SLOVENSKÁ TECHNICKÁ UNIVERZITA

Fakulta informatiky a informačných technológii

Dokumentácia projekt VINF

Michal Lüley

Predmet: Vyhľadávanie informácii

Cvičiaci: Ing. Igor Stupavský

Akademický rok: 2022/23

Zadanie

O34

Programovací jazyk

Python 3.8.5 (framework jupyter notebook)

Framework

PySpark 3.3.1 (Scala 1.12)

Znenie zadania

Parsovanie ochorení z Wikipedie, vytvorenie služby umožňujúcej: vyhľadanie ochorenia na základe symptómov a krajiny, zistenie spôsobu šírenia ochorenia a typ ochorenia (vírusové, bakteriálne), vyhľadanie liekov na liečbu daného ochorenia pomocou databázy liekov.

Zdroj dát

https://dumps.wikimedia.org/enwiki/latest/

-Použité kompletné dáta wikipédie – 80GB

Github link

https://github.com/michal-luley/VINF-odovzdanie.git

Idea

Tému som si vybral vlastnú keďže sa zaujímam o použitie informatiky v oblasti zdravotníctva. Pozeral som možné projekty, na ktoré by bolo možné aplikovať vedomosti počas štúdia predmetu. Vybral som si tému, ktorá by raz mohla pomôcť pacientom ako aj lekárom. V projekte vyhľadávam ochorenia, krajiny ich výskytu, príznaky, symptómy. Používateľ si dokáže na základe parametra, ktorý si zvolí vyselektovať pomocou programu konkrétne ochorenia.

Prehľad súčasných riešení

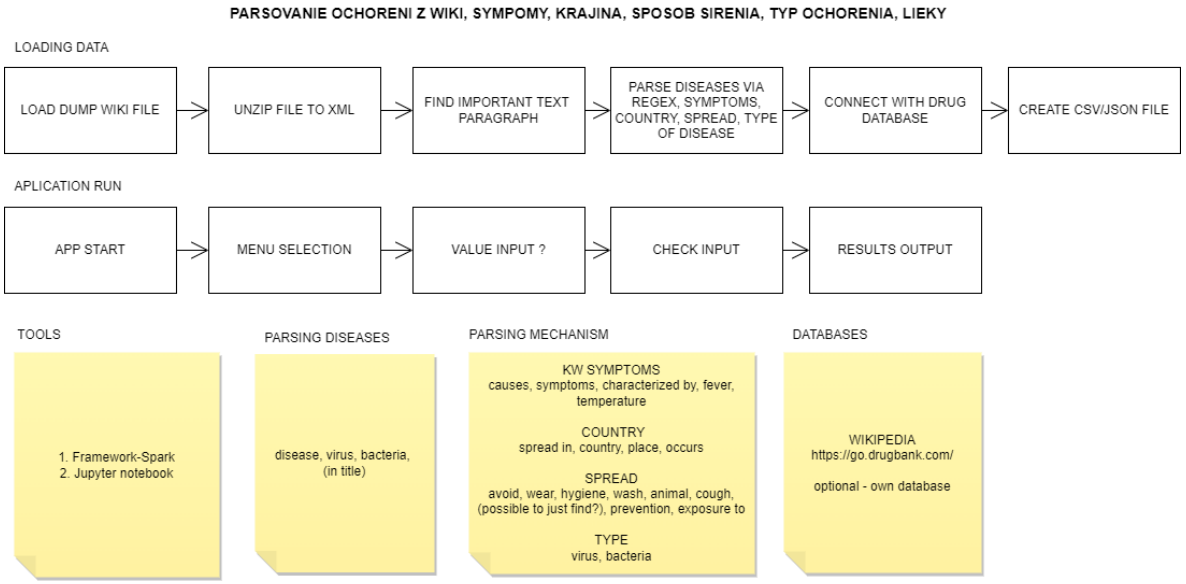
Na riešenie daného problému nie sú v súčasnosti dostupné žiadne riešenia. Z toho dôvodu ich neuvádzam.

Popis riešenia – fáza1 (do 30.9.)

Na začiatku projektu som si vyberal tému. Premýšľal som nad takou, ktorá by najviac mohla byť použiteľná v rámci projektu a súčasne mala potenciál. Definoval som si programovací jazyk, ktorý budem používať - Python

Popis riešenia – fáza2 (do 7.10.)

V rámci fázy 2 som sa venoval krokom, ktoré budú potrebné na úspešné vyhľadanie potrebných informácii. Postup som zhrnul do grafického pseudokódu.



Popis riešenia – fáza3 (do 21.10)

Vo fáze 3 som predviedol funkčný kód jednoduchý kód na malej vzorke dát.

Použil som python, konkrétne framework jupyter notebook, aby som mohol jednoduchšie pracovať s kódom.

V rámci fázy som používal na načítanie knižnicu pandas.

