Predviđanje vremenskih veza pomoću matričnih i tenzorskih faktorizacija

Tri tenzora - Filip Čačković, Hrvoje Olić i Bruno Ljubičić

1 Kleinbergov model - *Hubs and Authorities*

Neka je $G = (V, E)$ neusmjeren graf sa vrhovima V i bridovima E. Cilj je rangirati vrhove prema tome na koliko vrhova pokazuju (tzv. *hub* score) i prema tome koliko vrhova pokazuju na njih (tzv. *authority* score). Dakle vrh je dobar *hub* (pisat ćemo H-vrh) ako pokazuje na dobre *authority* vrhove (koje označavamo sa A-vrh). Primjerice, ako zamislimo da je G graf koji modelira internet, stranica PMF-a će imati dobar A-score, dok će stranice koje imaju poveznicu na web-stranicu PMF-a imati dobar H-score. Cilj je konstruirati algoritam koji će primiti graf G i vratiti A i H vrijednosti svakog vrha.

Za vrh $j \in V$ uzmemo $h_j^0 > 0$ i $a_j^0 > 0$ neke proizvoljne vrijednosti. Na te vrijednosti gledamo kao na H odnosno A vrijednosti vrha j. Te vrijednosti ćemo ažurirati sljedećim relacijama:

$$\begin{cases} h_j^{k+1} & \leftarrow & \sum_{i:(j,i)\in E} a_i^k \\ a_j^{k+1} & \leftarrow & \sum_{i:(i,j)\in E} h_i^k \end{cases}.$$

Dakle, H-vrijednost vrha j je suma svih A-vrijednosti onih vrhova na koje j pokazuje. Slično, A-vrijednost vrha j je suma svih H-vrijednosti onih vrhova koji pokazuju na j.

Neka je B matrica koja na mjestu (i, j) ima broj usmjerenih bridova iz vrha i u vrh j u grafu G. Ako je $V = \{j_1, \ldots, j_n\}$, definiramo

$$\mathbf{h}^k = (h_{j_1}^k, \ldots, h_{j_n}^k)^T, \quad \mathbf{a}^k = (a_{j_1}^k, \ldots, a_{j_n}^k)^T.$$

Tada se gornje relacije ažuriranja mogu zapisati kao:

$$\begin{bmatrix} \mathbf{h}^{k+1} \\ \mathbf{a}^{k+1} \end{bmatrix} = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{h}^k \\ \mathbf{a}^k \end{bmatrix}, \quad k = 0, 1, \ldots$$

Ili još jednostavnije, ako stavimo

$$x_k = \begin{bmatrix} \mathbf{h}^k \\ \mathbf{a}^k \end{bmatrix}, \quad M = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix},$$

tada je gornja relacija ekvivalentna

$$x_{k+1} = Mx_k, \quad k = 0, 1, \ldots$$

Primijetimo da je matrica M simetrična i nenegativna. Kako nas zanimaju samo odnosi H i A vrijednosti, možemo promatrati normirani niz:

$$z_0 = x_0 > 0, \quad z_{k+1} = \frac{Mz_k}{\|Mz_k\|_2}, \quad k = 0, 1, \ldots \tag{1}$$

Htjeli bismo provjeriti pod kojim uvjetima ovakav niz konvergira. Ideja je uzeti limes niza z_k kao definiciju za H i A vrijednosti grafa.

Međutim, brzo ćemo vidjeti da ovaj niz ne konvergira uvijek. Ipak, pokazat ćemo da parni i neparni podnizovi uvijek konvergiraju:

$$z_{\text{even}} = \lim_k z_{2k}, \quad z_{\text{odd}} = \lim_k z_{2k+1}.$$

Zapravo, za simetrične i nenegativne matrice M, niz dan sa (1) uvijek zadovoljava to svojstvo.

Drugi problem je što limesi podnizova z_{even}, z_{odd} općenito ovise od z_0. Stoga definiramo:

$$Z = \{z_{\text{even}}(z_0), z_{\text{odd}}(z_0) : z_0 > 0\}.$$

Ideja je odabrati jedan $z^* \in Z$ koji će nam biti definicija za H i A vrijednosti. Pokazat ćemo da $z_{\text{even}}(\mathbf{1})$ zadovoljava nekoliko lijepih svojstava koji ga izdvajaju kao prirodni izbor - iako je izbor i dalje prilično arbitraran.

2 Drugi primjer

Neka je $G = (V, E)$ graf sa matricom susjedstva B. Neka su sada svakom vrhu $j \in V$ pridružene tri vrijednosti x_{i1}, x_{i2}, x_{i3} po principu:

$$1 \to 2 \to 3.$$

Odnosno, vrijednosti inicijaliziramo sa nekim pozitivnim $x_{i1}^0, x_{i2}^0, x_{i3}^0$ pa ih ažuriramo relacijom

$$\begin{aligned}
x_{i1}^{k+1} &\leftarrow \sum_{j:(i,j)\in E} x_{i2}^k \\
x_{i2}^{k+1} &\leftarrow \sum_{j:(j,i)\in E} x_{i1}^k + \sum_{j:(i,j)\in E} x_{i3}^k \\
x_{i3}^{k+1} &\leftarrow \sum_{j:(j,i)\in E} x_{i2}^k
\end{aligned}$$

ili u matričnom obliku:

$$\begin{bmatrix} \mathbf{x}_1^{k+1} \\ \mathbf{x}_2^{k+1} \\ \mathbf{x}_3^{k+1} \end{bmatrix} = \begin{bmatrix} 0 & B & 0 \\ B^T & 0 & B \\ 0 & B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^{k} \\ \mathbf{x}_2^{k} \\ \mathbf{x}_3^{k} \end{bmatrix},$$

što opet možemo zapisati kao: $x_{k+1} = Mx_k$. Sada je situacija identična onoj u prethodnom primjeru i svi argumenti konvergencije vrijede i ovdje.

3 Generalni slučaj

Pretpostavimo da imamo dva grafa $G_A = (V_A, E_A)$ i $G_B = (V_B, E_B)$ sa brojem vrhova n_A odnosno n_B. O grafu G_A razmišljamo kao o strukturnom grafu koji igra ulogu grafova

$$hubs \to autorities, \quad 1 \to 2 \to 3$$

u prethodnim primjerima. Za sve $i \in V_B$ i $j \in V_A$ definiramo početne vrijednosti $x_{ij}^0 > 0$ i ažuriramo ih pomoću relacije

$$x_{ij} \leftarrow \sum_{r:(r,i)\in E_B,\ s:(s,j)\in E_A} x_{rs} + \sum_{r:(i,r)\in E_B,\ s:(j,s)\in E_A} x_{rs}.$$

Također, relacija se može napisati matrično. Neka je $X_k = (x_{ij}^k) \in \mathbb{R}^{n_B \times n_A}$ te A i B matrice susjedstva grafova G_A odnosno G_B. Tada je gornja relacija ekvivalentna:

$$X_{k+1} = BX_kA^T + B^TX_kA, \quad k = 0, 1, \ldots$$

Glavni rezultat koji je uporište za primjenu algoritma je sljedeći teorem:

Teorem: Neka su A i B matrice susjedstva grafova G_A odnosno G_B. Neka je $Z_0 > 0$ unaprijed odabrana pozitivna matrica. Definiramo:

$$Z_{k+1} = \frac{BX_kA^T + B^TX_kA}{\|BX_kA^T + B^TX_kA\|_F}, \quad k = 0, 1, \ldots$$

Tada matrični podnizovi Z_{2k} i Z_{2k+1} konvergiraju k Z_{even} i Z_{odd}. Nadalje, među svim matricama u skupu:

$$Z = \{Z_{\text{even}}(Z_0), Z_{\text{odd}}(Z_0) : Z_0 > 0\}$$

matrica $Z_{even}(\mathbf{1})$ je jedinstvena matrica najveće 1-norme.

