenutnega drevesa r: $d(a, r) = k$
3) rekurzivno vstavimo a v k-to poddrevo.

Psevdokoda dodajanja v drevo je videti takole:

```
function ADD(a, tree)
    if tree is empty then
        return new BKTree(a)
    else
        k = d(a, root(tree))
        tree.subtrees[k]=ADD(a, tree.subtrees[k])
        return tree
    end if
end function
```

Ko imamo tako drevo zgrajeno, nam njegova struktura omogoča, da ob enem povpraševanju `query(a, k)` ne primerjamo niza `a` z vsemi, ampak zaradi lepih lastnosti metrik obiščemo zgolj majhen del drevesa. Slika 3 prikazuje primer drevesa, ki je bilo skonstruirano iz besed: `mama, nama, tema, drama, koža, majhen, krama , roža, nima`. Uteži na povezavah povedo, koliko so objekti v poddrevesu oddaljeni od korena, osenčena vozlišča pa so tista vozlišča, ki smo jih obiskali ob povpraševanju `query(doma,1)`. Natančnejši opis povpraševanja bomo na tem mestu izpustili, na voljo je v [2].

Že pri zelo preprostem primeru, ki je podan na sliki, vidimo, da lahko pri povpraševanju dobršen del drevesa izpustimo. V praktičnih primerih je izpuščeni del drevesa še bistveno večji. V nadaljevanju bomo natančneje empirično ovrednotili učinkovitost te podatkovne strukture, ki nam je omogočila zelo hitro zgraditev grafa.

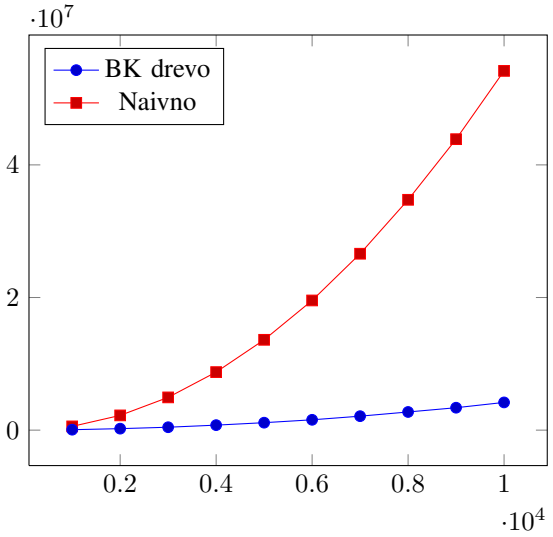
3.2 Učinkovitost dreves BK

Za zgraditev grafa besed moramo za vsako besedo poiskati vse tiste, ki so od nje oddaljene za 1. Kot smo že zgoraj omenili, je z naivnim preiskovanjem za gradnjo grafa n besed potrebnih $\binom{n}{2}$ izračunov Levenshteinove razdalje.

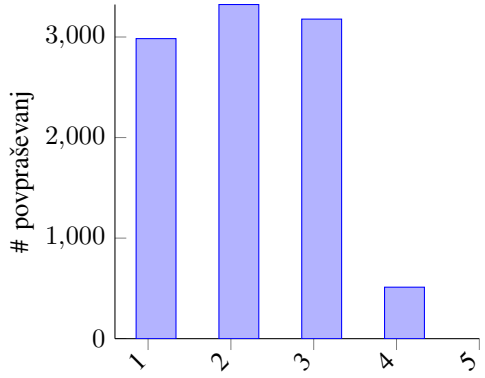
V prvem testu učinkovitosti smo za različne velikosti grafov primerjali število izračunov Levenshteinove razdalje. Slika 4 prikazuje primerjavo med drevesom BK in naivnim izračunom. Na x-osi je število besed, na y-osi pa število izračunov, ki smo jih potrebovali pri graditvi grafa.

