Hľadanie podobností v textoch ľudových piesní

Michal Mižák

Vedúci práce: doc. RNDr. Stanislav Krajči PhD

Oživenie pamäti

- Vložím/ nájdem pieseň alebo úryvok textu
- Dostanem zoznam podobných piesní resp. piesní, ktoré v nejakej forme úryvok obsahujú
- Štatistické vyhodnotenie podobnosti
- „Mergovací“ systém
- API jednoducho využiteľné v iných aplikáciách

Využitie

- Od dediny ku dedine, od domu k domu pretvorený/ skomolený text
- Pri folklórnom výskume sa väčšinou piesne získavajú od starších ľudí – spievajú tak, ako si spomenú
  - Síce autentickosť, ale pamätníci to môžu opraviť
- Pri rozumnej databáze porovnávanie regionálnych rozdielov v piesňach
- Priestor pre odborníkov a historikov na „skonzistentnenie“ záznamov o piesňach
  - Nesprávny zápis hlások v nárečí

Využitie

- [: **Dzifče** počarovne, **ňezaľup** śe do mňe :]
  [: ja chlopec vandrovni, co ci budze zo mňe :]

  [: Ja chlapec vandrovni, zo šireho poľa :]
  [: co ci budze zo mňe, frajirečko moja? :]

- [: **Dzifče** počarovne, **nezaľub** se do mne :]
  [: ja chlapec vandrovny, co ci budze zo mňe :]

  [: Ja chlapec vandrovny, zo šireho poľa :]
  [: co ci budze zo mne, frajirečko moja? :]

- [: **Dzivče** počarovne, **nepopatraj** na mne :]
  [: ja chlapec vandrovny, co ci budze zo mne :]

  [: Ja chlapec vandrovný zo šireho poľa :]
  [: co ci budze zo mne draha duša moja? :]

Ako?

- 4 hlavné časti
  - Príprava databázy
  - Analýza frekvencie slov v konkrétnom tvare
  - Zjednocovanie rôznych tvarov slova do jedného
  - Technologická implementácia UI

Príprava databázy

- Scraping a rôzne formy skriptovania
  - Dáta bolo treba niekde zohnať
    - Viktor Gliganič - primáš (huslista) Ľudovej hudby FS Zemplín
    - Piesne372
    - Iné zdroje v stave ToDo

Budeme ignoranti

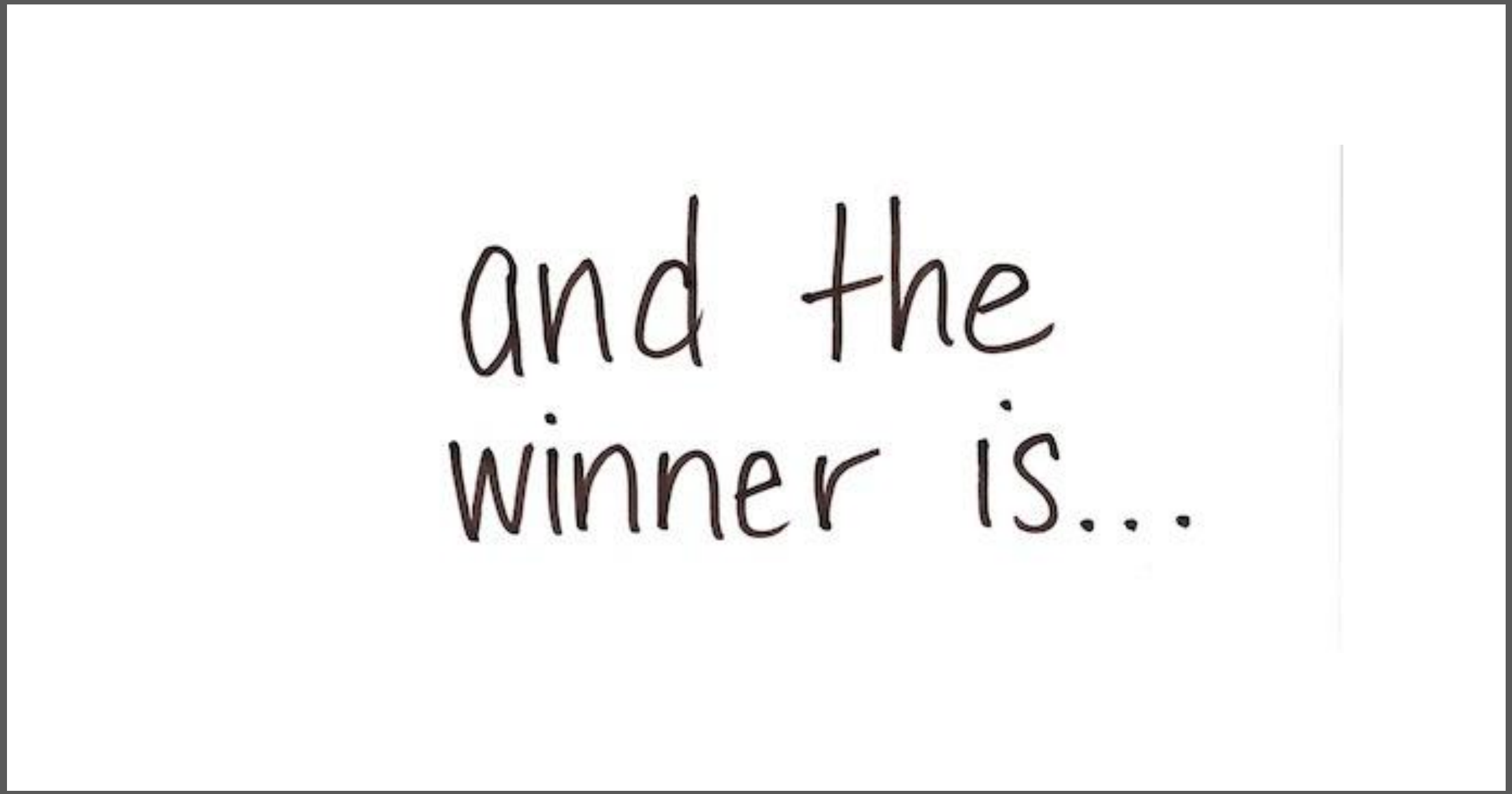
- Opakovačky
  - Štandardná štruktúra ľudovky
    - [: Sloha :] [: Refrén :]
  - Pre samotnú pieseň nie je opakovaný text refrénu/ slohy zaujímavý
  - Zátvorky a špeciálne znaky (teda {([/:.,‘“])}) tým pádom môžeme ignorovať
- Čísla ignorujeme tiež