Za definiciju matrice sličnosti, uzet ćemo upravo $Z_{even}(\mathbf{1})$. Iz definicije i gornjeg teorema navodimo aproksimacijski algoritam za računanje matrica sličnosti među grafovima:

1. `Postavi` $Z_0 = \mathtt{1}$
2. `Iteriraj paran broj iteracija do konvergencije:`

$$Z_{k+1} = \frac{BX_kA^T + B^TX_kA}{\|BX_kA^T + B^TX_kA\|_F}$$

3. `Vrati zadnju vrijednost` Z_k

4 Grafovi i nenegativne matrice

Svakom usmjerenom grafu $G = (V, E)$ pridružujemo nenegativnu matricu susjedtsva B koja je indeksirana po vrhovima. Njeni elementi b_{ij} predstavljaju broj bridova iz vrha i u vrh j. Slično $(B^k)_{ij}$ predstavlja broj puteva duljina k iz vrha i u vrh j. Kažemo da je graf jako povezan ako za svaka dva vrha $i, j \in V$ postoji $k \in \mathbb{N}$ takav da je $B_{ij}^k > 0$. Za matrice čiji je graf jako povezan kažemo da su ireducibilne.

Može se pokazati da je to svojstvo za nenegativnu matricu M ekvivalentno svojstvu da ne postoji matrica permutacije P za koju vrijedi:

$$PMP^T = \begin{bmatrix} A & C \\ 0 & D \end{bmatrix}$$

pri čemu su A i D kvadratne matrice. Ako dodatno pretpostavimo da je M simetrična matrica - to je ekvivalentno tome da ne postoji matrica permutacije P takva da je

$$PMP^t = \begin{bmatrix} A & 0 \\ 0 & D \end{bmatrix},$$

pri čemu su A i D kvadratne simetrične matrice.
Odavdje lako slijedi, da ako simetrična nenegativna matrica M nije ireducibilna, da postoji matrica permutacije P takva da je

$$PMP^T = \begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_r \end{bmatrix}$$

pri čemu su M_i ireducibilne nenegativne simetrične matrice ili 0 za i=1,. . . ,r. Iskažimo važan rezultat za nenegativne ireducibilne matrice.

Teorem (Perron-Frobenius) Neka je B nenegativna ireducibilna matrica i neka je ρ njen spektralni radijus, tj.

$$\rho = \max_{\lambda \in \sigma(B)} |\lambda|.$$

Tada je ρ svojstvena vrijednost od B algebarskog multipliciteta 1 za koju postoji nenegativan svojstveni vektor $x \geq 0$. ρ nazivamo Perronov korjen od B, a vektor $v = \frac{x}{\|x\|_2}$ nazivamo Perronov vektor od M.

U simetričnom slučaju može se napraviti i više:

Teorem 1. Neka je M simetrična i nenegativna matrica spektralnog radijusa ρ. Tada je ρ svojstvena vrijednost od M čiji su algebarski i geometrijski multipliciteti jednaki. Nadalje, postoji nenegativna matrica $X \geq 0$ čiji stupci razapinju pripadni svojstveni potprostor od ρ.

Dokaz. Gore smo pokazali da za svaku nenegativnu simetričnu matricu M postoji matrica permutacija P takva da je

$$PMP^T = \begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_r \end{bmatrix}$$

pri čemu su M_i simetrične, nenegativne i ireducibilne matrice ili 0. Po Perron-Frobeniusovom teoremu za $M_i \neq 0$, postoji Perronov korjen ρ_i sa pripadnim Perronovim vektorom x_i. Budući da je spektar od M upravo unija spektara od M_i, slijedi da ρ mora biti jednak nekom ρ_i - dakle ρ je svojstvena vrijednost od M. Neka bez smanjenja općenitosti za $i = 1, \ldots, k \leq r$ vrijedi $\rho = \rho_i$, a za

$k+1, \ldots, r$ vrijedi $\rho > \rho_i$. Odavde lako slijedi da su i algebarski i geometrijski mutliplicitet od ρ jednaki r. Matricu $X \geq 0$ dobivamo kao

$$X = P^T \begin{bmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_k \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}.$$

Zaista,

$$MX = P^T \begin{bmatrix} M_1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & M_2 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & M_k & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & M_{k+1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & M_r \end{bmatrix} PP^T \begin{bmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_k \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}$$

$$= P^T \begin{bmatrix} M_1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & M_2 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & M_k & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & M_{k+1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & M_r \end{bmatrix} \begin{bmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_k \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}$$

$$= P^T \begin{bmatrix} M_1x_1 & 0 & \ldots & 0 \\ 0 & M_2x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & M_kx_k \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix} = \rho P^T \begin{bmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_k \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix} = \rho X$$

□

Teorem 2. Neka je M nenegativna simetrična matrica spektralnog radijusa ρ. Neka je $z_0 > 0$ proizvoljan pozitivan vektor. Promotrimo niz:

$$z_{k+1} = \frac{Mz_k}{\|Mz_k\|_2} = \frac{M^k z_0}{\|M^k z_0\|_2},\ k = 0, 1, \ldots$$

Postoje dva slučaja:

(i) Ako $-\rho$ nije svojstvena vrijednost od M, tada z_k konvergira prema

$$\frac{\Pi z_0}{\|\Pi z_0\|}$$

pri čemu je Π matrica ortogonalnog projektora na svojstveni potprostor od ρ.

(ii) Ako $-\rho$ je svojstvena vrijednost od M, tada z_k ne konvergira, ali konvergiraju njegovi parni i neparni podnizovi:

$$z_{\text{even}}(z_0) = \lim_k z_{2k} = \frac{\Pi z_0}{\|\Pi z_0\|_2}, \quad z_{\text{odd}}(z_0) = \lim_k z_{2k+1} = \frac{\Pi M z_0}{\|\Pi M z_0\|_2}$$

pri čemu je Π ortogonalni ortogonalni projektor na ortogonalnu sumu svojstvenih potprostora od ρ i $-\rho$.

U oba slučaja, skup svih limesa je

$$Z = \{z_{\text{even}}(z), z_{\text{odd}}(z) : z > 0\} = \left\{ \frac{\Pi z}{\|\Pi z\|_2} : z > 0 \right\}$$

i $z_{\text{even}}(\mathbf{1})$ je jedinstveni vektor najveće 1-norme u Z.

