

Flickr reranking

Řešitelé: Daniel Kavan <kavanda1@fit.cvut.cz>, Filip Rajnoch <rajnofil@fit.cvut.cz>

Předmět: MI-WMV, sudé pondělí 12:45, paralelka 101

Obsah

POPIS PROJEKTU	2
SPECIFIKACE SYSTÉMU	2
POUŽITÉ TECHNOLOGIE	2
POPIS JEDNOTLIVÝCH FUNKCÍ.....	2
POPIS MOŽNOSTÍ SROVNÁNÍ A EXTRAHOVANÉ VLASTNOSTI	2
LOGICKÉ VRSTVY APLIKACE (MVC)	3
ZPŮSOB ŘAZENÍ	3
POPIS JEDNOTLIVÝCH FUNKCÍ ŘAZENÍ	3
UŽIVATELSKÉ ROZHRANÍ	4
ZÁVĚR	5
LITERATURA.....	5

Popis projektu

Projekt Flickr reranking se zabývá porovnáváním metadat z fotografií, obrázků a videí uložených na portálu [Flickr](#) a dostupných skrz webovou službu typu REST. Účelem projektu je seřazovat data podle zvolených kritérií (autora, podobnosti názvu, počtu zhlédnutí, GPS pozice pořízení nebo data) z množiny nalezených fotografií. Výsledkem je webová stránka s vyhledáváním ve fotografiích a implementovaný systém porovnávání metadat.

Specifikace systému

Systém využívá externích nástrojů pro získávání obrazových dat a získávání metadat. Tyto údaje jsou uchovávány a druhotně zpracovány systémem s vlastními metodami porovnávání. Systém umožňuje vyhledávání dat podle klíčových slov nebo vyhledání posledních přidaných. Zobrazuje pak podle volby 1, 2, 5, 10, 20, 50, 75 nebo 100 výsledků s možností filtrování s horní, dolní nebo oběma hranicemi data přidání.

Použité technologie

Projekt je programován v jazyce **PHP** používá databázový systém **MySQL** pro caching výsledků hledání. Celý projekt je koncipován architekturou MVC (model-view-controller), který odděluje funkční, datovou a prezentační část. Zpřehledňuje a ulehčuje tak další úpravy a rozšiřování systému.

Jako externí nástroj pro získání dat a extrakci jejich meta informací jsme použili třídu Flickr PHP API (`phpFlickr`), které poskytuje veškeré potřebné funkce pro získání obrazových dat, jejich meta informací a cachování do databáze.

Pro grafickou stránku projektu jsme využili **CSS** a **jQuery** pro dynamické javascriptové změny.

Popis jednotlivých funkcí

Popis možností srovnání a extrahované vlastnosti

Flickr reranking nabízí 8 různých seřazení zobrazených dat, a to podle:

- názvu
- podobnosti názvu vůči zadanému vzoru
- počtu zhlédnutí
- podobnosti vůči zadanému počtu zhlédnutí
- zadaných geografických dat
- typu media (video/fotografie)
- velikosti (rozměrů)
- data nahrání

Pro tyto účely extrahujeme pomocí Flickr PHP API tyto informace: název, počet shlédnutí, odkaz na flickr webovou stránku s obrázkem, adresu obrázku, GPS souřadnice, typ media, datum nahrání a rozlišení media.

Logické vrstvy aplikace (MVC)

Model

Vrstva modelu je reprezentována především třídou `phpFlickr`, která nám poskytuje potřebné rozhraní pro komunikaci se servery Flickru a ostiňuje nás od řešení získávání dat a přístupu k databázi za účelem cachingu.

Mezi pomocné třídy pro reprezentaci dat při běhu programu slouží třídy `Media`, `Photo`, `Video`, `Geo` a `Dimensions`.

Controller

Vrstvu funkční logiky tvoří `Search` a `Rerank`, jež jsou zodpovědné za klíčové prvky systému, tedy hledání s využitím vrstev Modelu a samotný reranking podle zvolených kritérií.

View

Vrstva uživatelského rozhraní se stává z třídy `UI` a jednotlivých šablon (templatů), které naplňujeme daty relevantními ke vstupům uživatele.