Manuálna práca, ktorá nikoho nezaujíma

- „ … “
  - Dopísaný text, manuálne prejdenie časti databázy
- Čiastočne zatriedenie piesní podľa regiónov, tónin (dur, mol) a charakteru (valčík, polka...)

Technológie – minulý rok

- Programovací jazyk Java
- Databáza
  - SQL alebo MongoDB
- User interface
  - Angular 2, JSF


and the
winner is...

Java™

{JSON}

JavaScript Object Notation

JavaServer™ Faces

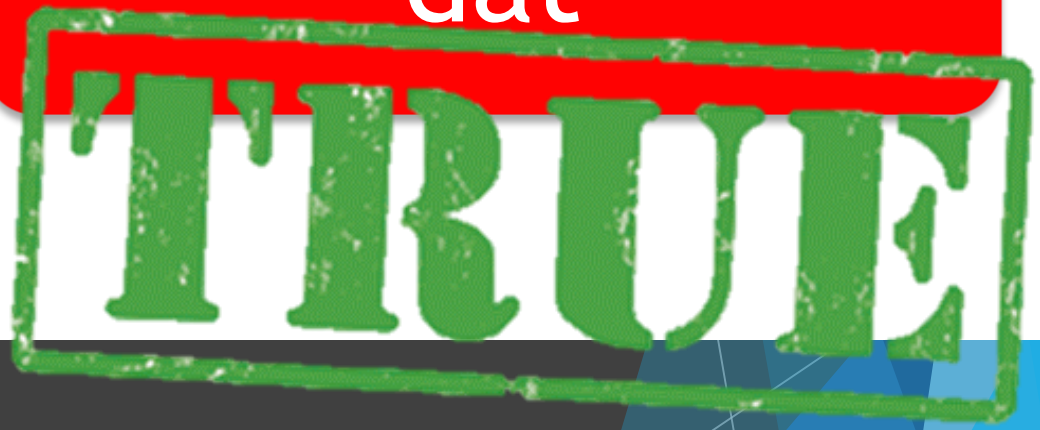
JSF

Zatiaľ statické, jednoduché dáta	Knižnica Jackson	„Glass cannon“
• Jednoduché na údržbu	• Rýchla serializácia/ deserializácia objektov	• Rýchle, odľahčené • Ťažšie sa udržuje konzistentnosť dát