Dokaz. Dokazat ćemo samo drugi slučaj, prvi slučaj se pokazuje analogno.
(ii). Neka je $-\rho$ svojstvena vrijednost od M. Neka su $\mathcal{V}_\rho$, $\mathcal{V}_{-\rho}$ i $\mathcal{V}_\mu$ pripadni svojstveni potprostori od redom ρ, $-\rho$ i ostatka spektra. Pretpostavimo da je $\mathcal{V}_\mu \neq \{0\}$, inače je situacija trivijalna. Neka su V_ρ, $V_{-\rho}$ ortogonalne (pravokutne) matrice sastavljene od svojstvenih vektora prirdruženih ρ i $-\rho$, a V_μ ortogonalna matrica sastavljena od neke ortogonalne baze za $\mathcal{V}_\mu$. Dakle za njih vrijedi:

$$MV_\rho = \rho V_\rho, \quad MV_{-\rho} = -\rho V_{-\rho}, \quad MV_\mu = V_\mu M_\mu$$

pri čemu je M_μ neka kvadratna simetrična matrica, spektralnog radijusa $\mu < \rho$. Sada se M može rastaviti kao

$$M = [V_\rho, V_{-\rho}, V_\mu] \begin{bmatrix} \rho I & & \\ & -\rho I & \\ & & M_\mu \end{bmatrix} [V_\rho, V_{-\rho}, V_\mu]^T = \rho V_\rho V_\rho^T - \rho V_{-\rho} V_{-\rho}^T + V_\mu M_\mu V_\mu^T$$

Iz čega slijedi

$$M^2 = \rho^2 (V_\rho V_\rho^T + V_{-\rho} V_{-\rho}^T) + V_\mu M_\mu^2 V_\mu^T = \rho^2 \Pi + V_\mu M_\mu^2 V_\mu^T$$

pri čemu je Π ortogonalni projektor na $\mathcal{V}_\rho \oplus \mathcal{V}_{-\rho}$. Slično:

$$M^{2k} = \rho^{2k} \Pi + V_\mu M_\mu^{2k} V_\mu^T.$$

Budući da su $\rho^{2k}\Pi$ i $V_\mu M_\mu^{2k} V_\mu^T$ ortogonalni, za $z_0 > 0$ slijedi:

$$\left\|M^{2k} z_0\right\|_2 = \left\|\rho^{2k}\Pi z_0 + V_\mu M_\mu^{2k} V_\mu^T z_0\right\|_2 \geq \left\|\rho^{2k}\Pi z_0\right\|_2 = \rho^{2k}\|\Pi z_0\|_2.$$

Za nenegativne vektore $a, b \geq 0$ pišemo $a \leq b$ ako je $a - b \leq 0$. Dakle imamo

$$\frac{M^{2k} z_0}{\|M^{2k} z_0\|_2} \leq \frac{M^{2k} z_0}{\rho^{2k}\|\Pi z_0\|_2} = \frac{\rho^{2k}\Pi z_0 + V_\mu M_\mu^{2k} V_\mu^T z_0}{\rho^{2k}\|\Pi z_0\|_2} = \frac{\Pi z_0}{\|\Pi z_0\|_2} + \frac{1}{\|\Pi z_0\|_2}\left(V_\mu\left(\frac{1}{\rho}M_\mu\right)^2 V_\mu^T\right)^k z_0.$$

Lako se vidi da matrica

$$\left(V_\mu\left(\frac{1}{\rho}M_\mu\right)^2 V_\mu^T\right)$$

ima spektralni radijus $\left(\frac{\mu}{\rho}\right)^2 < 1$. Može se pokazati da za proizvoljnu matricu $A \in M_n$ vrijedi da ako je njen spektralni radijus manji od 1, da je tada $\lim_k A^k = 0$.
Sada je

$$0 \leq \left\| \frac{M^{2k} z_0}{\|M^{2k} z_0\|_2} - \frac{\Pi z_0}{\|\Pi z_0\|_2} \right\|_2 \leq \frac{1}{\|\Pi z_0\|_2} \left\| \left(V_\mu\left(\frac{1}{\rho}M_\mu\right)^2 V_\mu^T\right)^k z_0 \right\|_2 \xrightarrow{k\to\infty} 0.$$

Sada slijedi:

$$z_{\text{even}}(z_0) = \lim_k z_{2k} = \frac{\Pi z_0}{\|\Pi z_0\|_2}, \quad z_{\text{odd}}(z_0) = \lim_k z_{2k+1} = \frac{\Pi M z_0}{\|\Pi M z_0\|_2}.$$

Uočimo da Πz_0 i $\Pi M z_0$ nisu 0 jer $z_0 > 0$, a Π i M su netrivijalne nenegativne matrice. Sada iz nenegativnosti od M (pa je $Mz_0 > 0$) i formula za $z_{\text{even}}(z_0)$ i $z_{\text{odd}}(z_0)$ slijedi da oba limesa leže u

$$\left\{\frac{\Pi z}{\|\Pi z\|_2} : z > 0\right\}.$$

Vrijedi i obratno, za $\hat{z}_0 \in \left\{\frac{\Pi z}{\|\Pi z\|_2} : z > 0\right\}$ tvrdimo da postoji neki $z_0 > 0$ takav da je

$$\hat{z}_0 = z_{\text{even}}(z_0).$$

Znamo da je Π nenegativna, pa je $\hat{z}_0 \geq 0$, no taj vektor možda ima neke komponente jednake 0. Sada konstruiramo z_0^ε tako da je jednak $\hat{z}_0$ na pozitivnim mjestima, a jednak $\varepsilon > 0$ drugdje. Primijetimo da će $\hat{z}_0$ na mjestu i imati 0 ako i samo ako je i-ti redak od $\Pi = (\pi_{ij})_{ij=1}^n$ jednak 0. Zaista

$$0 = (\hat{z}_0)_i \iff 0 = (\Pi z_0)_i = \sum_{k=1}^n \pi_{ki}(z_0)_i \iff \pi_{ki} = 0, k = 1, \ldots, n.$$

Odavde odmah slijedi da je $\Pi(z_0 - z_0^\varepsilon) = 0$, odnosno

$$\Pi z_0^\varepsilon = \Pi \hat{z}_0 = \hat{z}_0,$$

odnosno,

$$z_{\text{even}}(z_0^\varepsilon) = \frac{\Pi z_0^\varepsilon}{\|\Pi z_0^\varepsilon\|_2} = \hat{z}_0.$$

