Full-text vyhledávání

Full-text vyhledávání (často označované jako plnotextové vyhledávání) je pokročilá technika vyhledávání textových informací v rozsáhlých databázích, dokumentech nebo na webových stránkách. Na rozdíl od běžných relačních databázových dotazů, které hledají přesnou shodu znaků (např. pomocí operátoru `LIKE '%slovo%'`), full-textové systémy chápou přirozený jazyk, ignorují bezvýznamová slova, umí pracovat se skloňováním a dokáží výsledky seřadit podle relevance.

Tato technologie je naprosto klíčová pro moderní webové vyhledávače, e-shopy, znalostní báze a systémy pro správu logů, kde je nutné prohledávat miliony textových dokumentů v řádech milisekund.

Princip fungování a Invertovaný index

Základním stavebním kamenem full-textového vyhledávání není sekvenční procházení textu od začátku do konce (což by bylo u objemných dat extrémně pomalé), ale využití datové struktury zvané **Invertovaný index** (*Inverted Index*).

Předtím, než lze v textu vůbec vyhledávat, musí každý dokument projít procesem analýzy a indexace. Tento proces se skládá z několika fází:

- **Tokenizace (Tokenization):** Vstupní text je rozsekán na jednotlivá slova (tokeny). Odstraní se interpunkční znaménka a všechna písmena se zpravidla převedou na malá (lowercase).
- **Filtrace stop-slov (Stop words):** Systém odstraní slova, která nemají téměř žádnou informační hodnotu a vyskytují se v jazyce příliš často (např. spojky, předložky, zájmena jako „a“, „nebo“, „se“, „v“).
- **Stemming a Lematizace:** Zbylá slova jsou oříznuta na svůj slovní základ (kořen). Například slova „běžet“, „běžel“ a „běžíme“ jsou převedena na společný základ „běž“. Díky tomu uživatel najde dokument i tehdy, když hledá slovo v jiném pádu nebo čase.
- **Tvorba invertovaného indexu:** Výsledná slova se uloží do struktury, která funguje podobně jako rejstřík na konci knihy. Index neříká „jaká slova obsahuje dokument A“, ale naopak „ve kterých dokumentech a na jaké pozici se nachází slovo X“.

Když poté uživatel zadá vyhledávací dotaz, systém jej analyzuje stejným způsobem a následně rovnou sáhne do invertovaného indexu, ze kterého okamžitě získá seznam relevantních dokumentů.

Relevance a skórování (TF-IDF)

Zásadní vlastností full-textových vyhledávačů je schopnost vracet výsledky seřazené podle toho, jak moc odpovídají dotazu uživatele. K výpočtu relevance se historicky nejčastěji používá algoritmus **TF-IDF** (*Term Frequency – Inverse Document Frequency*), nebo jeho modernější varianta **BM25**.

Tyto algoritmy počítají skóre na základě dvou hlavních metrik:

- **TF (Frekvence slova):** Čím častěji se hledané slovo vyskytuje v konkrétním dokumentu, tím vyšší je jeho skóre.

- **IDF (Inverzní frekvence v dokumentech):** Čím vzácnější je slovo napříč celou rozsáhlou databází, tím větší váhu má. Pokud uživatel hledá frázi „vzácná orchidej“, slovo „orchidej“ bude mít při výpočtu mnohem vyšší váhu než slovo „vzácná“, protože se v textech vyskytuje méně často.

Výhody a nevýhody full-textového vyhledávání

Výhody

- **Extrémní rychlost:** Díky invertovanému indexu je prohledávání obřích datasetů (desítky a stovky gigabytů textu) téměř okamžité.
- **Znalost jazyka:** Podpora synonym, skloňování a dokonce i automatická korekce překlepů (Fuzzy search).
- **Řazení dle relevance:** Uživatel dostane smysluplné výsledky na prvních místech, nikoliv pouze ty nejnovější nebo abecedně seřazené.

Nevýhody

- **Náročnost na úložiště:** Invertovaný index zabírá značné množství diskového prostoru. Systém musí udržovat původní text i rozsáhlé slovníkové indexy, což násobí velikost uložených dat.
- **Režie při indexaci (zápisu):** Každý nový dokument musí být analyzován a zapsán do indexu, což spotřebovává výkon CPU. Full-text není vhodný pro transakční data, která se extrémně rychle a neustále mění (např. zůstatky na bankovních účtech).

Populární technologie a enginy

Na trhu existuje několik dedikovaných vyhledávacích enginů, které jsou postaveny specificky pro plnotextové úlohy.

- **Elasticsearch / OpenSearch:** Absolutní průmyslový standard pro full-text, e-commerce a log management. Jedná se o distribuovaný, vysoce škálovatelný engine postavený na open-source knihovně Apache Lucene.
- **Apache Solr:** Starší, ale stále velmi robustní a oblíbený vyhledávač, rovněž postavený na Apache Lucene. Skvěle zvládá statické dokumenty.
- **Manticore Search / Sphinx:** Extrémně rychlé a na systémové zdroje nenáročné vyhledávače. Jsou populární v systémech vyžadujících bleskové odezvy s výrazně menší režií než Elasticsearch.
- **Relační databáze (PostgreSQL, MySQL):** Většina moderních SQL databází má zabudované své vlastní plnotextové schopnosti. Zejména PostgreSQL nabízí překvapivě výkonný full-text modul, který pro menší až střední projekty zcela eliminuje nutnost nasazovat složitý dedikovaný engine.

Srovnání: Full-text vs. Standardní SQL vyhledávání (LIKE)

Vlastnost	Relační SQL (operátor LIKE)	Full-textový engine (např. Elasticsearch)
Rychlost na velkých datech	Velmi pomalá (nutnost prohledávat záznamy sekvenčně, tzv. Full Table Scan)	Extrémně vysoká (využívá připravený invertovaný index)
Pochopení skloňování a synonym	Ne (hledá přesnou shodu sekvence znaků)	Ano (využívá lematizaci a custom slovníky synonym)
Řazení výsledků	Pouze podle hodnot ve sloupcích (abecedně, datum, ID)	Podle komplexní relevance (skórování shody s dotazem)
Zpracování překlepů	Ne (jakýkoliv překlep znamená nenalezení záznamu)	Ano (podporuje toleranci chyb – tzv. Fuzzy vyhledávání)
Infrastrukturní náročnost	Nízká (funkce zabudovaná ve stávající databázi)	Vysoká (často vyžaduje vlastní dedikovaný cluster serverů a údržbu)