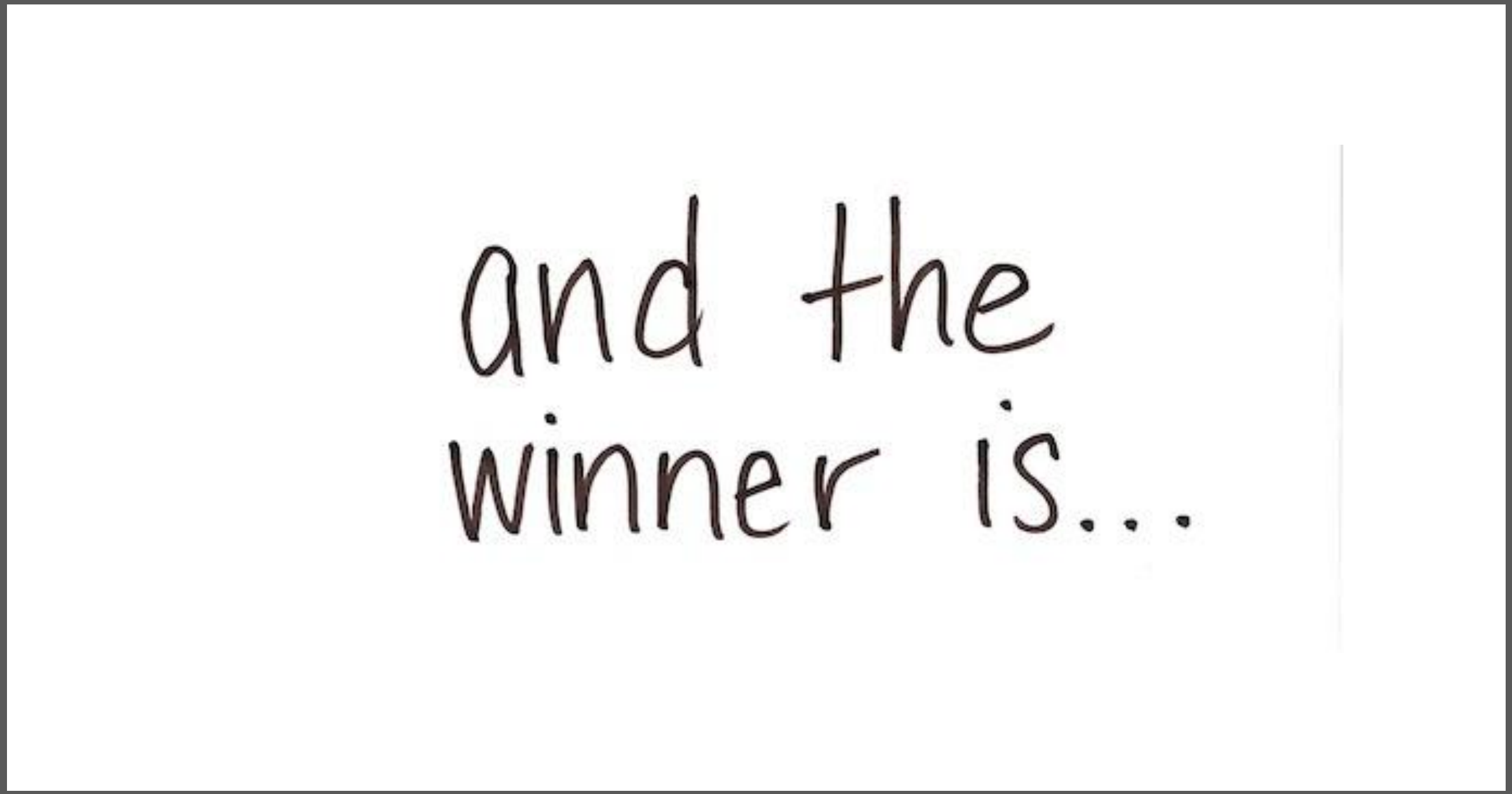
JSON

Zatiaľ statické, jednoduché dáta

Ťažšie sa udržuje konzistentnosť dát



„Oh, my sweet summer child...“


and the
winner is...


Java


Nejaké...
SQL


JavaServer™ Faces
JSF

Prečo nie?

Pre mňa nová technológia, mám sa s kým poradiť

JSF

JavaServer™ Faces
JSF

Čo vlastne kódim?

1. Vložím text
2. Spustím algoritmus
3. (?) Priebežne počas iterácie konfiguráciami algoritmu vypisujem výsledky
4. Vrátim percentuálnu podobnosť textov

AlgorithmConfiguration

- Abstraktná trieda rozšírená pre každý typ algoritmu
- Immutable – pre prípad, že ostane čas na paralelizáciu („Oh, my sweet summer child...“)
- Jej „deti“ obsahujú inštančné premenné definujúce beh algoritmu

AlgorithmConfiguration

- Zatiaľ pracujem len na „Vector space algorithm“
- Idea – text si rozdelím na vektory, kde dimenzia je „slovo“ a hodnota jeho váha
- Vektory dvoch textov algebraicky porovnám

Na pripomenutie

wat wat	wat wat2	wat2 wat2	wat2 wat
2	1	1	1

```java
@Immutable
public class VectorAlgorithmConfiguration extends AlgorithmConfiguration {

    private TermScheme termScheme;
    /**
     * In case term scheme is more complex
     */
    private Integer termDimension;

    private TermWeightType termWeightType;
    private VectorInclusion vectorInclusion;
    private TermComparationAlgorithm termComparationAlgorithm;
    private Double tolerance;
    private VectorComparationAlgorithm vectorComparationAlgorithm;
```

TermScheme - N-gramy

- Poradie?
  - Rozšírenie databázy o n-tice slov – n-gramy

TermScheme - zovšeobecnenie

- „Ja chlapec vandrovny, zo širehо poľa“
  - Dvojice
    - <Ja, vandrovny>, <chlapec, zo>, <vandrovny, širehо>
    - <Ja, poľa>
  - Trojice
    - <Ja, chlapec, zo>, <chlapec, vandrovny, širehо>
  - Mnoho ďalších...