Mal som hlavný dokument `main_document.csv` v ktorom som mal konkrétne ochorenia. Vytvoril som si ďalšie svoje vlastné dokumenty, v ktorých som mal všetky krajiny, symptómy a spôsoby prenosu ochorenia. (ktoré som následne v texte vyhľadával).

Konkrétny postup:

1. Otvoril som si zazipovaný dokumnet (articles1.xml.bz2)
2. Dokument som čítal riadok po riadku, kontroloval či ide o riadok v ktorom sa nachádza <title> alebo </page>
3. Ak hodnoty `bacteria`, `virus` alebo `viral` v čítanom riadku neboli, riadok nespracovávalo.
4. Ak sa tam nachádzali, snažil som sa v title nájsť či má daný článok pre nás zmysel (či sa tam nachádza ochorenie z datasetu).
5. Ak sa nachádzalo, program vyhľadal v riadku symtómy, krajiny a spôsoby šírenia a vložil ich ku ochoreniu.

Popis riešenia – fáza4(do 22.11)

V ďalšej časti som použil spark. Používal som spark data frame. Do tohto data frame-u som si povkladal title a text ako dva stĺpce. V stĺpci text som následne pomocou vlastnej UDF funkcie v ktorej som používal regexy vyhľadával krajiny, symptómy, spôsoby šírenia, typ ochorenia. Pomocou spomínanej UDF funkcie som si počas vyhľadávania vytváral stĺpce symptoms, transmissions, type, country, s ktorým som postupne vkladal keywords, ktoré som na základe regexov vyhľadal. Typ ochorenia som určil na základe majority typu (ak tam našlo viac ochorenie typu bacteria, ochorenie som pridružil tzpu bacteria ak naopak bola častejšia hodnota virus, ochorenie som privlastnil typu virus. Zvyšné hodnoty ako country, symptoms, tansmissions som zobral zo stĺpcov a vložil do môjho hlavného dokumentu. Spark mi poslúžil na rozdistribuovanie dát na spracovanie na viacerých nodoch. Na základe toho som dosiahol lepšie výsledky ako iba použitím pandasu.

Spracovanie polovice datasetu trvalo približne 1h. Z toho môžeme predpokladať, že spracovanie celého datasetu by trvalo 2,3h s pandasom, so sparkom nám vykonanie trvalo 1h.

Zistili sme, že použitie sparku nám pomohlo v urýchlení spracovania.

Detaily finálneho riešenia

Postup vyhľadávania:

1. Vytvorenie spark session.
2. Definovanie potrebných ciest ku súborom.
3. Vytvorenie regex patternov na aplikovanie regexového vyhľadávania.
4. Načítanie vstupného súboru do spark data frame, ktorý sa načítava pomocou rowTagu, aby sa načítali iba potrebné informácie, teda priamo stránky. V spark data frame, nám reprezentuje jeden článok jeden riadok.
5. Filtrovanie potrebných stĺpcov -> zostanú iba stĺpce `title` a `revision.text._VALUE`
6. Filtrovanie riadkov podľa stĺpca `revision.text._VALUE`. Odstraňujú sa riadky (stránky), ktoré sú redirektované, keďže sú pre nás nepotrebné a taktiež stránky, ktoré v sebe nemajú regexy bacteria, virus a viral, keďže také sú pre nás taktiež nepotrebné. Záznamy sme odstraňovali, aby sme zvýšili efektívnosť ďalšieho hľadania.
7. Vytvorili sme si UDF funkcie, aby sme mohli vyhľadávať pomocou regexov nad jednotlivými stránkami (riadkami v data frame)
8. Postupne spúšťame prehľadávanie nad textom stránky, vytvárame príslušné stĺpce do ktorých vkladáme nájdené data.
9. Vymažeme stĺpce _VALUE a title, takže nám ostanú už iba relevantné stĺpce disease, type, symptoms, countries a transmissions.
10. Nasleduje cyklus, v ktorom postupne vyberám zo spark data frame spracované stránky (jednotlivé riadky) a ukladám ich do môjho hlavného dokumnetu, pričom ak sa ku ochoreniu už dáta nachádzali, pridá nové informácie o ochorení.
11. Uloženie hlavného dokumentu

Postup indexovania:

1. Načítal som si hlavný dokument.

2. Pre každú kategóriu som vytvoril vlastné dictionary, kde kľúč je hladaná kategória a value je pole indexov na riadky v hlavnom dokumente. Teda mám napríklad pre typ ochorenia naslednovné - {virus:[1,25,120], bacteria:[20,50,180]. Na základe indexov riadkov si potom viem zistiť názov ochorenia.
3. Dictionaries si následne pomocou knižnicke pickle uložím.

Postup main:

1. Načítanie hlavného dokumentu a indexov
2. Nasleduje interakcia s používateľom. Najprv si vyberie kategóriu, z nej sa vypíšu možnosti. Následne si vyberie jednu možnosť a program zobrazí všetky ochorenia prislúchajúce výberu.

