ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA APLIKOVANÝCH VĚD

Komplexní IR systém

Semestrální práce z předmětu KIV/IR
KIV/IR - Vyhledávání informací

Jaroslav Lehečka
(A22N0054P)

leheckaj@students.zcu.cz

19. května 2023

Obsah

Kapitola 1

Úvod

1.1 Úvod

Cílem práce je vytvořit komplexní IR systém. Systém bude vyhledávat stránky (fulltextově) v nacachovaných stránkách z prvního cvičení IR. Systém zobrazí míry Tf-idf hodnot u výsledků. Pro tuto semestrální práci jsem si vybral stránku s diskuzními příspěvky: https://forum.nette.org

1.2 Zadání

Cílem semestrální práce je naučit se implementovat komplexní IR systém s využitím hotových knihoven pro preprocessing. Vedlejším produktem bude hlubší porozumění indexerům, vyhledávacím systémům a přednáškám. Systém po předchozím předzpracování zaindexuje zadané dokumenty a poté umožní vyhledávání nad vytvořeným indexem. Vyhledávání je možné zadáním dotazu s logickými operátorem AND. Výsledek dotazu by měl vrátit top x (např. 10) relevantních dokumentů seřazených dle relevance, případně přes ListView.

Kapitola 2

Hlavní entity programu

Při tvorbě máme několik možností, jakým způsobem můžeme daný vyhledávací systém stavět. Nejčastější možností je Tf-Idf s cosinovou vzdáleností.

Implementace je provedena v Javě s pomocí grafické knihovny JavaFX. Systém je rozdělen do několika balíků dle typu operací, které provádí: components, org.controller, preprocessing, tfidf, ui, utils a ui.

2.1 Controller

V rámci package org.controller najdeme controllery pro ovládání celého GUI. Stará se o spouštění jednotlivých akcí, které jsou volány v rámci UI (zadání dotazu + vyhledávání + init). V rámci controlleru jsou uvedeny veškeré komponenty užívané v rámci GUI vyhledávacího systému.

2.2 Prerprocessing

V rámci modulu preprocessing se nachází část z druhého cvičení, která slouží pro přípravu indexů + částečný preprocessing.

2.3 Tfidf

Balík tf-idf obsahuje veškeré třídy a komponenty používané při výpočtu tf-idf a kosinové vzdálenosti. Popis implementace bude dále rozepsán.

2.4 UI

V rámci UI balíku najdeme MainWindow, která spouští instanci okna JavaFX.

2.5 Utils

Utils obsahují základní konstanty užívané napříč programem. Dále obsahují Processor, který je volán z Controlleru pro vytvoření jednotlivých akcí volaných z UI vrstvy. Dále je zde zastoupen návrhový vzor Přepravka pro předávání informací z metod.

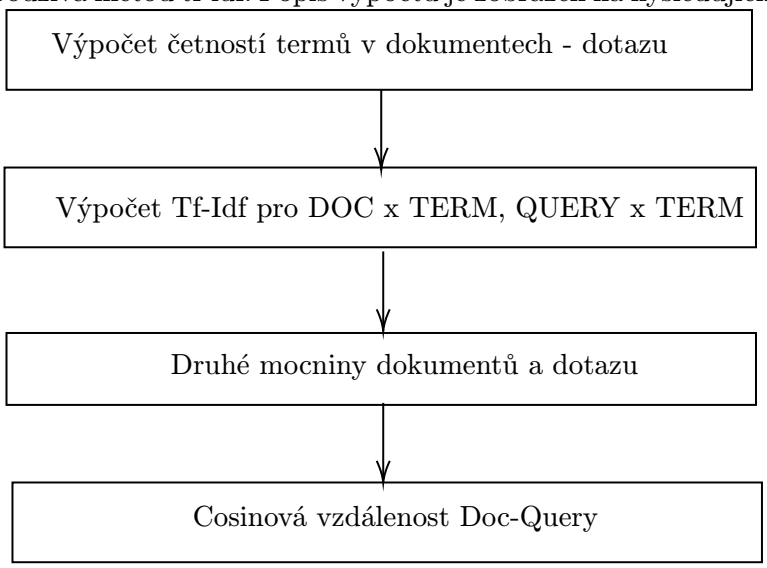
Kapitola 3

Implementace

Pro potřeby vyhledávání může být použito mnoho technologií. Jednou z nich je Tf-idf ve spojení s kosinovou vzdáleností dotazu. Cosinová vzdálenost lépe popisuje vzdálenost hledaného termu k dostupným vyhledaným dokumentům. Bližší schéma fungování Tf-idf bude ukázáno na následujícím schématu.