Vidimo, da je število izračunov bistveno manjše z uporabo drevesa BK, poleg tega pa je zahtevnost praktično linearna (pri naivni graditvi je kvadratna), zato je razlika pri večjih grafih še toliko večja.

Drugi test je bil izveden na drevesu z vsemi slovenskimi besedami. Naključno smo izbirali 10.000 besed in za vsako zagnali iskanje vseh besed na razdalji 1 z uporabo drevesa BK. Merili smo odstotek preiskanega drevesa pri posameznem povpraševanju. Rezultati tega testiranja so predstavljeni na sliki 5.



Slika 4: Primerjava med naivno gradnjo in gradnjo z BK-drevesom



Slika 5: Odstotek preiskanega drevesa BK

Kot vidimo, čisto vsa povpraševanja obiščejo manj kot 5 % drevesa, velika večina celo manj kot 3 %, kar v povprečju pomeni, da se graditev pohitri kar za 50-krat kar je omočilo zgraditev grafa slovenskih besed v manj kot eni uri.

4 Graf vseh besed

Dobljeni graf si bomo ogledali na tri načine:

- prek osnovnih značilk,
- prek porazdelitve stopenj grafa in
- neposredne vizualizacije.

Vsak od teh načinov nam namreč da drugačen vpogled v strukturo grafa.

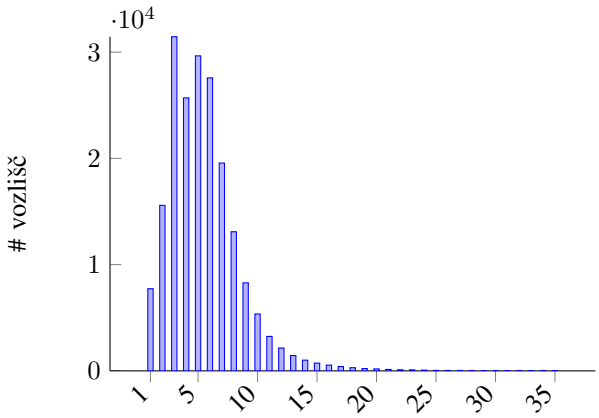
V tabeli 2 so najprej podane nekatere osnovne značilke grafa. Poudarimo lahko, da v tem grafu obstaja velika komponenta, ki vsebuje več kot 70.000 besed. Vse druge komponente so bistveno manjše, naslednja komponenta ima namreč samo okrog 1000 vozlišč. Povprečna stopnja grafa je več kot 4, kar pomeni, da z preprostimi transformacijami iz ene slovenske besede v

Značilka	Vrednost
Število vozlišč	194767
Število povezav	436165
Povprečna stopnja	4.479
Št. povezanih komponent	18050
Velikost največje komponente	71532
Koeficient gručanja	0.389

Tabela 2: Nekatere značilke grafa slovenskih besed

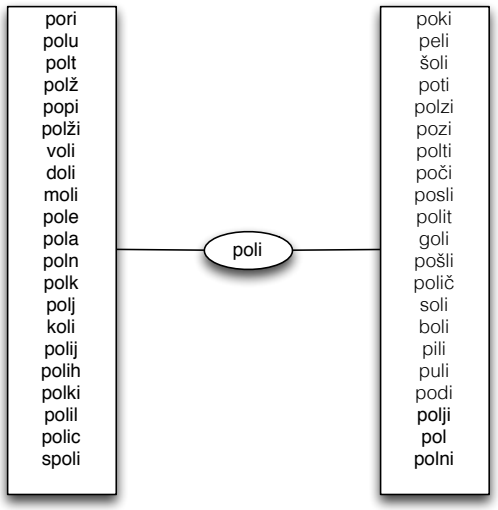
povprečju dobimo več kot štiri veljavne besede. Zadnja lastnost, ki smo jo naračunali iz tega grafa, je koeficient gručanja. Ta koeficient za eno vozlišče pove, koliko je njegova soseščina blizu polnega grafa (klike). Vrednost 1 dobimo, ko so vsi sosedje povezani med sabo, vrednost 0 pa, ko ni noben sosed povezan z nobenim drugim sosedom. Izračunana vrednost gručanja (0.389) je povprečje prek vseh vozlišč. Ta koeficient je precej visok, kar pomeni, da je graf slovenskih besed t. i. mali svet (angl. small world).