Sada dokazujemo zadnju tvrdnju. Matrica Π je nenegativna te zadovoljava $\Pi = \Pi^T = \Pi^2$. Stoga vrijedi:

$$\|\Pi\mathbf{1}\|_2 = \sqrt{\mathbf{1}^T\Pi\mathbf{1}}, \quad \|\Pi\mathbf{1}\|_1 = \mathbf{1}^T\Pi\mathbf{1},$$

pa je

$$\left\| \frac{\Pi\mathbf{1}}{\|\Pi\mathbf{1}\|_2} \right\|_1 = \sqrt{\mathbf{1}^T\Pi\mathbf{1}} = \|\Pi\mathbf{1}\|_2$$

sada primjenom CSB nejednakosti slijedi:

$$\left\| \frac{\Pi z_0}{\|\Pi z_0\|_2} \right\|_1 = \frac{\mathbf{1}^T\Pi z_0}{\|\Pi z_0\|_2} = \frac{\mathbf{1}^T\Pi^2 z_0}{\|\Pi z_0\|_2} = \frac{(\Pi\mathbf{1})^T(\Pi z_0)}{\|\Pi z_0\|_2} \leq \|\Pi\mathbf{1}\|_2 = \left\| \frac{\Pi\mathbf{1}}{\|\Pi\mathbf{1}\|_2} \right\|_1,$$

pri čemu se jednakost postiže ako i samo ako je za neki $\lambda \in \mathbb{C}$ vrijedi

$$\Pi z_0 = \lambda\Pi\mathbf{1} \implies z_{\text{even}}(z_0) = \frac{\Pi z_0}{\|\Pi z_0\|_2} = \frac{\Pi\mathbf{1}}{\|\Pi\mathbf{1}\|_2} = z_{\text{even}}(\mathbf{1}).$$

□

5 Generalizacija Hubs & Authorities algoritma

Vratimo se na promatranje samo jednog grafa $G_B = (V, E)$ s matricom susjedstva B. U Kleinbergovom modelu hubs i authorities score tog grafa se računaju kao dominantni svojstveni vektori matrica B^TB i BB^T, redom. Konkretno, može se pokazati da vrijedi sljedeći teorem.

Teorem Neka je B matrica susjedstva grafa G_B. Normiranih (u normi $||\cdot||_2$) svojstveni vektori pripadne dominantne svojstvene vrijednosti matrica BB^T i B^TB odgovaraju hubs i authorities score vektorima grafa G_B, uz uvjet da je Perronova vrijednost kratnosti 1. Ukoliko taj uvjet nije zadovoljen, hubs i authorities score vektori odgovaraju projekciji vektora $\mathbf{1}$ na pripadne dominantne svojstvene potprostore.

Pokažimo da je naša definicija sličnosti vrhova grafova zaista generalizacija Kleinbergovog modela. Naime, promotrimo sljedeći strukturni graf:

$$hubs \longrightarrow authorities$$

Tvrdimo da sličnost vrhova grafa G_B s vrhovima *hubs* i *authorities* zaista daju, redom, hubs i authorities scoreove vrhova od G_B. Naime, vektor sličnosti x_k definirat ćemo nizom iteracija:

$$x_{k+1} = (A \otimes B + A^T \otimes B^T)\, x_k = Mx_k$$

gdje će nam A predstavljati matricu susjedstva strukturnog grafa, tj.

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Tada imamo sljedeće:

$$M = A \otimes B + A^T \otimes B^T = \begin{bmatrix} 0 & B \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$$

Tada, jer algoritam računa niz parnih potencija za M, vrijedi da će rezultantni vektor x biti aproksimacija svojstvenog vektora dominantne svojstvene vrijednosti matrice:

$$M^2 = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} = \begin{bmatrix} BB^T & 0 \\ 0 & B^TB \end{bmatrix}$$

Iz ovoga upravo vidimo da će pripadni hubs i authorities vektori biti računati kao svojstvene vrijednosti matrica BB^T i B^TB metodom potencija.

6 Centralnost

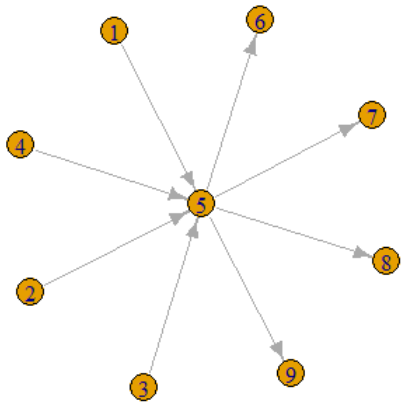
Neka je ponovno dan graf $G_B = (V, E)$ s matricom susjedstva B. U ovom poglavlju ćemo promatrati sličnost s drugim strukturnim grafom:

$$1 \longrightarrow 2 \longrightarrow 3$$

Uvest ćemo pojam centralnosti kao sličnost s vrhom 2 u strukturnom grafu. S predavanja nam je poznat sljedeći teorem, koji opisuje kako računati tu mjeru.

Teorem Neka je B matrica susjedstva grafa G_B. Normirani scoreovi centralnosti vrhova grafa G_B dani su kao normalnizirani svojstveni vektor dominantne svojstene vrijednosti matrice $B^TB + BB^T$, uz uvjet kratnosti 1 Perronovog korijena, inače kao normirana projekcija vektora $\mathbf{1}$ na pripadni svojstveni potprostor.

Ilustrirajmo primjerom zašto je ovo bolja mjera centralnosti od primjerice hub ili authority scorea iz Kleinbergovog modela. U tu svrhu, promotrimo sljedeći graf: Ako u tom grafu imamo m vrhova



Usmjereni "mašna" (bowtie) graf

slijeva, te n vrhova zdesna, te ako ih poredamo tako da za prvoga uzmemo centralni vrh, potom

lijeve, a onda desne vrhove, matrica susjedstva je dana s:

$$B = \left[\begin{array}{c|c|c} 0 & 0\dots0 & 1\dots1 \\ \hline 1 & & \\ \vdots & \mathbf{0_n} & \mathbf{0} \\ 1 & & \\ \hline 0 & & \\ \vdots & \mathbf{0} & \mathbf{0_m} \\ 0 & & \end{array}\right]$$

Tada je matrica $BB^T + B^TB$ kojoj računamo Perronov vektor jednaka

$$BB^T + B^TB = \begin{bmatrix} m+n & 0 & 0 \\ 0 & \mathbf{1}_n & 0 \\ 0 & 0 & \mathbf{1}_m \end{bmatrix}$$

Odmah je jasno da je Perronov korijen jednak $\rho = \sqrt{m+n}$ i da je matrica sličnosti jednaka:

$$\mathbf{S} = \frac{1}{\sqrt{m+n+1}} \left[\begin{array}{c|c|c} 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{array}\right]$$

Dakle, iz matrice sličnosti jasno vidimo kako centalni vrh zaista jest najsličniji vrhu 2 iz strukturnog grafa, te u ovom primjeru sličnost zaista daje dobru mjeru centralnosti vrha u grafu, pa imamo opravdanje za uvesti mjeru kao sličnost s vrhom 2 u strukturnom grafu. Napomenimo i da ovaj rezultat ne ovisi o vrijednostima m i n (naravno uz pretpostavku da su i m i $n > 0$).

S druge strane, ako računamo hubs i authorities scoreove vrhova. U ovisnosti o odnosu m i n, tu ćemo imati sljedeće situacije:

$$\mathbf{S}_{m>n} = \frac{1}{\sqrt{m+1}} \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \hline 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{array}\right], \quad \mathbf{S}_{m<n} = \frac{1}{\sqrt{n+1}} \left[\begin{array}{c|c} 0 & 1 \\ \hline 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{array}\right],$$

Ovdje uočavamo dva bitna problema s modelom. Prvo, rezultat se drastično mijenja u ovisnosti o slučaju $m < n$, odnosno $m > n$. Drugo, vidimo da u oba slučaja nema razlike u scoreu centralnog vrha u odnosu na scoreove ostalih vrhova grafa. Stoga ćemo se u nastavku fokusirati na primjene našeg modela kako bi mjerili centralnost pojedinih vrhova grafa.