3.1 Schéma Tf-idf

Celý vyhledávací systém je postaven na bázi rankování a vyhodnocování. Pro zhodnocení relevance daného vyhledávaného termu k dokumentu se používá metod tf-idf. Popis výpočtu je zobrazen na nýsledujícím schématu



V první fázi je načtený a nacrawlovaný soubor zprocesován a je v něm vypočtena četnost jednotlivých termů v jednotlivých dokumentech. Vznikne nám tak matice doc-term.

V další fázi tuto matici použijeme a vypočteme hodnotu Tf-Idf pro jednotlivé dokumenty-termy na základě následujícího vzorce:

$w_{t,d} = (1 + log\ \ tf_{t,d}) * log\ \frac{N}{df_t}$

Nyní budou popsány veškeré proměnné. V první závorce se počítá logaritmická hodnota term frequency (=četnost termů) pro zadaný dokument a term. Jako násobitel je zlogaritmovaný podíl počtu dokumentů v kolekci dělený počtem dokumentů se zadaným termem. Takto je vypočtena hodnota tf-idf.

3.2 Cosinová vzdálenost

Na hodnoty tf-idf je následně aplikována cosinová vzdálenost, která je vypočtená podle následujícího vzorce:

$cos(\vec{q},\ \vec{d}) = \frac{\vec{q}\ \ \vec{d}}{|\vec{q}|\ |\vec{d}|}$

Cosinovská podobnost v čitateli pronásobý vektor dokumentu a termu (suma: hodnota na pozici i v termu * hodnota na pozici i v dokumentu). V rámci jmenovatele je norma, která bere součin absolutních hodnot druhých mocnin vektorů dokumentů a termů.

Kapitola 4

Uživatelská příručka

4.1 Předpokládané technologie

Program byl vyvíjen a testován na Windows 10 a Javě 11.

4.2 Ukázka spuštění programu

Pro spouštění programu se bude používat skript. V rámci skriptu je nutné zmínit instalaci JavyFX.

Případně je možné spouštět pomocí skriptu v němž je nutné upravit cesty k JavaFX.

```
$ run
```

4.3 Ukázka spuštění programu s natrénovaným modelem

Případně, pokud chceme startovat s předučeným modelem, tak je možné upravit soubor run takto:

```
java --module-path "C:\Program␣Files\OpenJFX11\lib"
--add-modules javafx.controls,javafx.fxml -jar SP.jar
-file=model_1684496135.ser
```

Soubor model_1684496135.ser je odkaz na serializovaný naučený model.

4.4 Jak ovládat program

Po spuštění aplikace se do horního šedého rámečku zadá hledaný výraz. Následně se v levé části vybere požadovaný dokument dle titulku toho dokumentu. V pravé části aplikace se zobrazí celý titulek aplikace + odkaz

na stránku + Cosinová vzdálenost + úryvek textu. Náhledy uživatelského rozhraní jsou v příloze.

Kapitola 5

Závěr

Výsledkem práce je funkční program. Práce mi byla přínosem k získávání zkušeností s programováním IR systému v Javě a obecně o ohodnocování a klasifikaci textů a dokumentů v rámci datové sady.

V rámci semestrální práce byly naimplementovány tyto funkcionality navíc oproti základu:

- File-based index
- ošetření HTML tagů
- Dokumentace psaná v TEXu
- GUI v JavaFX
- Zvýraznění hledaného textu v nadpisu
- Zvýraznění hledaného textu v těle příspěvku
- osekání html tagů v crawlované části

5.1 Příloha - Obrazovky programu