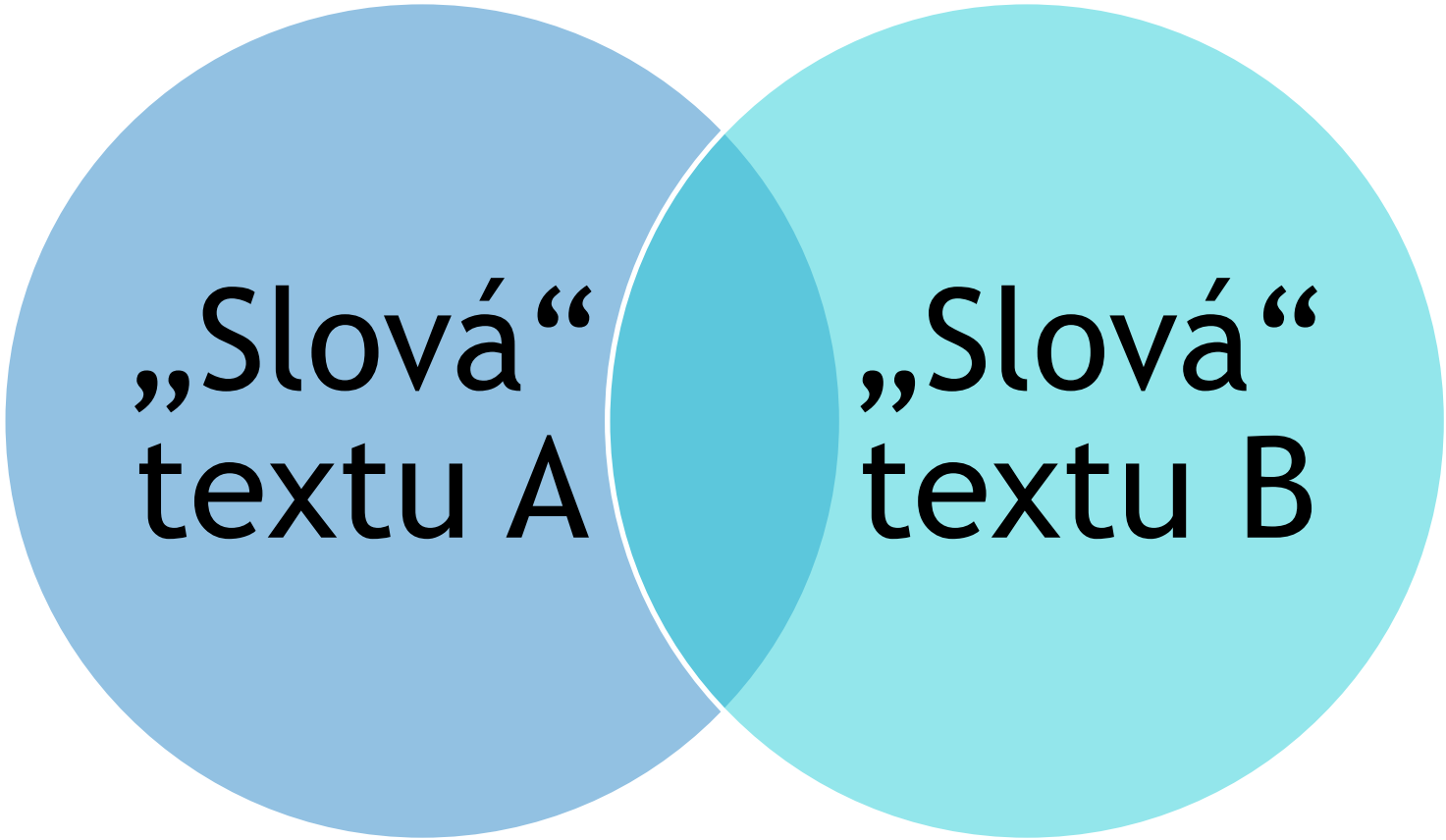
Term weighting

- „Slová“ v texte sú pre nás rôzne zaujímavé
  - Tf – term frequency
  - Idf – inverse frequency
  - Tf-idf (term frequency–inverse document frequency)
    - „A ja taka **čarna** jak **čarna čarnica...**“
    - „A ja taka **dzivočka cingi lingi bom...**“
  - **Mnoho ďalších...**

Vector inclusion

Poznámka – pracujeme s dimenziami vektora

Vector inclusion



Vector inclusion

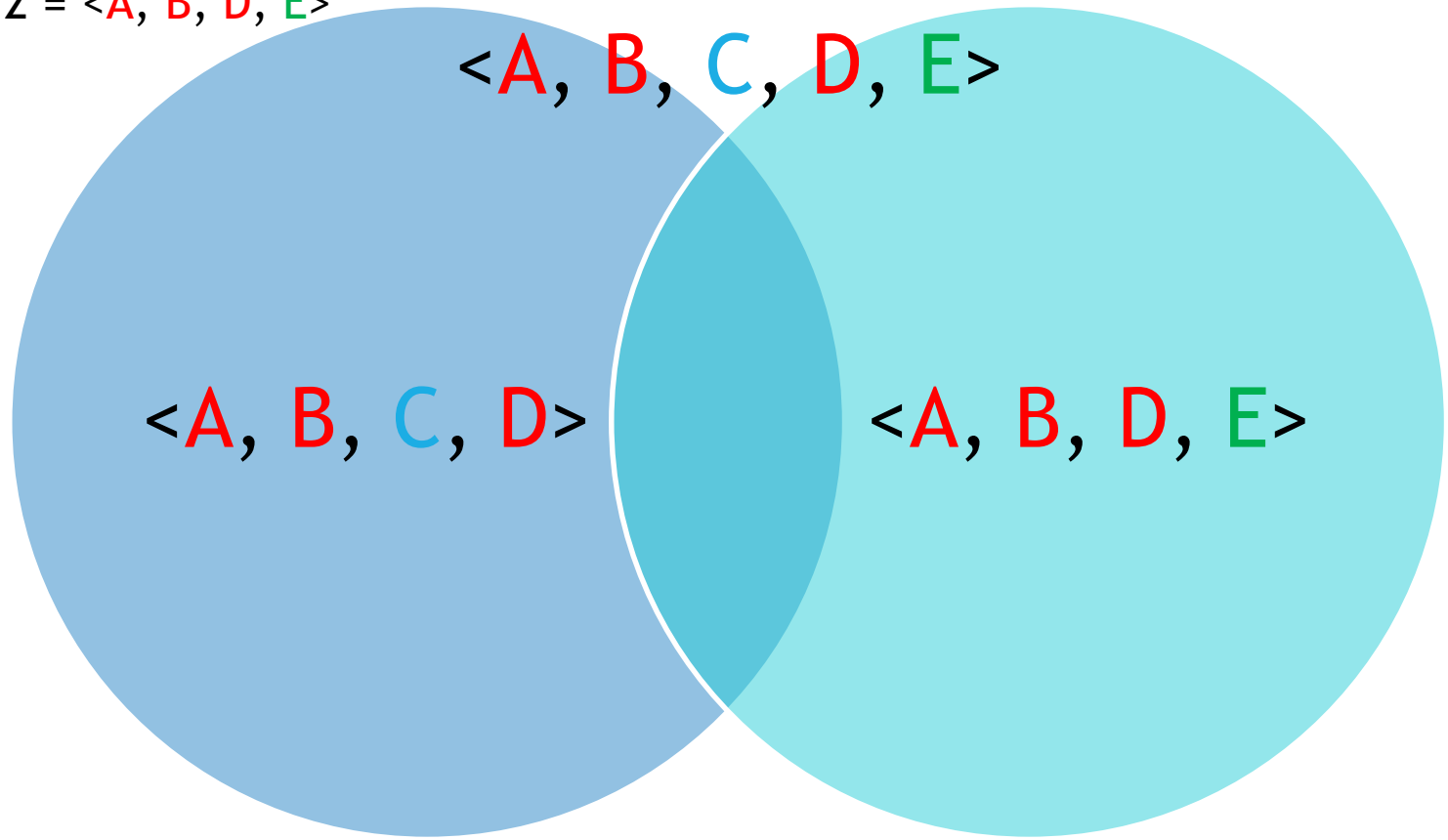
- Vektor 1 = <A, B, C, D>
- Vektor 2 = <A, B, D, E>