Drugi vpogled v graf nam daje histogram stopenj, ki je prikazan na sliki 6. Iz te slike vidimo, da ima porazdelitev stopenj presenetljivo “dolg rep”. Čeprav je povprečna stopnja grafa 4.479, obstajajo tudi vozlišča, ki od te vrednosti bistveno odstopajo. Najvišja stopnja v tem grafu je kar 42. To stopnjo ima beseda `poli`, vse njene sosede pa so predstavljene na sliki 7.

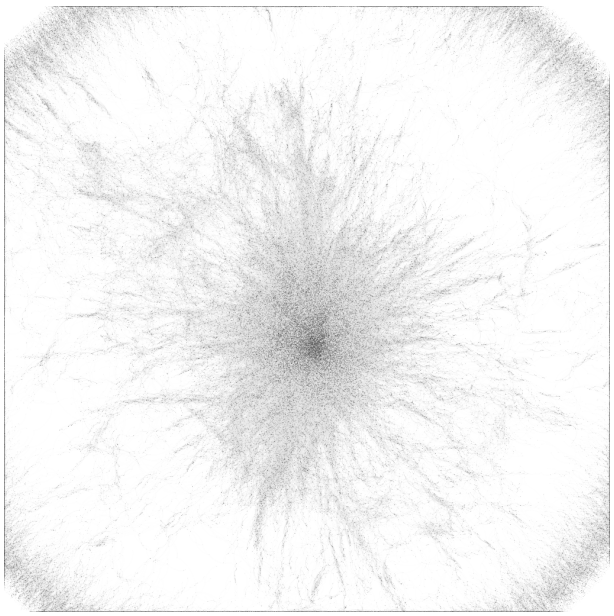


Slika 6: Histogram stopenj vozlišč slovenskega grafa

Še zadnji vpogled v graf smo dobili z neposrednim izrisom. Vizualizacija tega grafa je sama po sebi velik računski zalogaj, saj obstoječi algoritmi za izris tako velikega grafa potrebujejo veliko časa. Mi smo za risanje grafa uporabili orodje za analizo velikih omrežij Gephi [1]. Slika 8 prikazuje največjo komponento tega grafa. Iz slike vidimo, da obstaja nekakšno ”jedro”, ki je močneje povezano med seboj, iz jedra pa izhajajo kraki med seboj bolj sorodnih besed. To je še ena izmed lastnosti, ki so uporabne za nadaljnjo primerjavo med jeziki.



Slika 7: Beseda z najvišjo stopnjo 42



Slika 8: Pogled na največjo komponento grafa slovenskih besed

5 SKLEP

V tem delu smo prikazali konstrukcijo grafov besed slovenskega jezika. Vsako besedo smo predstavili z enim vozliščem grafa, vozlišča pa smo povezali, če sta besedi na razdalji 1 po določeni metriki. Za to smo uporabili dve metriki, prva je bila Hammingova razdalja, druga pa Levenshteinova razdalja. Pri Hammingovi razdalji smo dobili množico grafov, za katere smo izračunali osnovne značilnosti. Pri Levenshteinovi razdalji smo vse besede organizirali v enoten graf. Pri sami graditvi takega grafa naletimo na težave zaradi računske zahtevnosti iskanja sosedov določene besede. Zato smo pri graditvi uporabili naprednejše podatkovne strukture, drevesa BK, ki omogočajo veliko hitrejši izračun grafa. Ta pohitritev se bo izkazala še veliko bolj ključna za nadaljnje delo, ko bomo zgradili podobne grafe za veliko množico jezikov. Neposredna aplikativna uporaba pridobljenih grafov je namreč primerjalno jezikoslovje. Bogat nabor značilk, ki so prisotne v grafih, nam lahko da obilico novih informacij o jezikih.

