Vyhledávání a enterprise aplikace

* **enterprise search** = identifikace specifického obsahu napříč enterprise systémem a povolení k:
  - indexaci
  - vyhledání
  - zobrazení

  } autorizovaným uživatleům

  - je to organizované vrácení strukturovaných a nestrukturovaných dat v aplikaci

  - rozdíly oproti klasickému vyhledávání:
    - vícero zdrojů dat (weby, soubory, email,…)
    - kolekce a indexace dat
    - relevance a ranking algoritmy
    - uživatelé
    - bezpečnost (autentizace,…)
    - federativní vyhledávání (jeden request, prohledání vícero search enginů)

Enterprise Search Components

Collecting data / Sběr dat

- nalezení obsahu, stažení do systému
- „Crawlers“ vrací dokumenty a další obsah
  - přes HTTP
  - používají adaptéry k připojení k RDBMS, DMS,…

Content processing / Zpracování obsahu

- **identifikace vět** - rozhodování dle interpunkčních znamének apod.
  - The operator operates successfully!

- **Tokenizace** – rozdělování na tokeny (slova, fráze, symboly,…)
  - [The] [operator] [operates] [successfully]

- **Normalizace** – lower-case tokenů → case-insensitive vyhledávání
  - [the] [operator] [operates] [successfully]

- **Stop-words removing** – odstranění zbytečných tokenů (there, so, other,…)
  - [operator] [operates] [successfully]

- **Stemming and lemmatization** – normálová forma slova
  - [operate] [operate] [success]

- **Synonym expansion** – manuální či automatický thesaurus (Wordnet,…)

- **POS (part-of-speech) tagging**
  - the **book** on the table (podst. jméno), to **book** a flight (sloveso)

Indexing / Indexace

- výsledné termíny jsou ukládány v indexu (namísto ukládání celého dokumentu)
- obsahuje slovník všech unikátních slov
- seskupuje informace do logických kategorií, které mohou být prohledávány

Indexing – TF-IDF

- **TF = Term Frequency** = jak často se termín objevuje v jednom dokumentu
  - = počet výskytů / počet všech termínů v dokumentu

- **IDF = Inverse Document Frequence** = jak důležitý je temrín v korpusu
  - log (počet všech dokumentů / počet dokumentů s výskytem termínu)

- slovo je **populárnější**, objevuje-li se v dokumentu vícekrát
- slovo je **důležitější**, objevuje-li se v méně dokumentech

Vyhledávací platformy/knihovny

Lucene

- Java open-source full-text vyhledávací knihovna
- snadné přidání full-textového vyhledávání
- není kompletní aplikací, ale knihovnou a API
- addDoc() = přidání dokumentu do indexu
- TextField(…) = obsah, který chceme tokenizovat
- StringField(…) = ID a další obsah, který nechceme tokenizovat

Elasticsearch

- open-source vyhledávací server, poháněn Lucene
- napsán v Javě, ale multiplatformní
- škálovatelná, distribuovaná architektura
- HTTP REST API
- bezschématické JSON dokumenty
- prakticky real-time vyhledávání
- např. Wikimedia, Quora, SoundCloud, Github, Netflix,…

- **mapping**:
  - full-text: { "type": "string", index: "analyzed" }
  - přesný string: { "type": "string", index: "not_analyzed" }
  - nevyhledávatelné: { "type": "string", index: "no" }

- **filtry vs. dotazy:**

Filters	Queries
{ "term": {"date": "2018-1-3" }}	{ "match": {"tweet": "search" }}
přesná shoda	full-text
binární – ano/ne	skóre relevance
rychlé	těžké
cacheable	not cacheable

- **agregace** = seskupení dat (dle podmínek)

Solr

- postavené na Lucene, podobné ElasticSearch
- perfektní pro vyhlevání na jednom serveru
- Solr – pro textové vyhledávání
- ElasticSearch – pro filtrování, seskupování, analytické dotazy,…

Evaluation of search system

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

	Documents Retrieved (search results)		
Actual Documents (Should be retrieved)		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Jaké je špatné vyhledávání?

- žádný search box
- příliš shod – vrátí 20k, ale průměrný uživatel mrkne na top 20
- špatné skórování – nejrelevantnější výsledky nejsou navrchu
- špatná detekce duplikátů
- neschopnost posouzení uživatelova záměru (spell checking, auto-complete,…)