Vector inclusion

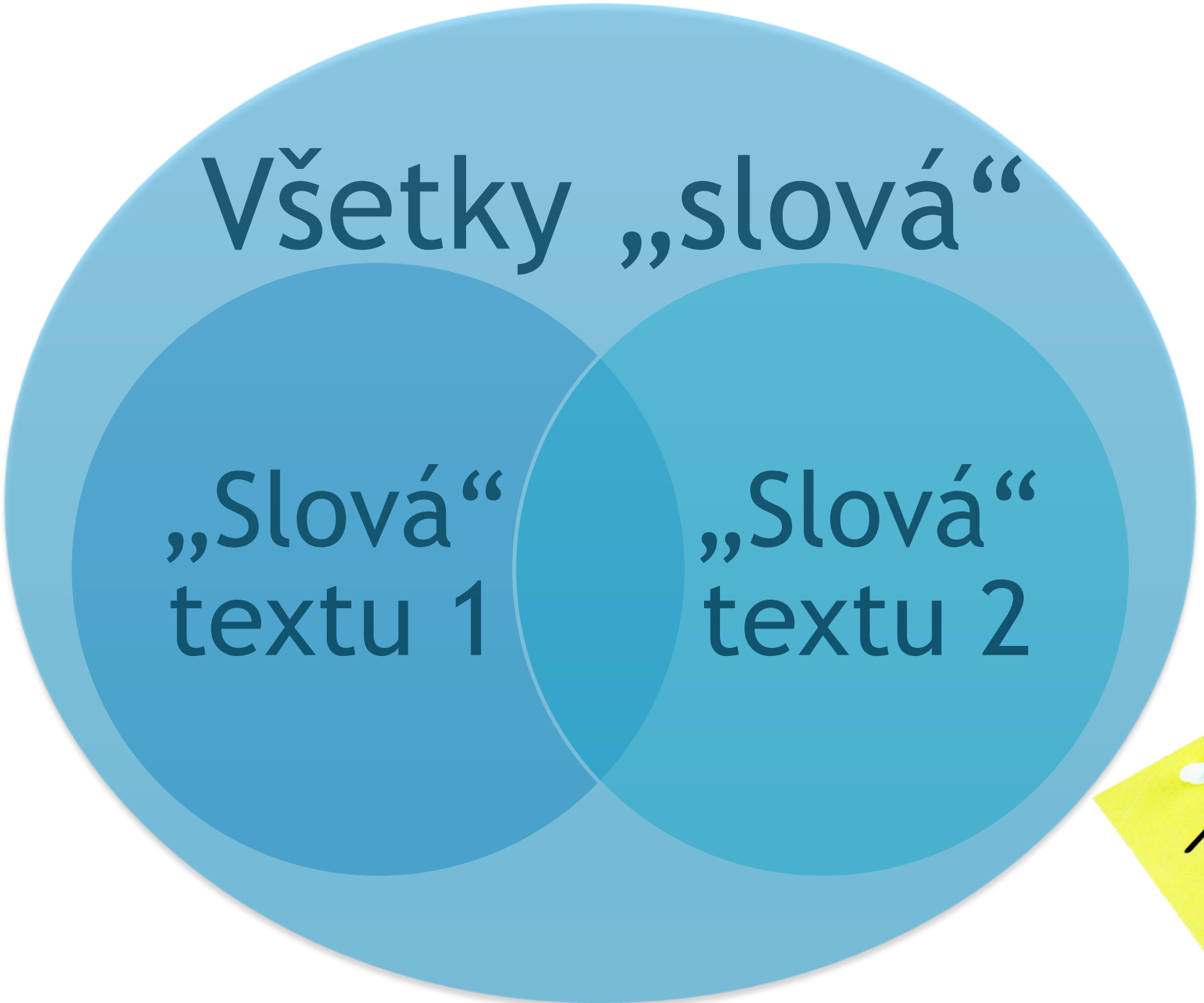
- A forma – C vo vektore B priradíme 0
  - < A, B, C, D >
- B forma - E vo vektore A priradíme 0
  - < A, B, D, E >
- AB forma
  - < A, B, C, D, E >

Vector inclusion

Vektor 1 = <A, B, C, D>
Vektor 2 = <A, B, D, E>



Krásne formátovanie, viem


Všetky „slová“
„Slová“ textu 1
„Slová“ textu 2


To Be Continued...

