

NLP 8

Spracovanie prirodzeného jazyka

Peter Bednár

Latentné reprezentácie jazyka a modelovanie tém

Latentné Sémantické Indexovanie

- Rozklad term-dokument matice A na singulárne vektory a čísla
- Redukcia matice A na k hlavných latentných dimenzií

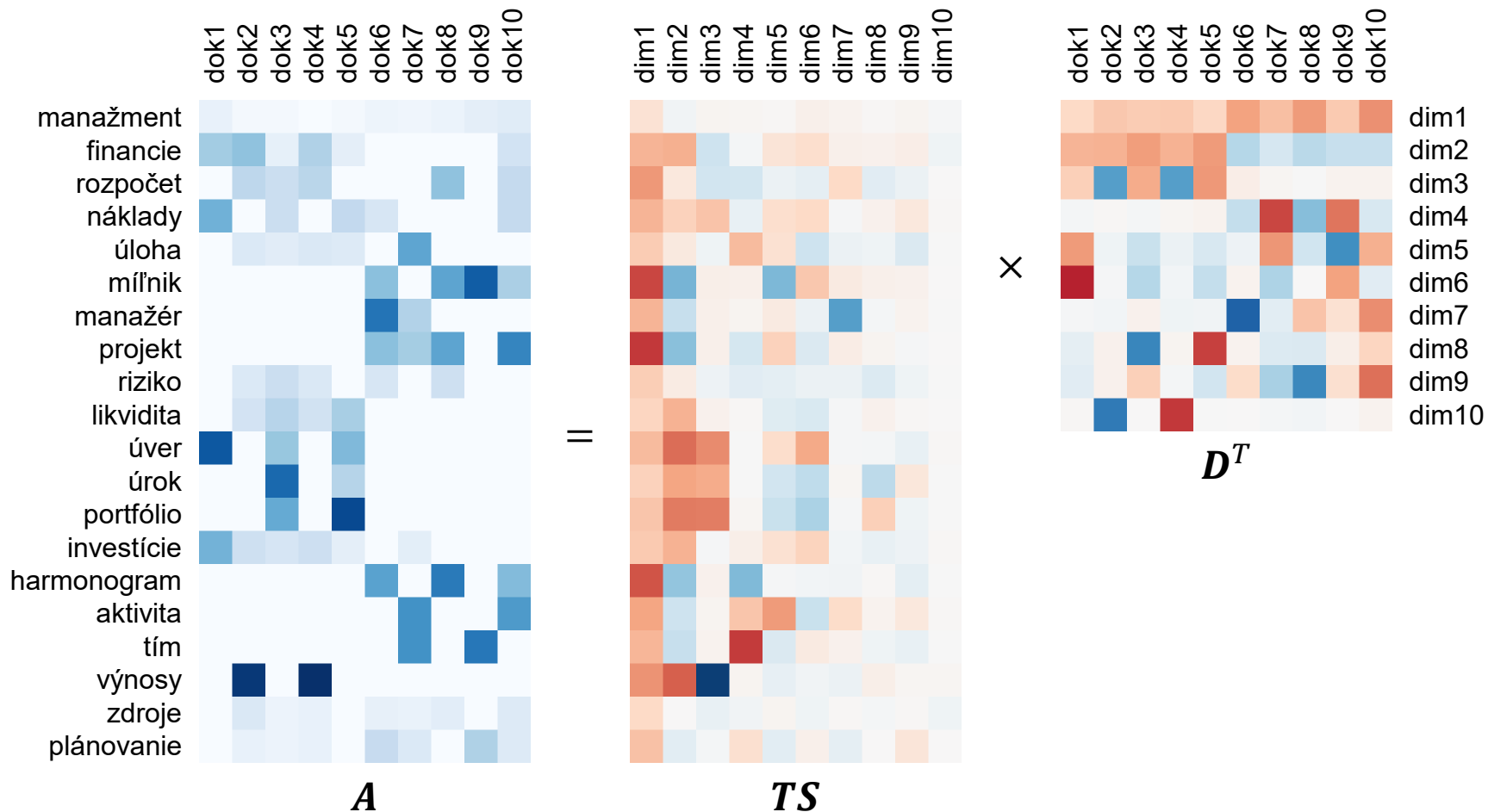
$$A \approx A_k = T_k S_k D_k^T$$

kde T_k mapuje termy na latentné dimenzie,