7 Algoritam za traženje sinonima

7.1 O algoritmu

Dan je sljedeći problem: dan nam je jednojezični rječnik u formatu

$$\textit{riječ} : \textit{definicija}.$$

Točnije, dan nam je graf generiran tim rječnikom čiji su vrhovi riječi, a brid između riječi r_1 i r_2 postoji, ako se r_2 pojavljuje u definiciji od r_1. Cilj je na za danu riječ iz rječnika, naći drugu riječ koja je će mu po značenju biti "najbliža". Idealno bi bilo kada bi za danu riječ, algoritam vratio njen sinonim, ali mnoge riječi nemaju jasno definirane sinonime pa ćemo se zadovoljiti sa riječima koje najbolje opisuju traženu riječ. Ideja je iskoristiti rezultate prethodnog poglavlja - točnije, za danu riječ, iz grafa izvučemo podgraf generiran svim njenim susjedima (ali bez tražene riječi). Ideja je da kada bi tražena riječ bila u podgrafu, podgraf bi imao strukturu grafa "mašne", pri čemu bi se tražena riječ nalazila u "čvoru". Kako se ta riječ ne nalazi u podgrafu, ideja je da bi dobar sinonim preuzeo tu zadaću.

Naš se algoritam sastoji od sljedećih koraka:
1. `Dani su riječ` r `i graf` $G = (V, E)$

2. `Nađi skup svih susjeda` $V' \subseteq V$ `riječi` r`,`

$$V' = \{r' \in V : (r', r) \in E \text{ ili } (r, r') \in E\}$$

3. `Nađi skup svih bridova` $E' = E \cap (V' \times V')$ `induciranog podgrafa.`
4. `Za taj inducirani pograf (V',E') i graf`

$$1 \to 2 \to 3$$

`nađi njihovu matricu sličnosti` S`.`
5. `Vrati riječ koja odgovara najvećoj vrijednosti u srednjem stupcu matrice` S`.`

7.1.1 Implementacija

U ovom poglavlju, opisat ćemo neke od ideja koje mogu značajno ubrzati izvođenje algoritma ili dovesti do značajne uštede memorije.

Vrhove našeg grafa poistovjećujemo sa sortiranim vektorom V indeksa riječi u sortiranom rječniku, dok skup bridova prikazujemo dvostupčanom matricom $E \in M_{m2}$, pri čemu je i-ti redak

$$E[i,] = (a, b)$$

ako postoji brid $a \to b$. E je sortirana leksikografski po prvom, pa po drugom stupcu kako bi ubrzali traženja. Vektor susjeda V' dobivamo kao:

$$V' = \text{sort}(E[E[,2] == r, 1] \cup E[E[,1] == r, 2]),$$

a matricu podbridova E' kao

$$E' = [(E[,1] \cap V'), (E[,2] \cap V')].$$

Konačno, promatramo matricu $E'' \in M_{m2}$,

$$E'[k,] = (i,j) \implies E''[k,] = (\varphi(i), \varphi(j)), \quad k = 1, \ldots, m$$

pri čemu je $\varphi : V \to \{1, \ldots, n\}$ funkcija koja za dani vrh i vraća poziciju od i u V'. Funkciju φ možemo efikasno evaluirati korištenjem binarnog traženja s obzirom da je V' sortirani vektor.

Prisjetimo se, ključni korak pri računanju matrice sličnosti, bilo je računanje

$$BZA^T + B^T ZA,$$

pri čemu su B i A matrice susjedstva pripadnih grafova.
Glavni motiv ovog poglavlja je izbjegavanje alociranja memorije matrice susjedstva B. Naime, za neke riječi, vektor vrhova podgrafa V' može biti velik, a da taj podgraf nema puno bridova. Ako je $\text{card}(V') = n$, tada je matrica susjedstva iz M_n. Primjerice riječ *man* je spojena sa 1076 vrhova, dok pripadni podgraf ima samo 1836 bridova što je znatno manje od 1076^2. Ako gledamo veću dubinu susjeda, tada će matrica susjedstva biti još veća, pa alociranje takve matrice ne bi bilo moguće. Ispostavi se da alokacija u ovom slučaju nije potrebna već se sva matrična množenja mogu dobiti iz matrice $E'' \in M_{m2}$. Neka je $\Delta_{ij} \in M_n$ matrica koja ima jedinicu na mjestu (i,j), a drugdje 0. Tada za proizvoljan $Z \in M_{n3}$ vrijedi da je $\Delta_{ij} Z \in M_{n,3}$ matrica koja za i-ti redak ima ima $Z[j,]$. Primjetimo da se matrica B može prikazati kao:

$$B = \sum_{k=1}^{m} \Delta_{E''[k,1]E''[k,2]}$$

odnosno umnožak

$$BZ = \sum_{k=1}^{m} \Delta_{E''[k,1]E''[k,2]} Z$$

koji se svodi na m dodavanja redaka od Z. Ovaj način množenja BZ "košta" $3m$ zbrajanja, dok klasično matrično množenje ima ukupno $3n^2$ zbrajanja (od kojih su $n^2 - m$ zbrajanja s nulom). U najgorem slučaju (kada je B matrica jedinica) imamo $m = n^2$.
Za transponiran slučaj, uočimo da vrijedi

$$B^T = \sum_{k=1}^{m} \Delta^T_{E''[k,1]E''[k,2]} = \sum_{k=1}^{m} \Delta_{E''[k,2]E''[k,1]}$$

pa vrijedi analogan rezultat.
Nadalje za matricu susjedstva A grafa $1 \to 2 \to 3$ vrijedi

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad A^T = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

pa za $Z \in M_{m3}$ vrijedi

$$ZA^T = [\mathbf{0}_n, Z[,1], Z[,2]], \quad ZA = [Z[,2], Z[,3], \mathbf{0}_n].$$

7.2 Rezultati u engleskom jeziku

Graf engleskog rječnika sastoji se od 111 988 vrhova i 1 398 424 bridova. Zbog postizanja boljih rezultata, odlučujemo se u računanje podgrafa ne uključiti 183 riječi koje su spojene s najviše drugih riječi. Takve riječi bi i u pripadnim podgrafovima bile predominantne. Takve riječi su pretežno veznici, čestice, zamjenice koje vjerojatno ne bi niti bile ničjiji sinonim. Ipak, pojavljuju se i neke imenice poput *act*, *manner*, *quallity* i *person*.

Kao što je već opisano, za unaprijed odabranu riječ ne očekujemo da će algoritam predložiti njezin sinonim, ali će predložiti skup riječi koji ju opisuju, među kojima ipak očekujemo da bi se mogao naći i pravi sinonim. Obzirom da je jezik bogata struktura u kojoj čak riječi sastavljene od istih slova mogu imati različita znalčenja (npr. *luk* kao povrće, oružje, vrsta nadsvođenja, dio kružnice), teško je za očekivati jednoznačne odgovore. Među biranim riječima preferiramo one koje generiraju veći podgraf. Iz riječnika engeskog jezika odabiremo nekoliko riječi za koje očekujemo da imaju veći broj sinonima. Sam odabrir riječi je donekle proizvoljan: *Ask*, *Begin*, *Fast*, *Get*, *Story*.

