Anchor texts a štatistika k anchor textom. Document frequency, collection frequency.

Matúš Krajčovič

Problém a motivácia

- **anchor text** = viditeľný text linku (alternatívny popisok)

```
<a href="link">Anchor text</a>
```

- sú dôležité pre **SSO**, skóre vo vyhľadávaní ale aj z hľadiska používateľského zážitku
- dajú sa využiť aj na vyhľadávanie synoným či významu slov

Existujúce riešenia

- získavanie sémantických informácií z linkov
  - každý hovorí o vzťahu dvoch pojmov
- zisťovanie podobnosti pojmov = boli vytvorené aj modely
- zisťovanie synoným
- podľa výskumov sú odkazy najlepšou metrikou vzťahu dvoch pojmov
- tiež je dôležitá pozícia linku v texte
- existujú riešenia na parsovanie wikipédie, aj konkrétne pre linky na wikipédii

Postup práce

1. **Parsovanie**
   - použitie Sparku
   - spracovanie tagov <title> a <text> v XML súbore
   - použitie regexu na vyhľadávanie linkov
   - počítanie document a collection frekvencie pomocou map() a reduce()
2. **Indexovanie a vyhľadávanie**
   - použitie PyLucene
   - vytvorenie indexu pre názov linku a jeho frekvencie
   - umožnenie vyhľadávania = zoradenie podľa collection frekvencie
3. **Testovanie**
   - na malých dátach = validácia správnosti parsovania a hodnôt collection a document frekvencii

Použité dáta a softvér

Dáta

- XML dump z ENG wikipédie

```
<siteinfo/>
<page/>
  <title/>
  <id/>
  ...iné tagy...
  <text/>
    infobox
    samotný text článku
<page/>
<page/>
```

```
[[link|alt]]
```

Softvér

- *parse.sh* = spúšťa **parse.py** skript so spark-submit príkazom a funkciami na map() a reduce()
- *index.sh* = spúšťa PyLucene indexovanie pomocou **index.py** skriptu nad výstupom z parsovania
- *search.sh* = pomocou **search.py** skriptu iteratívne vyhľadáva v indexe

Záver

- podarilo sa sparsovať linky na celej wikipédii pomocou Sparku
- skúšal som alternatívne popisky aj samotné linky = alternatívnych je viac
- úspešne vytvorený index a vyhľadávanie nad ním