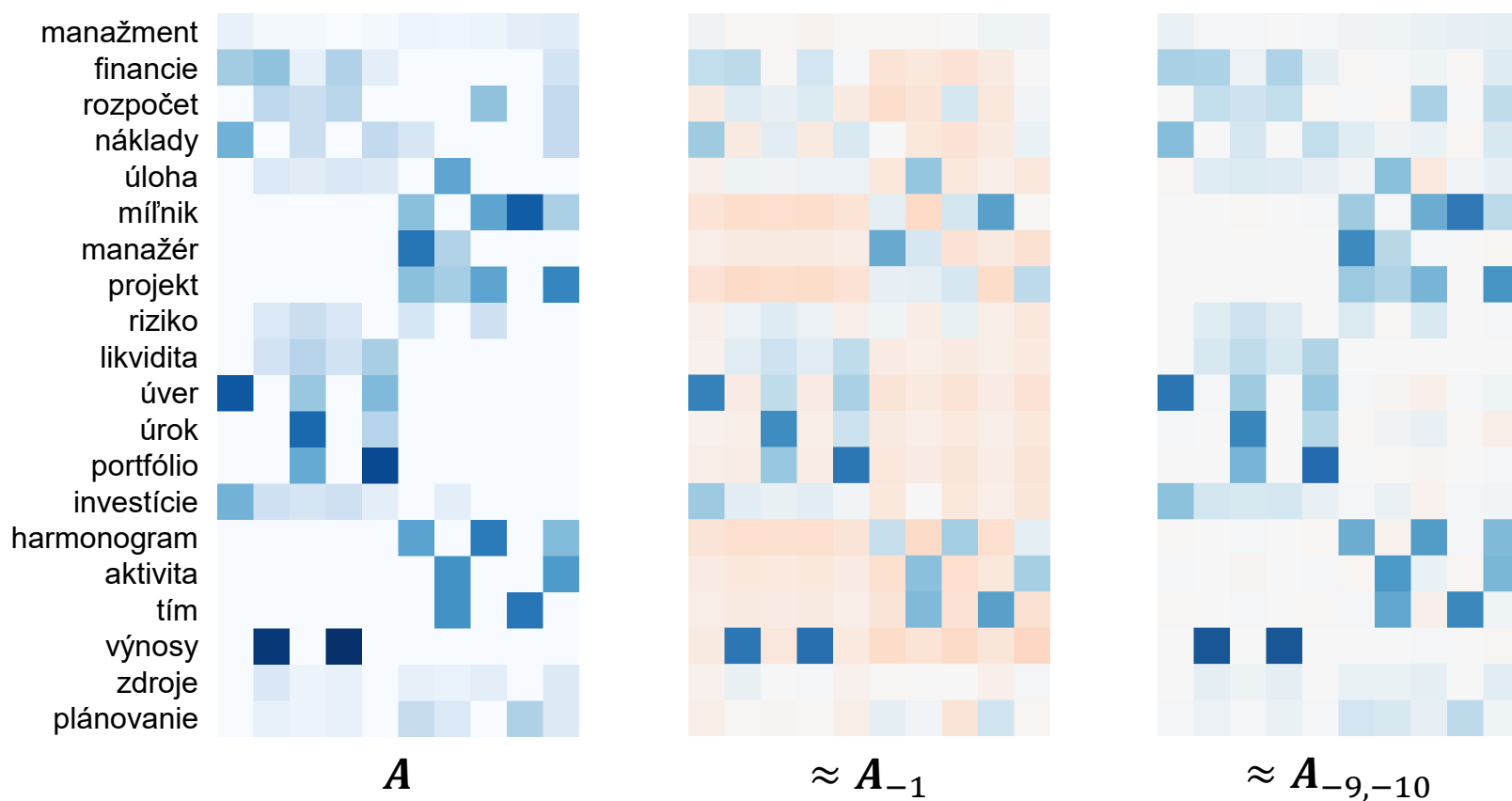
S_k je diagonálna matica, ktorej prvky $s_1 \geq s_2 \geq \dots \geq s_k$ vyjadrujú dôležitosť jednotlivých latentných dimenzií pre celú množinu dokumentov; a

D_k^T mapuje dokumenty na latentné dimenzie

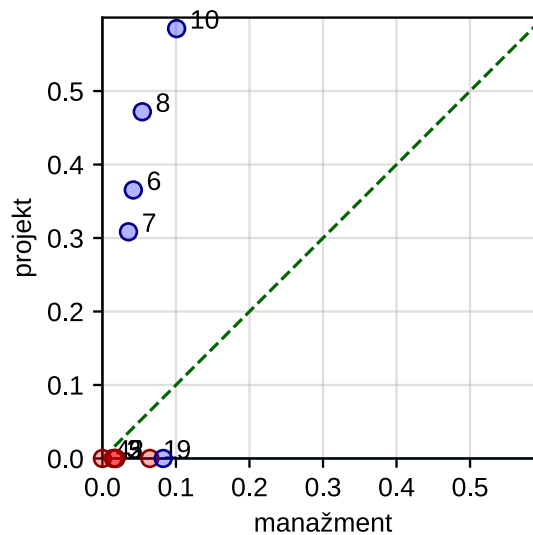
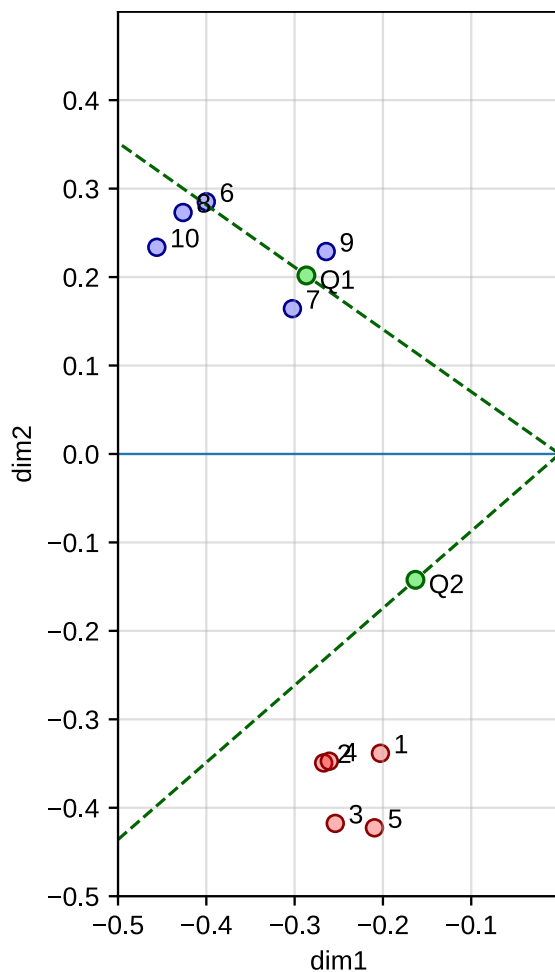
LSI – príklad (1)