Term comparation algorithm

- Pri formátovaní vektorov je potrebné „slová“ porovnať
- Pre základné vyhľadávanie bez podobnosti slov stačí naivný algoritmus

```java
public class NaiveTermComparator implements ITermComparator {

    @Override
    public double compare(Term t1, Term t2) {
        return t1.getLyricsFragment().equals(t2.getLyricsFragment()) ? 1 : 0;
    }

}
```

Term comparation algorithm + Vector Inclusion

- Keď už hľadáme podobnosť piesní, tak prečo nie aj „slov“ ☺
- Musíme zlučovať stĺpce

Term comparation algorithm + Vector Inclusion

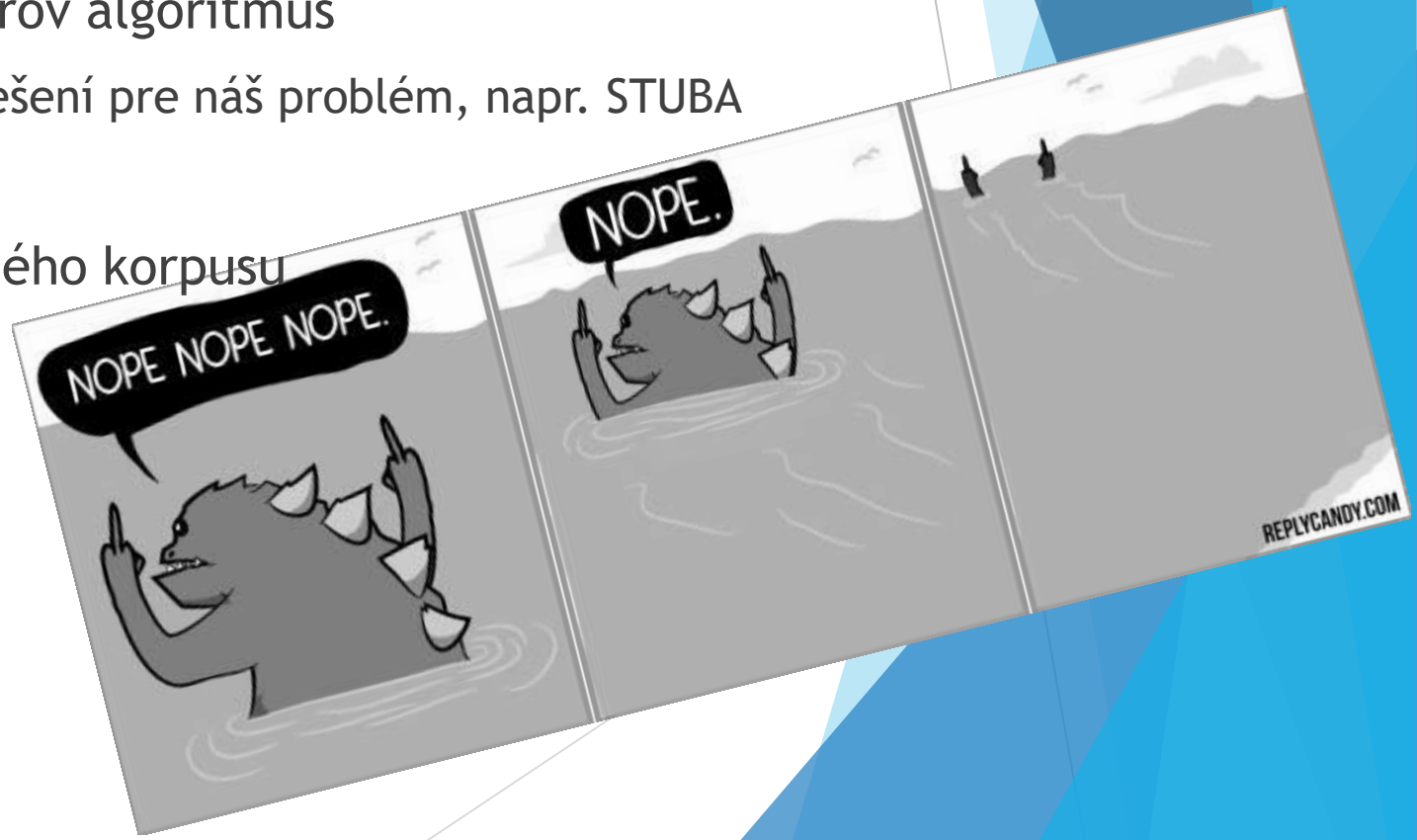
- Vektor 1 = <A, B, C, D>
- Vektor 2 = <A, B, D, E>
  - Pozn. toto sú dimenzie vektora, nie hodnoty

Term comparation algorithm + Vector Inclusion

- Predpokladajme, že C a E sú v tolerancii podobnosti (definovaná v alg. config.)
- Dostaneme
  - Vektor 1 = <A, B, C, D>
  - Vektor 2 = <A, B, E, D>