	Ask		Begin		Fast		Get		Story	
Broj vrhova s kojima je povezana	68		119		240		214		160	
Broj bridova podgrafa	124		203		701		489		288	
Ukupan broj iteracija	34		26		26		28		31	
Vrijeme izvršavanja	0,3611979 secs		0,2872319 secs		1,224609 secs		1,14322 secs		0,7271762 secs	
1	demand	0,2082008	beginning	0,1854655	hold	0,23034976	gain	0,22796147	event	0,1852573
2	question	0,1967142	new	0,1653341	fasten	0,17234227	obtain	0,16667877	narrative	0,1615132
3	inquire	0,1709743	originate	0,1607697	stick	0,13140006	come	0,13658899	narration	0,1558927
4	request	0,1577772	initiate	0,1586397	fix	0,12779702	reach	0,13383302	tale	0,1449304
5	seek	0,1448598	rise	0,1392863	lock	0,10162057	draw	0,12653646	relation	0,1424001
6	entreat	0,1419865	start	0,13622	close	0,09195036	bring	0,12631254	fable	0,1385712
7	asking	0,136011	commence	0,1317253	stop	0,08789086	attain	0,12290919	recital	0,1372981
8	inquiry	0,1328396	enter	0,1266303	fixed	0,08763296	procure	0,12034992	tell	0,1341748
9	petition	0,1091138	course	0,1210939	bind	0,08449491	win	0,10455783	romance	0,1138621
10	query	0,1058134	move	0,1098665	attach	0,0816429	arrive	0,09900043	history	0,1128431

Za riječ *Ask* algoritam dosta dobro predlaže riječi koje ju opisuju. Među rezultatima imamo i riječi koje ovisno o kontekstu mogu predstavljati sinonime. Slično možemo vidjeti i kod riječi *Begin*, *Get* i *Story*. Kod riječi *Fast* algoritam nije uspio pronaći zadovoljavajuće riječi poput *hurry*, *quick*.

7.3 Hrvatski rječnik i modificirana Levenshteinova udaljenost

Za razliku od engleskog rječnika, hrvatski rječnik nismo dobili već formatiran kakav nam treba da bismo ga ubacili u algoritam, stoga se ovo poglavlje bavi opisivanjem procesa nastajanja takvog rječnika. Riječi i definicije smo preuzeli sa web stranice `https://rjecnik.hr/`. Skup podataka, sastoji se od oko 30000 riječi i definicija. Nakon prvotnog dotjerivanja podataka - brisanja interpunkcijskih znakova, zagrada, skraćenica, naglasaka, pretvaranja velikih u mala slova i slično, ostali smo sa setom podataka oblika riječ : definicija. Na primjer,

preuzeti : uzeti što od koga, primiti što kao svoje, poprimiti, prihvatiti, pristati na koju dužnost ili obvezu koju tko daje ili ju je dotad imao, odgovornost pristati, biti odgovornim za što preuzimati

profil : lice gledano sa strane, pogled sa strane, uzdužni ili poprečni presjek čega, skup svojstava zajedničkih čemu

tama : stanje u kojemu nema svjetla ili ga nema dovoljno, mrak, tmina, svjetlo, svjetlost, stanje neznanja, zaostalosti

Ideja je za svaku riječ r_2 u definiciji neke riječi r_1, dodati brid između riječi r_1 i riječi r_2. Međutim riječi u definicijama ne javljaju se nužno u obliku u kakvom se javljaju u rječniku - primjerice imenice se javljaju u dativu, glagol u perfektu i slično. Ideja je svakoj rječi definicije, naći najbližu riječ u rječniku koja bi služila kao zamjena za r_2.

rjecnik.hr

7.3.1 Modificirana Levenshteinova udaljenost

Za riječ r_1 i riječ r_2, njihova Levenshteinova udaljenost je najmanji broj brisanja, zamjena ili dodavanja znakova u r_2 kako bi dobili r_1. Primjerice lako se vidi da riječi *heal* i *hello* imaju Levenshteinovu udaljenost 2

$$heal \implies hell \implies hello.$$

Implementacija Levenshteinovog algoritma temelji se na dinamičkom programiranju - ideja je razbiti problem na manje potprobleme promatranjem svih prefiksa obje riječi. Neka su riječi r_1 duljine m i r_2 duljine n. Uvodimo matricu Levenshteinove udaljenosti

$$D = (d_{ij})_{\substack{i=0,\dots,n \\ j=0,\dots,m}},$$

pri čemu za $i > 0$ i $j > 0$ d_{ij} predstavlja Levenshteinovu udaljenost između prefiksa $r_1[1:j]$ i $r_2[1:i]$. Vrijedi:

$$d_{ij} = \begin{cases} i, & j = 0 \\ j, & i = 0 \\ d_{i-1,j-1}, & i,j > 0,\ r_1[j] = r_2[i] \\ \min(d_{i-1,j-1}, d_{i-1,j}, d_{i,j-1}) + 1, & i,j > 0,\ r_1[j] \neq r_2[i] \end{cases}$$

Intuitivno, ako su $r_1[j] = r_2[i]$, tada je Levenshteinova udaljenost $r_1[1:j]$ i $r_2[1:i]$ jednaka Levenshteinovoj udaljenosti $r_1[1:j-1]$ i $r_2[1:i-1]$, inače je potrebno napraviti brisanje, dodavanje ili substituciju.

Prikažimo nekoliko koraka algoritma na riječima *hello* i *heal*.

		H	E	L	L	O
	0	1	2	3	4	5
H	1					
E	2					
A	3					
L	4					

$\Longrightarrow$

		H	E	L	L	O
	0	1	2	3	4	5
H	1	0				
E	2					
A	3					
L	4					

$\Longrightarrow$

		H	E	L	L	O
	0	1	2	3	4	5
H	1	0	1	2	3	4
E	2					
A	3					
L	4					

Na kraju dobivamo matricu

		H	E	L	L	O
	0	1	2	3	4	5
H	1	0	1	2	3	4
E	2	1	0	1	2	3
A	3	2	1	1	2	3
L	4	3	4	1	1	2

dobili smo da je Levenshteinova udaljenost zaista 2.

Iako bi Levenshteinova udaljenost mogla poslužiti kao dobra mjera udaljenosti između riječi u različitom obliku (padežu, vremenu...), htjeli bismo da naša udaljenost više penalizira promjene na sredini riječi nego na kraju riječi. Primjerice, riječ *majka* ima Levenshteinovu udaljenost 1 i u odnosnu na riječ *majke* i na riječ *marka*. Stoga uvodimo modificiranu Levenshteinovu matricu

$$H = (h_{ij})_{\substack{i=0,\dots,n \\ j=0,\dots,m}}$$

danu sa

$$h_{ij} = \begin{cases} i, & j = 0 \\ j, & i = 0 \\ h_{i-1,j-1}, & i,j > 0,\ r_1[j] = r_2[i] \\ \min(h_{i-1,j-1}, h_{i-1,j}, h_{i,j-1}) + 1 + \max(m-j, n-i), & i,j > 0,\ r_1[j] \neq r_2[i] \end{cases}$$

i modificiranu Levenshteinovu udaljenost sa h_{nm}. Primjerice, riječi *majka* i *marka* imaju modificiranu Levenshteinovu udaljenost 3, a *majka* i *majke* 1.