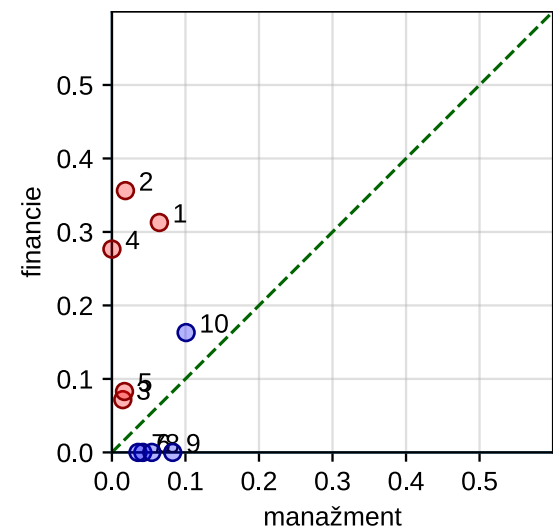
LSI – príklad (2)



LSI – príklad (3)



Q1: manažment + projekt
Q2: manažment + financie



Modelovanie tém

- Nedostatky LSI pre modelovanie tém
 - latentné dimenzie nemusia mať priamo tematickú interpretáciu
 - mapovanie $\mathbf{T}_k$ a $\mathbf{D}_k^T$ môže obsahovať aj kladné aj záporné hodnoty
- Pravdepodobnostný model tém
 - priradenie medzi témami / dokumentami a medzi témami / termami je charakterizované pravdepodobnosťou
 - dokument $d \rightarrow$ distribúcia tém $\theta_d = P(k | d)$
 - téma $k \rightarrow$ distribúcia termov $\phi_k = P(t | k)$

Latentná Dirichletova Alokácia

- Dirichletove rozdelenie

- parameter $\alpha < 1$ - čím menšia hodnota, tým je pravdepodobnosť sústredená na menší počet hodnôt (pre $\alpha = 1$ dostaneme uniformné rozdelenie)

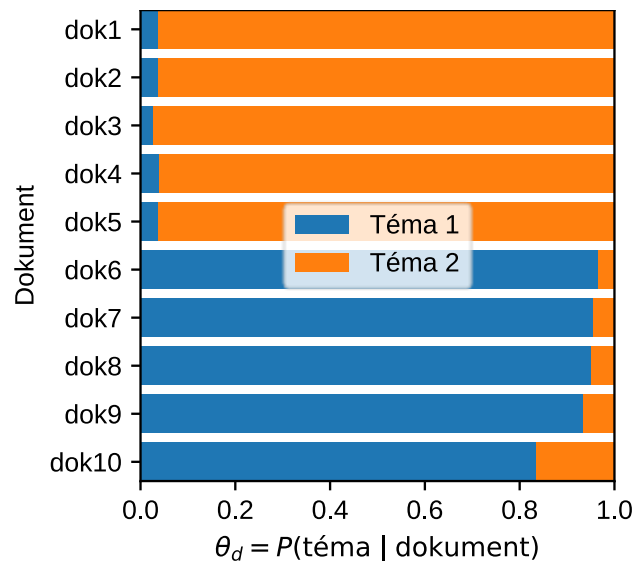
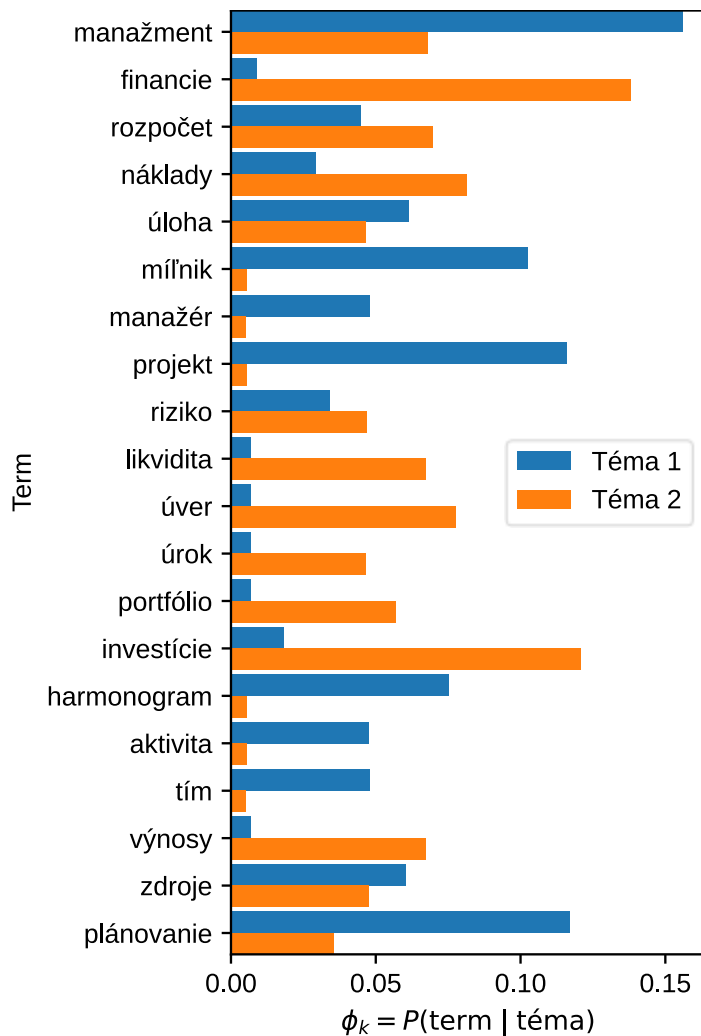
- Pri modelovaní tém:

- dokument obsahuje prevažne iba niekoľko tém, napr. $\theta_d = (0.55, 0.02, 0.03, 0.35)$
- podobne každá téma má iba niekoľko kľúčových termov

Latentná Dirichletova Alokácia

- Vstup:
 - term-dokument matica s frekvenciami výskytov termov
 - K – počet vyextrahovaných latentných tém
- Výstup:
 - per každý dokument distribúcia tém θ_d
 - per každú tému distribúcia termov ϕ_k

LDA – príklad



Porovnanie LSI a LSA

LSI

algebraický model

latentné dimenzie

dokument – vektor v latentnom priestore

term – vektor v latentnom priestore

vyhľadávanie informácií,
redukcia príznakového priestoru

LSA

pravdepodobnostný generatívny model

témy

dokument – distribúcia tém

téma – distribúcia termov

modelovanie tém

obidve metódy nezohľadňujú poradie slov v dokumente

Aplikácie spracovania prirodzeného jazyka

Klasifikácia textu

- Úlohou je priradiť textu (napr. vete, dokumentu, e-mailu, recenzii alebo príspevku) jednu alebo viac vopred definovaných tried
- Typické aplikácie
 - klasifikácia tém, kategorizácia dokumentov, analýza sentimentu, detekcia spamu, detekcia toxického obsahu, identifikácia jazyka, rozpoznávanie zámeru

Typy úloh klasifikácie textu

- Binárna
 - filtrovanie dokumentov, napr. je spam / nie je spam
- Viac tried
 - napr. politika / šport / veda / ekonomika
- Viacnásobná
 - napr. jeden dokument $\rightarrow$ {AI, medicína, etika}
- Hierarchická
 - kategorizácia dokumentov podľa tezaura, napr. veda $\rightarrow$ informatika $\rightarrow$ umelá inteligencia
- Ordinálna
 - napr. negatívne / neutrálne / pozitívne, hodnotenie 1–5

Metódy klasifikácie textu (1)

- Pravidlové metódy
 - založené na slovníkoch, regulárnych výrazoch a expertných pravidlách
 - nevyžadujú trénovacie dáta, priamo interpretovateľné
- Strojové učenie
 - frekvenčná reprezentácia textu
 - *Bag of Words* / TF-IDF + n-gramy
 - mnohorozmerná riedka reprezentácia: logistická regresia a SVM
 - výpočtovo aj dátovo nenáročné, dobrá východisková metóda

Metódy klasifikácie textu (2)

- Strojové učenie + distribuovaná reprezentácia slov
 - pred-učené statické *embedding* vektory slov (Word2Vec, GloVe)
 - *embedding* vektory slov $\rightarrow$ *pooling* pre celý text $\rightarrow$ klasifikátor
 - lepšie sa zachytáva význam jednotlivých slov
- Pooling pre postupnosť *embedding* vektorov slov v texte $x_1, x_2, \dots, x_n$:
 - suma $\sum_i x_i$
 - priemer $\frac{1}{n} \sum_i x_i$
 - max po elementoch
 - TF-IDF váhovanie $\frac{\sum_i w_i x_i}{\sum_i w_i}$, kde w_i je váha i -tého slova podľa TF-IDF

Metódy klasifikácie textu (3)

- Hlboké učenie
- Konvolučné siete
 - *embedding* vektory slov → 1D konvolučná sieť → pooling pre celý text → klasifikácia
 - poradie slov a kontext sú zachované iba čiastočne, zachytia sa frázy a lokálne vzory na úrovni n-gramov
- Kontextová reprezentácia
 - klasifikácia sekvencií
 - využitie rekurentných sietí (LSTM, GRU), alebo architektúry Transformer
 - *embedding* vektory slov → kodér → kontextová reprezentácia → klasifikácia

Vyhodnotenie klasifikácie textu

- Základné metriky

- presnosť P , návratnosť R a F_1 miera pre jednotlivé triedy
- výpočet z kontingenčnej tabuľky:

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}, F_1 = \frac{2PR}{P+R}$$

- Makro-spriemerovanie

- priemer z hodnôt presnosti, návratnosti a F_1 pre jednotlivé triedy
- všetky triedy sú započítané s rovnakou váhou (aj minoritné triedy pri nevybalancovaných dátach)

- Mikro-spriemerovanie

- výpočet z kumulatívnej kontingenčnej tabuľky s TP , FP , TN , FN spočítanými pre všetky triedy