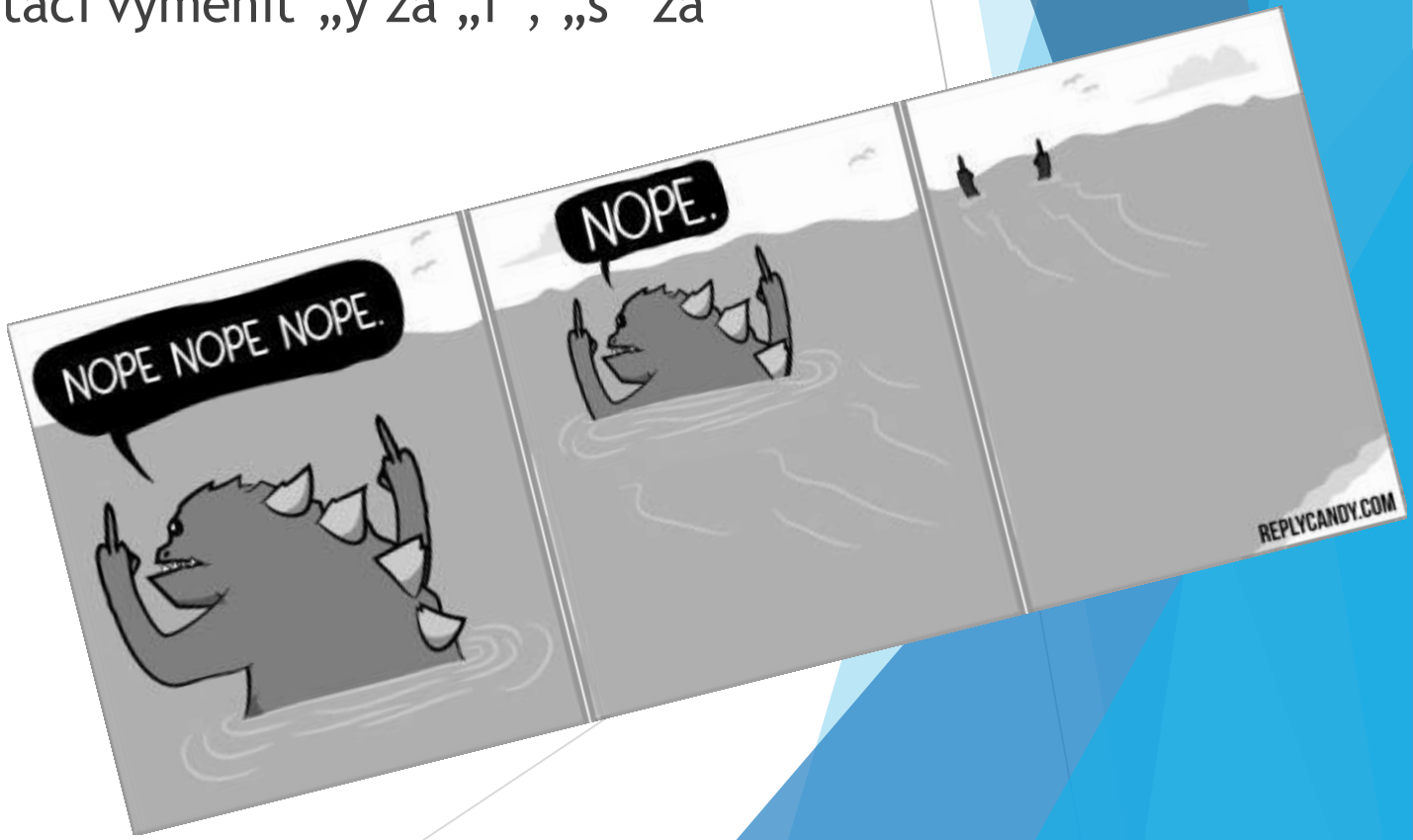
Term similarity

- Hľadanie základu slova – zlučovanie stĺpcov
  - Lemmatization, stemming – porterov algoritmus
    - Analýza efektivity existujúcich riešení pre náš problém, napr. STUBA
    - Ohýbanie riešení na ľudový jazyk
  - Korpus nárečí Slovenského národného korpusu



Term similarity

- Tvaroslovník – vytváranie vlastného ľudového „tvaroslovníka“
  - Podobnosti sú často veľmi veľké, stačí vymeniť „y za „i“, „š“ za „ś“, „h“ za „g“ a pod.
  - Generovanie koncoviek