Radi ubrzavanja algoritma, iz rječnika i definicija izbacit ćemo sve riječi duljine strogo manje od 3. Nadalje , za svaku riječ r duljine m iz definicije, tražit ćemo najbližu riječ r_2 u rječniku, samo za one riječi za koje je

$$r_1[1 : m/_{\mathbb{Z}}2] = r_2[1 : m/_{\mathbb{Z}}2]$$

pri čemu je $m/_{\mathbb{Z}}2$ cjelobrojno djeljenje m sa 2.

Iako ovaj pristup postiže dobre rezultate, i dalje postoje situacije u kojima grješi. Primjerice najbliža riječ riječi *kakav* je *kakao* - iako je teško zamisliti udaljenost koja bi odvojila te dvije riječi i postigla jednako dobre rezultate.

7.3.2 Rezultati

Graf za hrvatski rječnik sastoji se od 29 564 vrhova i 235 831 bridova. Kao i prije, za postizanje boljih rezultata, odlučujemo se iz rječnika izbaciti 29 najdominantnijih riječi.

U hrvatskom jeziku odabiremo riječi: *tuga*, *mrak*, *obitelj*, *susret*, *gledati*, *slika*, *prilika*, *vrijeme*. Hrvatski rječnik je sadržavao manje riječi. Kao što možemo vidjeti neke od riječi koje smo gledali su slabo povezane te generiraju relativno mali podgraf. Unatoč tome za riječi *tuga*, *mrak*, *susret*, *gledati* algoritam predlaže zadovoljavajuće kandidate. No uočimo da za *mrak* i *tuga* najveću mjeru centralnosti dobivaju antonimi tj. riječ suprotnog značenja. Za *tuga*, najveću centralnost ima *sreća*, dok za *mrak* najveću centralnost ima *svjetlo*.

	Tuga		Mrak		Obitelj		Susret	
Broj vrhova s kojima je povezana	38		33		27		27	
Broj bridova podgrafa	44		32		24		24	
Ukupan broj iteracija	165		22		3438		39	
Vrijeme izvršavanja	0.3441958 secs		0.02372599 secs		2.683011 secs		0.03290009 secs	
1	sreća	0,24057593	svjetlo	0,3137714	ljudi	0,26950903	sastajati	0,3011279
2	žalost	0,23055504	tama	0,27547862	domaći	0,25805914	sastati	0,3011279
3	radost	0,21758294	tmina	0,27547862	kuća	0,21809564	kim	0,2521286
4	veselje	0,21758294	svjetlost	0,2299047	dom	0,17890375	sastanak	0,1900931
5	sjetan	0,14824488	stanje	0,08977762	ugao	0,13421238	dogovoriti	0,1107905
6	melankoličan	0,11407597	mračan	0,05743374	zajednica	0,11412546	izbjegavanje	0,1020279
7	nujan	0,11407597	osvanuti	0,0560396	srodstvo	0,1018599	dan	0,000000008121505
8	neveseo	0,10919959	mrkli	0,03600963	ljudski	0,08323165	večer	0,000000008121505
9	tužan	0,10514629	sumrak	0,0355612	živ	0,07947047	pozdrav	0,000000007362777
10	šteta	0,08596661	nada	0,0245152	pleme	0,05713183	zdravo	0,000000003802407

	Gledati		Slika		Prilika		Vrijeme	
Broj vrhova s kojima je povezana	38		124		36		305	
Broj bridova podgrafa	30		111		36		584	
Ukupan broj iteracija	49		114		4862		77	
Vrijeme izvršavanja	0.03336906 secs		0.7520061 secs		5.460975 secs		1.172618 secs	
1	pogled	0,30440125	pomoć	0,29654652	omogućiti	0,309432013	određeno	0,32339708
2	gle	0,27799832	prikaz	0,15729228	omogućavati	0,268601734	održavati	0,12031909
3	pogledati	0,2566659	slikanje	0,14167149	onemogućiti	0,268601734	trenutak	0,11334818
4	vid	0,11373414	slikati	0,12636418	onemogućavati	0,262130463	održati	0,10178478
5	vidjeti	0,0952272	alegorija	0,11331642	svjetlo	0,106093871	stizati	0,09463647
6	blago	0,08715055	ilustracija	0,10717786	svoj	0,065124045	događaj	0,09391237
7	čuđenje	0,08715055	prozirnica	0,10134083	vrijeme	0,044801642	čas	0,09377526
8	izričaj	0,08715055	dočaravati	0,10131631	sreća	0,036090359	stići	0,09271654
9	upozoravati	0,08715055	dočarati	0,09905637	stanje	0,031378752	tren	0,07920746
10	dužina	0,07146588	fotografiranje	0,09838788	pojava	0,007230815	unajmiti	0,07278664

8 Alternativni pristupi

8.1 Šire susjedstvo

U dosadašnjem smo se dijelu fokusirali na sličnost podgrafa danog kao susjedstvo promatranog vrha s vrhom jednog specifičnog grafa. Iako je takav pristup producirao solidne rezultate, nije savršen. Stoga ćemo pokušati dati alternativne pristupe koji bi u nekim aspektima mogli biti bolji od originalnog rezultata.

Prva takva ideja je da za sličnost možemo drugačije gledati susjedstvo pojedinog vrha. Naime,

umjesto da gledamo samo susjede promatranog vrha (tj. riječi u rječniku), možemo uključiti i susjede susjeda. Nažalost, taj pristup se neće ispostaviti dobrim jer ima dva velika problema.

Prvi problem koji se javlja je taj da će graf susjedstva enormno porasti. Samo u prijelazu s prvih na prve i druge susjede, dobivamo ogromnu razliku, graf postaje znatno veći i samim tim algoritam radi znatno sporije. Primjerice, ako promotrimo graf susjedstva (samo za neposredne susjede promatranog vrha) za vrh "rectangle" dobijemo sljedeći graf.