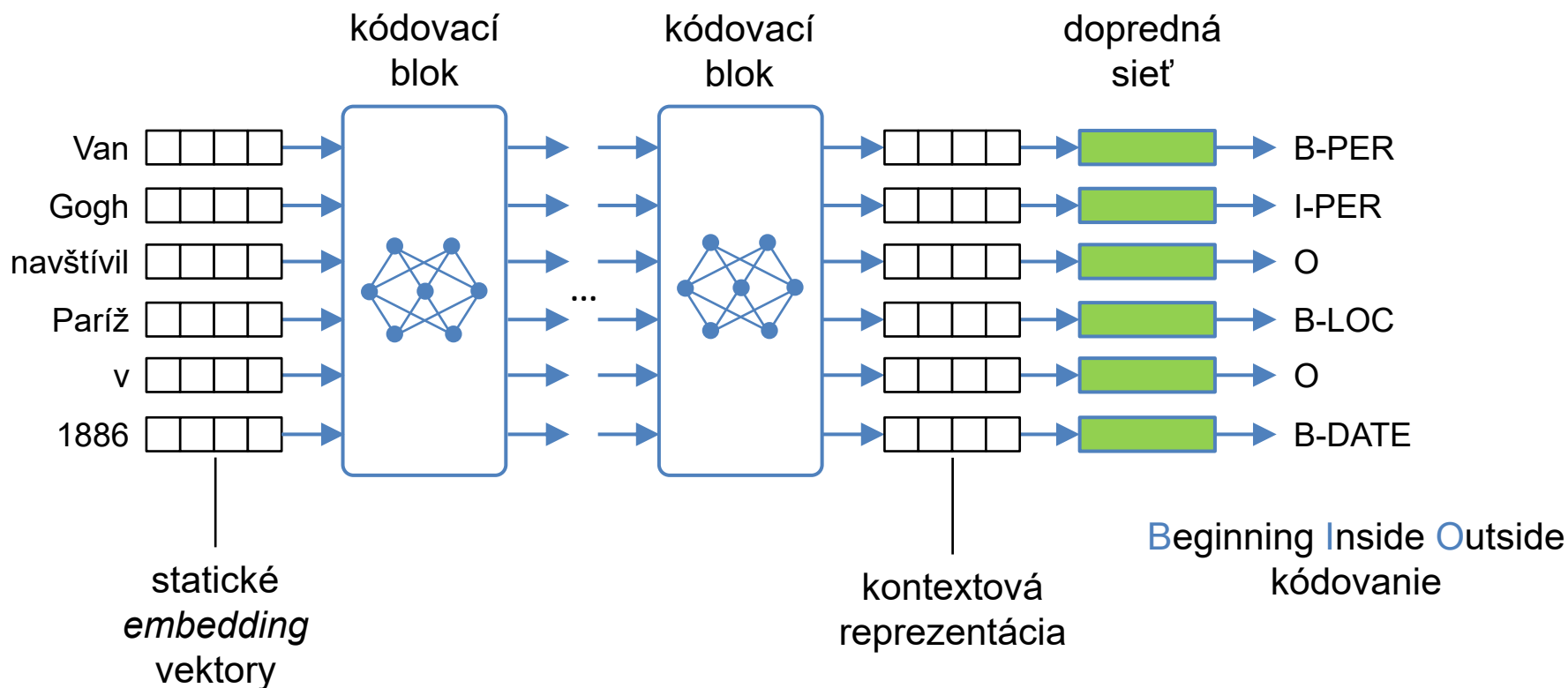
Extrahovanie informácií

- Rozpoznávanie pomenovaných entít (NER)
 - mestá, osoby, organizácie, udalosti ... produkty ... diagnózy, chemické zlúčeniny, lieky, gény, proteíny ...
- Rozlíšenie koreferencií
 - ktoré výrazy v texte (názvy, skratky, zámená) označujú rovnakú identitu
- Extrakcia vzťahov medzi entitami