Způsob řazení

Řazená probíhá vždy na vyhledaných datech za použití Flickr PHP API k extrahování potřebných metadat. Podle zvoleného typu řazení je volána příslušná metody třídy `Rerank` reprezentující řazení podle daných kritérií. Interně se pak taková řadící metoda skládá z využití modelu podobnosti pro dané kritérium a porovnání samotné je řešeno funkcí `usort`, která umožňuje jako parametr zadat název callback funkce použité pro porovnání, řadicím algoritmem je implementace QuickSortu.

Popis jednotlivých funkcí řazení

U běžných číselných údajů probíhá porovnávání testováním na větší/menší hodnotu. Toto se provádí u porovnání dle rozměrů, dle počtu zhlédnutí, dle rozdílu od zadaného počtu zhlédnutí, podle typu média a data nahrání média (údaj je zadán ve formátu UNIX timestamp).

Pro seřazení podle názvu se používá na porovnání PHP funkce `strcmp`, která vrací informaci o abecedním pořadí dvou řetězců.

Porovnání podobnosti názvu se zadaným textem se provádí třemi způsoby, podle toho, který si uživatel zvolí. Na výběr jsou metody:

1. Levenshteinova vzdálenost

- Udává, kolik znaků je třeba přidat, odebrat či zaměnit, abychom se dostali od řetězce a k řetězci b (zdroj 1a, 4b)
 - nejrychlejší funkci pro srovnání dvou textových řetězců z našeho výběru
2. Similar text
 - Vrací počet shodných znaků v řetězci
 3. Longest Common Substring
 - Zjišťuje délku nejdelšího společného podřetězce

Seřazení podle vzdálenosti probíhá pomocí funkce, která spočítá rozdíl jednotlivých bodů z geo informací a pomocí goniometrických funkcí vypočítá jejich vzdálenost (viz zdroj 5).

Uživatelské rozhraní

Aplikace má především akademický charakter – demonstrovat využití a práci s API, následně modelovat a aplikovat řešení problému podobnosti na daný výběr dat – uživatelské rozhraní není klíčovou částí aplikace.

Z toho důvodu jsme se rozhodli nevyužít žádný hotový prostředek pro prezentaci dat v podobě frameworků, místo toho prezentaci výsledků řešíme sami templatovací třídou `UI` a předdefinovanými templaty jednotlivých částí rozhraní. Jak naznačeno v ukázce templatu, dodržujeme filozofii MVC vrstev a důsledně uživ. rozhraní od zbytku aplikace oddělujeme:

Ukázka části templatu pro hlavní stránku.

```
<div class="input">
    {inputBlock}
</div>
<div class="message">
    {message}
</div>
<div class="output">
    {outputBlock}
</div>
```

Jak již bylo zmíněno v sekci Použité technologie, pro realizaci prezentační stránky používáme dynamicky vytvořený HTML dokument, pro definici formy obsahu využíváme kaskádových stylů CSS a některé ovládací prvky uživ. rozhraní jsou doplněny skripty napsanými v Javascriptu (včetně využití knihovny jQuery).

Všechny skripty jsou naše autorské s výjimkou volně dostupného doplňku Date Input (zdroj 6, MIT Licence¹)

¹ Text MIT Licence např. na http://cs.wikipedia.org/wiki/Licence_MIT

Závěr

Výsledek práce Flickr reranking splňuje zadání a díky extrakci údajů pomocí externí třídy metodám řazení pomocí vlastních tříd probíhá seřazování v přijatelném čase. Největší prodleva se vyskytuje při získávání samotných obrazových dat, které se získávají a stahují pomocí externí Flickr PHP API třídy. Díky hotové datatabázi obrazových dat na serveru Flickr a dobře vypracované metodě cachování dat jsme ale nenašli lepší řešení získání vzorků.

Literatura

1. Wikipedia.org
 - a. [Levenshtein distance](#)
 - b. [Longest common substring problem](#)
2. Wikibooks.org: [Longest common substring implementation](#)
3. [Flickr API Documentation](#)
4. PHP.net
 - a. [usort\(\)](#)
 - b. [levenshtein\(\)](#)
 - c. [similar_text\(\)](#)
5. [Gowtham: Calculating Distance Between Two Locations Given Their GPS Coordinates](#)
6. [Leighton, J.: jQuery Date Input](#)
7. [jQuery API Documentation](#)