Term similarity

- Levenshteinova vzdialenosť

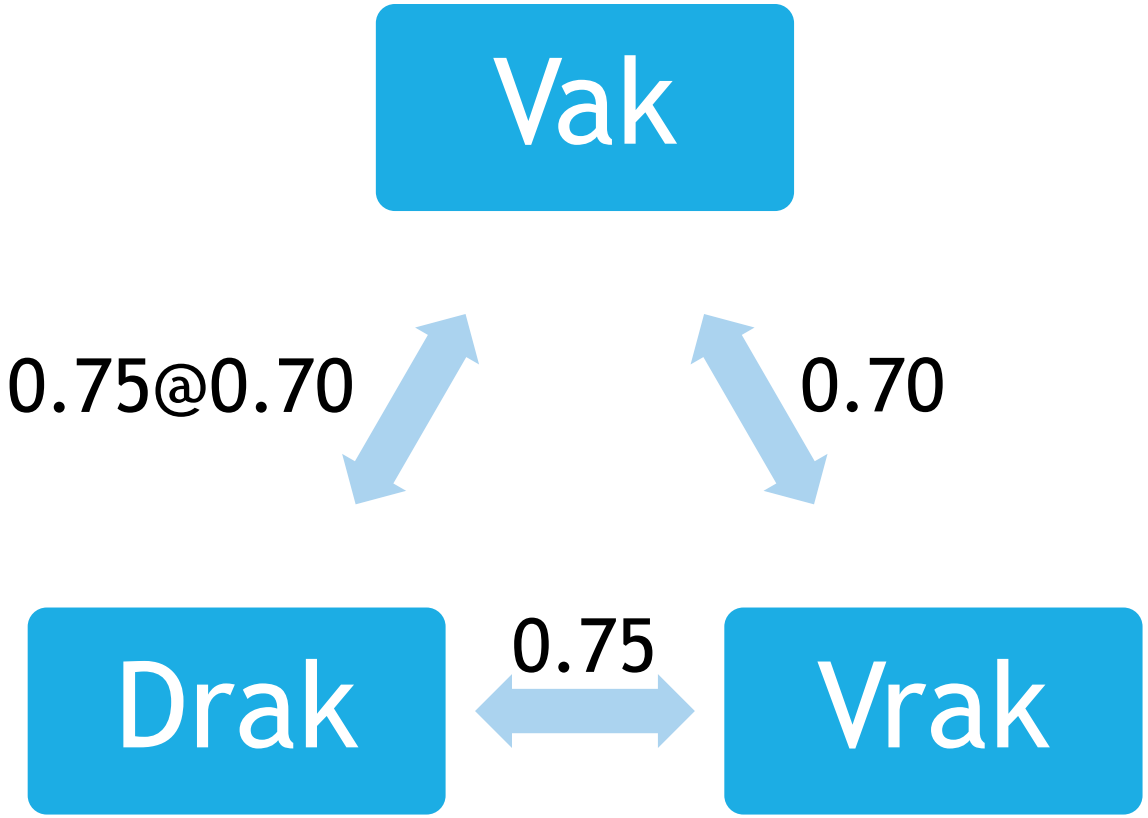
1. **k**itten → **s**itten (substitution of "s" for "k")
2. sitt**e**n → sitt**i**n (substitution of "i" for "e")
3. sittin → sittin**g** (insertion of "g" at the end).

- Mnoho ďalších...

Indirect term similarity

- Zatiaľ len nápad, ktorý treba doladiť
- Idea – „slová“ si nemusia byť podobné priamo, ale cez nejakého suseda
- Ilustrácia - číslo naznačuje, nakoľko sú si „slová“ podobné v danom porovnávacom algoritme
- @ - nejaká vhodná operácia, napríklad súčin

Indirect term similarity



Indirect term similarity

- Dostaneme teda vzorec
- $Pi(a,b) = \max(P(a,b), \max(Pi(a,c) @ Pi(b,c))), \ kde\ c\ \epsilon\ Vsetky"Slova"$
- Kde Pi je Indirect term similarity, P je Term similarity a a, b sú „slová“

Indirect term similarity

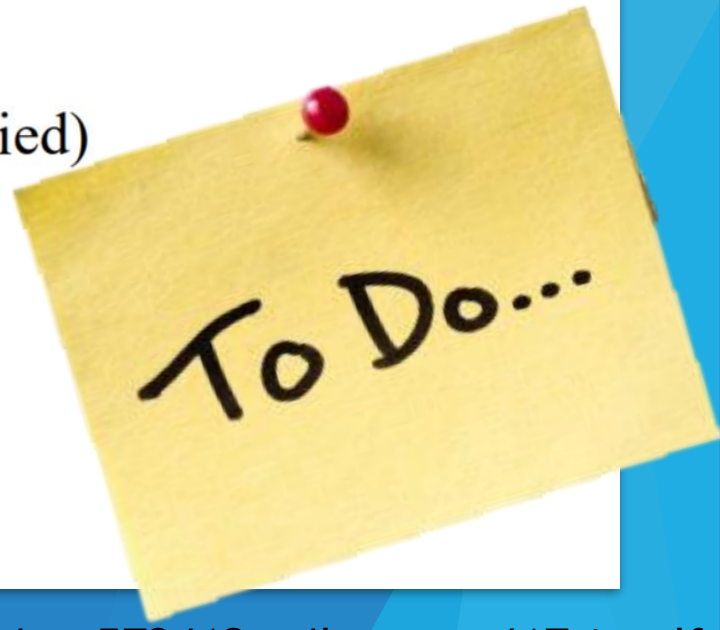
- Prípadne len
- $Pi(a,b) =$
  $\max(\ P(a,b),\ \max(\ P(a,c)\ @\ P(b,c))),\quad kde\ c\ \epsilon\ Vsetky"Slova"$
- Alebo inak doladiť „mieru nepriamosti“

Vector comparation algorithm

- Vzdialenosť dvoch vektorov
  - Kosínusová miera

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$, w

$\lvert Q \cap D \rvert$	Simple matching (coordination level match)
$2\frac{\lvert Q \cap D \rvert}{\lvert Q \rvert + \lvert D \rvert}$	Dice's Coefficient
$\frac{\lvert Q \cap D \rvert}{\lvert Q \cup D \rvert}$	Jaccard's Coefficient
$\frac{\lvert Q \cap D \rvert}{\lvert Q \rvert^{1/2} \times \lvert D \rvert^{1/2}}$	Cosine Coefficient (what we studied)
$\frac{\lvert Q \cap D \rvert}{\min(\lvert Q \rvert, \lvert D \rvert)}$	Overlap Coefficient



Zdroj: https://courses.cs.washington.edu/courses/cse573/12sp/lectures/17-ir.pdf

Zdroje

- Texty ľudových piesní
  - Rôzne stránky
  - Kontakty u nadšencov
- https://en.wikipedia.org/wiki/N-gram
- https://en.wikipedia.org/wiki/Levenshtein_distance
- https://courses.cs.washington.edu/courses/cse573/12sp/lectures/17-ir.pdf
- https://nlp.stanford.edu/IR-book/

Ďakujem za pozornosť