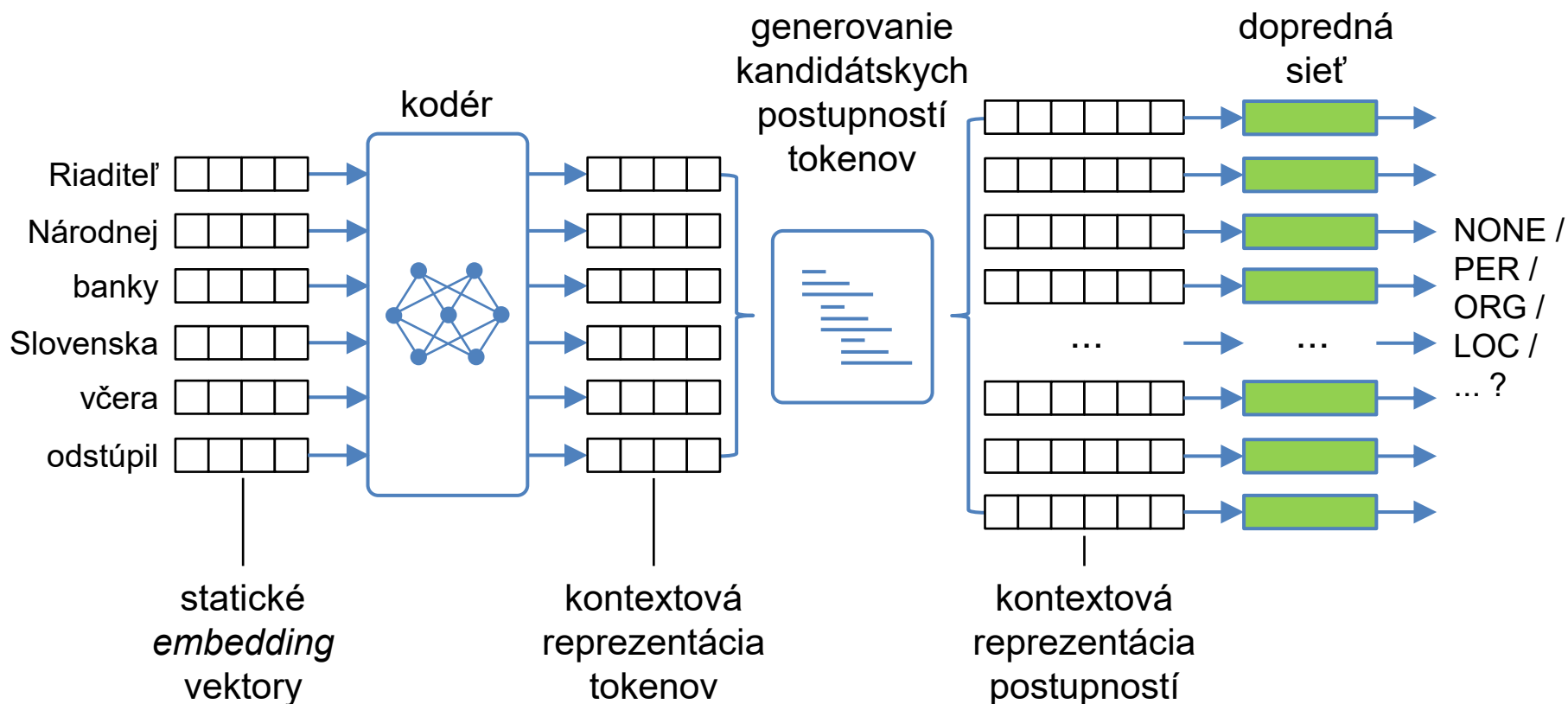
Rozpoznávanie pomenovaných entít

- **Neprekrývajúce sa entity**
 - BIO (*beginning, inside, outside*) označenie pre každý typ entity (napr. B-PER, I-PER, B-LOC, I-LOC, ..., O)
 - mapovanie sekvencií s rovnakou dĺžkou
- **Prekrývajúce sa entity**
 - napr. Študujem na [Technickej Univerzite v [Košiciach]_{LOC}]_{ORG}
 - 1. generovanie kandidátskych postupností tokenov (*spanov*) *s*
 - 2. klasifikácia či *s* označuje entitu daného typu

Rozpoznávanie neprekrývajúcich entít



Rozpoznávanie prekrývajúcich entít



Generovanie postupností tokenov

- Ohraničenie maximálnej dĺžky
 - pre text dlhý n tokenov je možné celkovo vygenerovať $\frac{n(n-1)}{2}$ postupností tokenov neohraničenej dĺžky
 - pre maximálnu dĺžku k je počet postupností iba $kn - \frac{k(k-1)}{2}$

Generovanie postupností tokenov

Riaditeľ Národnej banky Slovenska včera odstúpil

$n = 6$ pre $k = 3$, počet postupností $3.6 - \frac{3(3-1)}{2} = 15$

Postupnosti:

Riaditeľ	Slovenska
Riaditeľ Národnej	Slovenska včera
Riaditeľ Národnej banky	Slovenska včera odstúpil
Národnej	včera
Národnej banky	včera odstúpil
Národnej banky Slovenska	odstúpil
banky	
banky Slovenska	
banky Slovenska včera	

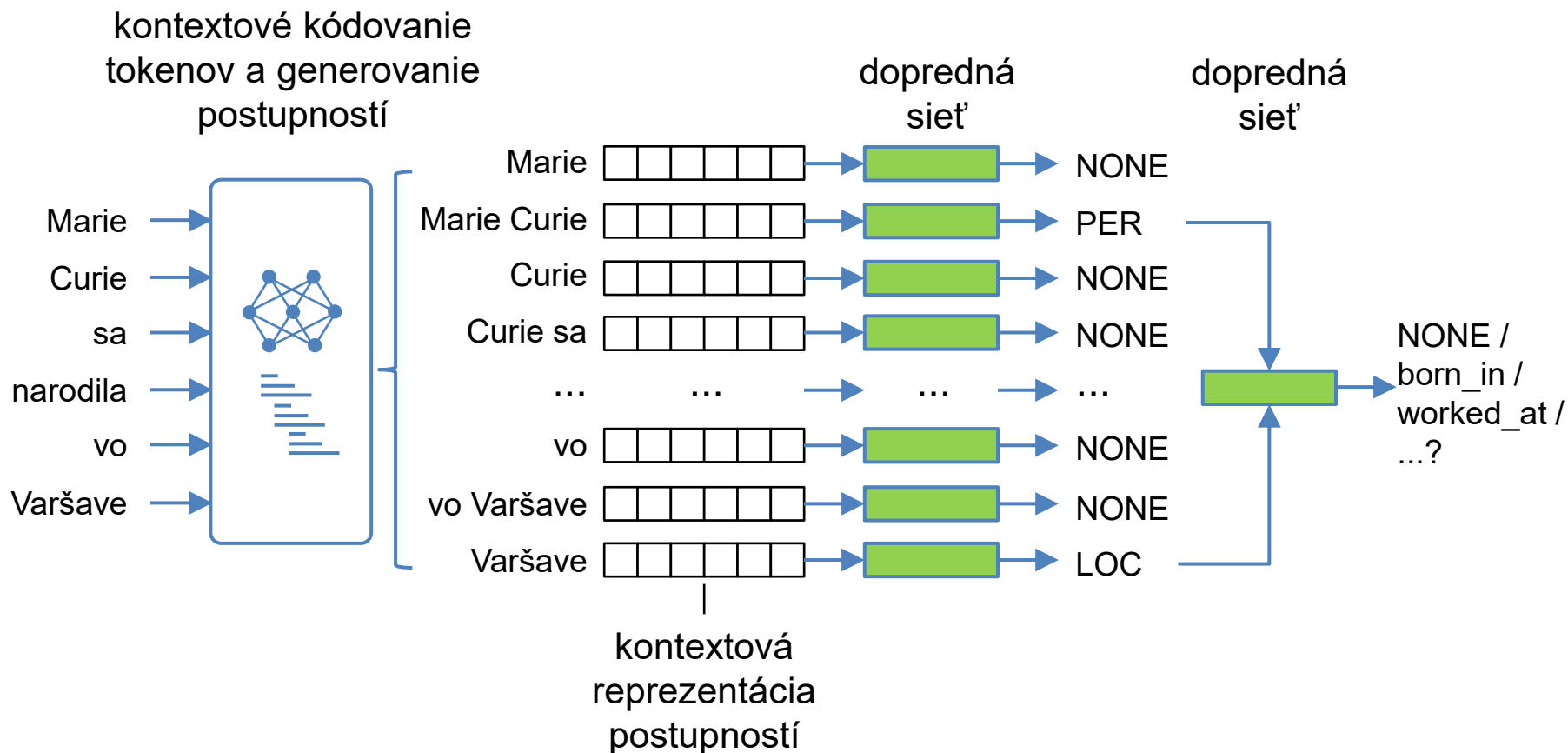
Kontextová reprezentácia postupností tokenov