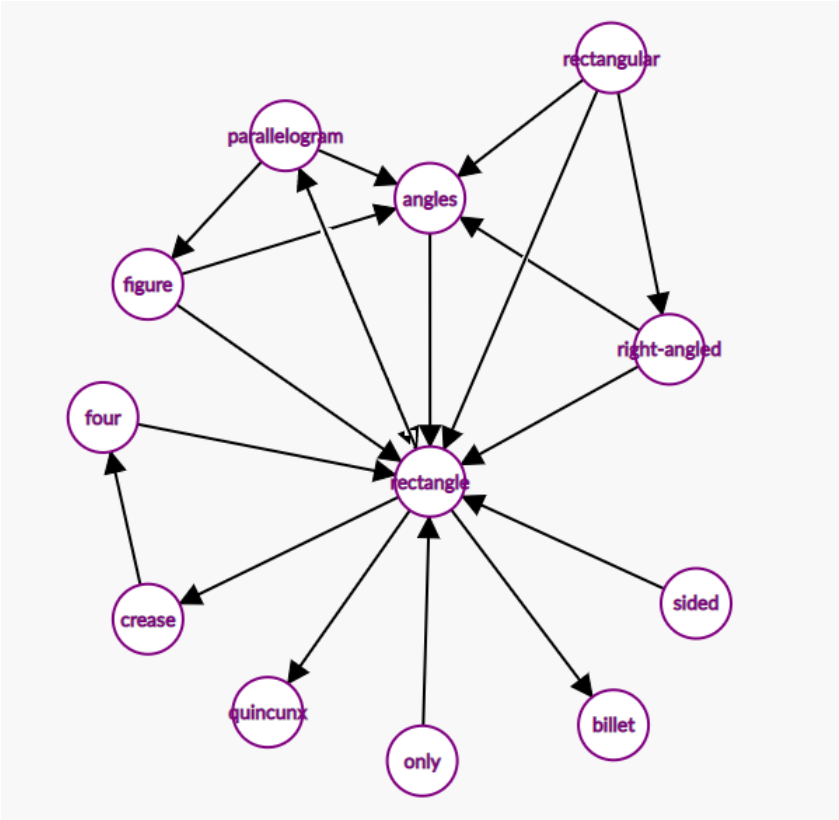


Figure 1: Graf susjedstva riječi "rectangle"

Za usporedbu, kada u podgraf uključimo i susjede susjeda, dobijemo podgraf s 2 129 vrhova i 11 487 bridova. Dakle, pristup koji će dopuštati više od neposrednih susjeda neće biti traktabilan.

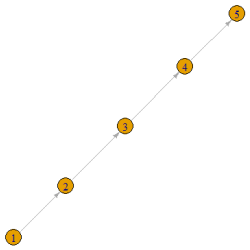
Drugi veliki problem pristupa s većom dubinom je taj da, iako smo u algoritmu izbacili riječi koje se javljaju u puno definicija, i dalje postoje riječi sa širokim spektrom značenja te će biti pristune kod mnogih drugih iako suštinski značenjem nisu povezane. To rezultira time da u podgrafu te jako povezane riječi preuzimaju ulogu centra i stoga će mjera centralnosti dobivena algoritmom vratiti upravo te riječi umjesto boljih kandidata za sinonime. Primjer s riječi "hot" možemo vidjeti niže.

```
> Sims("hot",edges,FF,index, Banned, sum_ind=sum_ind, d = 1)
[1] hot
111988 Levels: a a-mornings a-sea a-tiptoe aam aard-vark aard-wolf aaronic aaronical ab abaca abaci ... zythum
[1] "Povezan sa 164 vrhova"
[1] "Broj bridova podgrafa: 253"
[1] "Ukupno 22 iteracija"
Time difference of 0.2263939 secs
                [,1]
warm        0.20832955
cool        0.15643253
heat        0.12001069
cold        0.11900457
vehement    0.11417490
ardent      0.11262284
boil        0.10775966
warmth      0.10586266
calefactory 0.10331872
passionate  0.09587411
> Sims("hot",edges,FF,index, Banned, sum_ind=sum_ind, d = 2)
[1] hot
111988 Levels: a a-mornings a-sea a-tiptoe aam aard-vark aard-wolf aaronic aaronical ab abaca abaci ... zythum
[1] "Povezan sa 7326 vrhova"
[1] "Broj bridova podgrafa: 80256"
[1] "Ukupno 15 iteracija"
Time difference of 16.91461 secs
            [,1]
run    0.11037438
turn   0.06096032
strike 0.05930315
draw   0.05479328
lead   0.05302048
rise   0.05102081
touch  0.04466219
fall   0.04432333
pass   0.04318259
roll   0.04285972
```

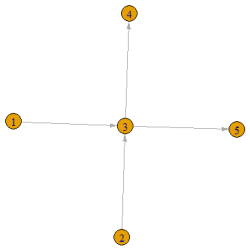
Figure 2: Usporedba outputa algoritma za podgrafove susjedstva dubine 1 i 2

8.2 Sličnost s drugim grafovima

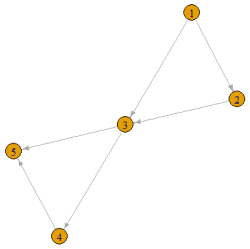
Druga promjena koju možemo napraviti u modelu je da gledamo sličnost s vrhovima u nekom drugom grafu. I ovdje će biti ideja da mjerimo centralnost promatranog vrha u svom susjedstvu, pa je prema tome logično gledati grafove koji imaju strukturu koja će uključivati neki vrh koji bi se trebao isticati kao svojevrsni centar. Primjerice, grafovi s kojima ćemo pokušati poboljšati algoritam mogu biti:



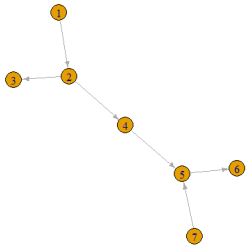
(a) Graf 1



(b) Graf 2



(c) Graf 3



(d) Graf 4

Figure 3: Primjeri grafova korištenih za algoritam

Za usporedbu svih varijanti (kombinacije grafovi 1-4 i početni graf, te dubina 1 ili 2) imamo sljedeću tablicu kandidata za sinonime.

Graf (Dubina) / Riječ	Osnovni (1)	1 (1)	2 (1)	3 (1)	4 (1)	Osnovni (2)	1 (2)	2 (2)	3 (2)	4 (2)
Ask	Demand	Demand	Demand	Demand	Asking	Run	Run	Run	Before	Before
	Pray	Crave	Pray	Question	Question	Turn	Turn	Turn	Do	He
	Crave	Pray	Crave	Request	Seek	Draw	Draw	Draw	Go	What
Begin	Initiate	Initiate	Initiate	Beginning	Beginning	Run	Run	Run	Turn	Move
	Rise	Rise	Rise	New	Originate	Turn	Turn	Turn	Run	Course
	Initiative	Enter	Initiative	Initiate	New	Lead	Lead	Lead	Pass	Position
Story	Fable	Fable	Fable	Event	Event	Run	Run	Run	Turn	Particular
	Tale	Tale	Tale	Narrative	Tell	Turn	Turn	Turn	Run	Subject
	Relation	Relation	Relation	Narration	Particular	Strike	Strike	Strike	Pass	He

Figure 4: Usporedba outputa algoritma u raznim varijantama

Prije svega, uočavamo da je output u većini slučajeva jednak (ili se razlikuje na jednom mjestu), osim eventualno za graf 4. Četvrti graf, iako daje različite rezultate, ne bismo rekli da su ti rezultati bolji.

Dakle, konačni zaključak ovog poglavlja je da je pristup odabran u originalnom radu zaista dobar iz nekoliko aspekata - brzina, jednostavnost, jasnoća interpretacije i možda najbitnije, kvaliteta rezultata.

9 Literatura

1. V.D. BLONDEL, A. GAJARDO, M. HEYMANS, P. SENELLART, P. VAN DOOREN *A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching*

2. HRVATKI RJEČNIK `https://rjecnik.hr/`

3. B.W. PALEO *Levenshtein Distance: Two Applications in Database Record Linkage and Natural Language Processing*

4. R.A. HORN & C.R. JOHNSON *Matrix Analysis*