Značilka	Vrednost
Število vozlišč	216521
Število povezav	361568
Povprečna stopnja	3.34
Št. povezanih komponent	73545
Velikost največje komponente	35532
Koeficient gručanja	0.142

Tabela 3: Nekatere značilke grafa angleških besed

V zadnjih letih je nastalo kar nekaj del [15], [7], [13], [14], ki so vpeljala Levenshteinovo razdaljo (in še nekaj modifikacij te metrike) v jezikoslovje. Denimo v [15] avtorji ugotovijo, da lahko s povprečno Levenshteinovo razdaljo med množicami besed zelo dobro modelirajo razvoj jezikov in ugotovijo, iz katerega jezika je kakšen nastal, ter celo kdaj se je to zgodilo.

Kot primer si poglejmo izračunane značilke, prikazane v tabeli 3 za graf angleških besed. Graf ima primerljivo število besed, bistveno pa se razlikujejo nekatere druge značilke. Najbolj izstopa koeficient gručanja, ki je v angleškem jeziku precej manjši. To je posledica manjšega števila oblik v primerjavi s slovenskim jezikom, podrobnejšo interpretacijo teh lastnosti pa seveda prepuščamo strokovnjakom s tega področja.

V nadaljnjem delu se bomo osredotočili na pridobivanje bogatejšega nabora informacij. Izdelali bomo grafe za številne indoevropske jezike in se povezali z jezikoslovci, s katerimi lahko najdemo informacije, ki bi lahko pripeljale do novih spoznanj o jezikih.

LITERATURA

[1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. 2009.

[2] Walter A. Burkhard and Robert M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, 1973.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

[4] Eric Davidson and Michael Levin. Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(14):4935, April 2005.

[5] Wouter de Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory social network analysis with Pajek*, volume 27. Cambridge University Press, 2005.

[6] Aleksander Žerdin. Vpliv zamenjave politične elite na omrežje ekonomske elite.

[7] Simon J Greenhill. Levenshtein distances fail to identify language relationships accurately. *Computational Linguistics*, 37(4):689–698, 2011.

[8] Brian Hopkins. Kevin Bacon and graph theory. *PRIMUS*, 14(1):5–11, 2004.

[9] Donald E. Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. ACM Press New York, 1993.

[10] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[11] David W Mount. Sequence and genome analysis. *Bioinformatics: Cold Spring Harbour Laboratory Press: Cold Spring Harbour*, 2, 2004.

[12] online. http://nl.ijs.si/gnusl/tex/tslovene/src/mte-sl.words.

[13] Filippo Petroni and Maurizio Serva. Measures of lexical distance between languages. *Physica A: Statistical Mechanics and its Applications*, 389(11):2280–2283, 2010.

[14] Job Schepens, Ton Dijkstra, FRANC Grootjen, et al. Distributions of cognates in europe as based on levenshtein distance. *Bilingualism: Language and Cognition*, 15(1):157–166, 2012.

[15] Maurizio Serva and Filippo Petroni. Indo-european languages tree by levenshtein distance. *EPL (Europhysics Letters)*, 81(6):68005, 2008.

[16] Stefania Vitali, James B. Glattfelder, and Stefano Battiston. The network of global corporate control. *CoRR*, 2011.

Uroš Čibej je leta 2007 doktoriral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani iz podvajanja podatkov v porazdeljenih sistemih. Trenutno je asistent in raziskovalec na omenjeni fakulteti. Raziskovalno se ukvarja s porazdeljenimi sistemi, snovanjem in analizo algoritmov za kombinatorično optimizacijo, npr. za iskanje vzorcev v grafih, programskimi jeziki in simulacijami.