- Pre postupnosť $s_{i,j}$ máme kontextové reprezentácie tokenov $\mathbf{h}_i, \mathbf{h}_{i+1}, \dots, \mathbf{h}_{j-1}, \mathbf{h}_j$
- Jednoduchá reprezentácia
 - $\mathbf{z}_{i,j} = [\mathbf{h}_i; \mathbf{h}_j]$ - zahŕňa iba reprezentáciu prvého a posledné tokenu
- Rozšírená reprezentácia
 - $\mathbf{z}_{i,j} = [\mathbf{h}_i; \mathbf{h}_j; p(\mathbf{h}_i, \dots, \mathbf{h}_j); \mathbf{e}_{j-i+1}]$ kde:
 - $\mathbf{h}_i; \mathbf{h}_j$ reprezentuje začiatok a koniec entity,
 - $p(\mathbf{h}_i, \dots, \mathbf{h}_j)$ vnútorný obsah (*pooling* vektorov $\mathbf{h}_i, \dots, \mathbf{h}_j$) a
 - $\mathbf{e}_{j-i+1}$ *embedding* kódujúci dĺžku entity (z matice pre dĺžku 1, 2, ..., max. dĺžka k)

Extrahovanie relácií medzi entitami

- Kontextová reprezentácia entít – postupnosti tokenov
 - pre každú kandidátsku dvojicu entít s_1 a s_2 s kontextovou reprezentáciou z_1 a z_2 sa klasifikuje, či medzi nimi existuje relácia daného typu
- Jeden model pre rozpoznávanie entít a extrahovanie relácií
 - *embedding* vektory slov $\rightarrow$ kodér $\rightarrow$ kontextová reprezentácia tokenov $\rightarrow$ generovanie kandidátskych postupností tokenov $\rightarrow$ kontextová reprezentácia postupností $\rightarrow$ klasifikácia entít $\rightarrow$ klasifikácia relácií

Rozpoznávanie entít a extrahovanie relácií



Rozlíšenie koreferencií

1. Vyextrahujú sa zmienky v texte, ktoré môžu odkazovať na entity
2. Určia sa klastre zmienok, ktoré odkazujú na tú istú entitu

- **Zmienka** môže byť:
 - pomenovaná entita (názov, pomenovanie, skratka, atď.),
 - zámeno
 - zamlčaný vetný člen
 - napr.

Ministerka predstavila poslancom návrh rozpočtu. _[ona] Následne ich požiadala o jeho podporu.

$C_1 = \{\text{Ministerka}, [\text{ona}]\}$, $C_2 = \{\text{poslancom}, \text{ich}\}$,

$C_3 = \{\text{návrh rozpočtu}, \text{jeho}\}$, $C_4 = \{\text{podporu}\}$

Rozlíšenie koreferencií

- Klasifikačná úloha
 - využíva sa kontextová reprezentácia postupností (*spanov*) vyextrahovaných zmienok
- Základná metóda
 - postupné spájanie s predchádzajúcimi zmienkami
 - pre každú zmienku sa klasifikuje do ktorého predchádzajúceho klastra má byť zaradená, alebo či sa má pre ňu vytvoriť nový klaster

Rozlíšenie koreferencií

- Zjednodušená reprezentácia klastrov
 - pre každú zmienku m_i sa určí množina kandidátov z predchádzajúcich zmienok $A_i = \{a_1, \dots, a_{i-1}, \emptyset\}$, kde $\emptyset$ označuje prázdneho kandidáta, pri ktorom sa má pre m_i vytvoriť nový klaster
 - dopredná sieť určí pre každú dvojicu skóre $s(m_i, a)$ a zaradí zmienku do klastra s najväčším skóre $\operatorname{argmax}_{a \in A_i} s(m_i, a)$

Rozlíšenie koreferencií – príklad

Ministerka predstavila poslancom návrh rozpočtu. _[ona] Následne ich požiadala o jeho podporu.

	∅	Ministerka	poslancom	návrh rozpočtu	[ona]	ich	jeho	
Ministerka								$C_1 = \{\text{Ministerka}\}$
poslancom								$C_2 = \{\text{poslancom}\}$
návrh rozpočtu								$C_3 = \{\text{návrh rozpočtu}\}$
[ona]								$C_1 = \{\text{Ministerka}, [\text{ona}]\}$
ich								$C_2 = \{\text{poslancom}, \text{ich}\}$
jeho								$C_3 = \{\text{návrh rozpočtu}, \text{jeho}\}$
podporu								$C_4 = \{\text{podporu}\}$

Vyhodnotenie extrahovania informácií

Rozpoznávanie pomenovaných entí

presnosť, návratnosť, F_1 miera	samostatne pre všetky typy entít, mikro spriemerovanie pre celkové hodnotenie
--------------------------------------	--

Extrahovanie vzťahov

presnosť, návratnosť, F_1 miera	vyhodnotenie na úrovni tripletov (správne určenie entít pre subjekt a objekt aj označenia relácie/predikátu)
--------------------------------------	--

Vyhodnotenie extrahovania informácií

Rozlíšenie koreferencií

- *MUC* metrika

- vypočíta sa ako F_1 z presnosti P_{MUC} a návratnosti R_{MUC} , kde
 - P_{MUC} penalizuje nesprávne spojenie $g(p)$ klastrov, ktoré neodkazujú na tú istú entitu, do jedného klastra; a
 - R_{MUC} penalizuje nesprávne rozdelenie jedného klastra na $p(g)$ klastrov

$$P_{MUC} = \frac{\sum_{p \in P} (|p| - g(p))}{\sum_{p \in P} (|p| - 1)} \quad R_{MUC} = \frac{\sum_{g \in G} (|g| - p(g))}{\sum_{g \in G} (|g| - 1)}$$

Vyhodnotenie extrahovania informácií

MUC príklad

$$P_{MUC} = \frac{\sum_{p \in P} (|p| - g(p))}{\sum_{p \in P} (|p| - 1)} \quad R_{MUC} = \frac{\sum_{g \in G} (|g| - p(g))}{\sum_{g \in G} (|g| - 1)}$$

referenčné klastre $G_1 = \{a, b, c, d\}, G_2 = \{e, f\}$

predikované klastre $P_1 = \{a, b, e\}, P_2 = \{c, d\}, P_3 = \{f\}$

$$P_{MUC} = \frac{(3-2)+(2-1)+(1-1)}{(3-1)+(2-1)+(1-1)} = \frac{2}{3} \approx 0.67 \quad R_{MUC} = \frac{(4-2)+(2-2)}{(4-1)+(2-1)} = \frac{2}{4} = 0.5$$

Vyhodnotenie extrahovania informácií

- B^3 metrika

- vypočíta sa ako F_1 z presnosti a návratnosti vypočítaných ako priemer cez všetky zmienky i podľa:

$$P_{B^3} = \frac{1}{N} \sum_{i=1}^N \frac{|G(i) \cap P(i)|}{|P(i)|} \quad R_{B^3} = \frac{1}{N} \sum_{i=1}^N \frac{|G(i) \cap P(i)|}{|G(i)|}$$

B^3 príklad

referenčné klastre $G_1 = \{a, b, c, d\}, G_2 = \{e, f\}$

predikované klastre $P_1 = \{a, b, e\}, P_2 = \{c, d\}, P_3 = \{f\}$

$$P_{B^3} = \frac{\frac{2}{3} + \frac{2}{3} + \frac{2}{2} + \frac{2}{2} + \frac{1}{3} + \frac{1}{1}}{6} = \frac{14}{3} \approx 0.778$$

$$R_{B^3} = \frac{\frac{2}{4} + \frac{2}{4} + \frac{2}{4} + \frac{2}{4} + \frac{1}{2} + \frac{1}{2}}{6} = \frac{3}{6} = 0.5$$

Vyhodnotenie extrahovania informácií

- *CEAF* metrika

- určí sa optimálne zarovnanie predikovaných a referenčných klastrov podľa vzdialenosti $\phi(G, P) = \frac{2|G \cap P|}{|G| + |P|}$, kde každý referenčný klaster môže byť priradený iba k jednému predikovanému klastru
- vypočíta sa ako F_1 s presnosťou $P_{CEAF} = \frac{\Phi}{N_P}$ a návratnosťou $R_{CEAF} = \frac{\Phi}{N_G}$ kde Φ je celkový súčet vzdialeností pre optimálne zarovnanie klastrov, N_P počet predikovaných klastrov a N_G počet referenčných klastrov

Vyhodnotenie extrahovania informácií

CEAF príklad

$$\phi(G, P) = \frac{2|G \cap P|}{|G| + |P|} \quad P_{CEAF} = \frac{\Phi}{N_P} \quad R_{CEAF} = \frac{\Phi}{N_G}$$

referenčné klastre $G_1 = \{a, b, c, d\}, G_2 = \{e, f\}$

predikované klastre $P_1 = \{a, b, e\}, P_2 = \{c, d\}, P_3 = \{f\}$

	P_1	P_2	P_3
G_1	$\frac{2.2}{4+3} \approx 0.571$	$\frac{2.2}{4+2} \approx \mathbf{0.667}$	$\frac{2.0}{4+1} = 0$
G_2	$\frac{2.1}{2+3} = 0.4$	$\frac{2.0}{2+2} = 0$	$\frac{2.1}{2+1} \approx \mathbf{0.667}$

$$\Phi = \frac{2}{3} + \frac{2}{3} = \frac{4}{3}$$

$$P_{CEAF} = \frac{4/3}{3} \approx 0.444$$

$$R_{CEAF} = \frac{4/3}{2} \approx 0.667$$

optimálne zarovnanie: $G_1 \leftrightarrow P_2, G_2 \leftrightarrow P_3$

Vyhodnotenie extrahovania informácií

- *CoNLL* F_1 metrika

- celkové vyhodnotenie, ktoré kombinuje MUC , B^3 a $CEAF$ metriku

$$CoNLL\ F_1 = \frac{F_1^{MUC} + F_1^{B^3} + F_1^{CEAF}}{3}$$