Popis riešenia – fáza5 (do 16.12)

V rámci fázy 5 som vytváral unit testy, dokumentáciu v kóde ako aj samostatnú dokumentáciu projektu v pdf a používateľskú príručku

Testovanie

Dokopy som vykonal 3 unit testy, ktoré sa nachádzajú v jupyter notebooku. Nachádza sa v priečinku source_code. Dôležité súbory ku testovaniu sú v ďalej v priečinu tmp_test. Kde v prvom teste kontrolujem spracovanie indexov, v druhom kontroluje správne načítanie indexov pre program a v poslednom kontrolujem spracovanie raw spark data frame, kde vyhľadávam pomocou regexov.

Zmeny výsledného projektu oproti zadaniu

Vyhľadanie liekov na liečbu ochorenia nevložené vzhľadom na významnosť z hľadiska vyhľadávania informácii. V rámci riešenia by postačilo iba vložiť cudziu databázu liekov.

Porovnanie výsledkov

Vzhľadom na to, že som nenašiel podobné riešenia, bolo porovnávanie v oblasti precission a recall nerelevantná.

Záver

Vrámci projektu som sa naučil vyhľadávať informácie z wikipédie, vyskúšal som si postupy, ktoré sa pri vyhľadávaní používajú. Využil som aj často používaný framework v danej oblasti Spark resp. jeho verziu PySpark vytvorenú pre jazyk Python, ktorá mi výrazne pomohla v rýchlosti vyhľadávania vďaka rozdistribuovaniu hľadania. Rád som sa venoval vyhľadávaniu o to viac, že som ho mohol aplikovať na projekt z oblasti, ktorá ma zaujíma.

Príloha A

Používateľská príručka

Zdrojové kódy sú vo forme jupyter notebookov, ktoré si je potrebné spustiť. Nachádzajú sa v priečinki VINF/source_code. Nachádzajú sa tu štyri zdrojové kódy.

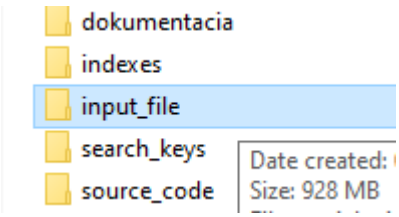
1. VINF_vyhladavanie obsahuje hlavný kód pre vyhľadávanie.

```
In [1]: # File system management
        import os
        import sys
        import re
        import pyspark
        from pyspark import jars
        import pandas as pd

In [2]: SOURCE_FILE_PATH = "articles.xml"

In [3]: # Setting up spark session
```

V druhej bunke sa nachádza názov súboru, ktorý sa má prečítať (rozbalený vo formáte xml). Súbor je potrebné vložiť do priečinku input_file.



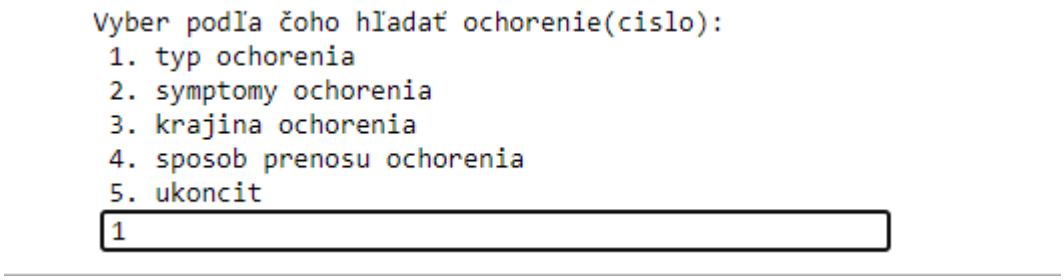
2. Vytváranie indexov je dostupné pomocou súboru VINF_indexy. Stačí ho spustiť a indexy sa automaticky povytvárajú podľa hlavného súboru `main_document.csv` v priečinku source_files.

3. Testovanie je dostupné v súbore VINF_testovanie, kde po každom teste test vypýše buď Test succeeded alebo Test failed.

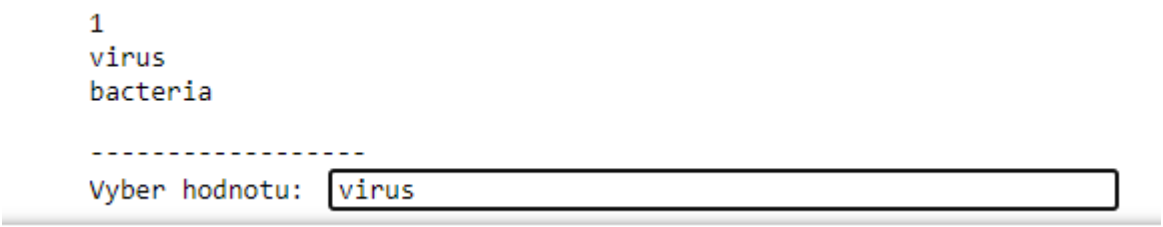
```
Test succeeded
```

4. V hlavnom súbore nazvanom VINF_main obsahuje hlavné používateľské rozhranie.

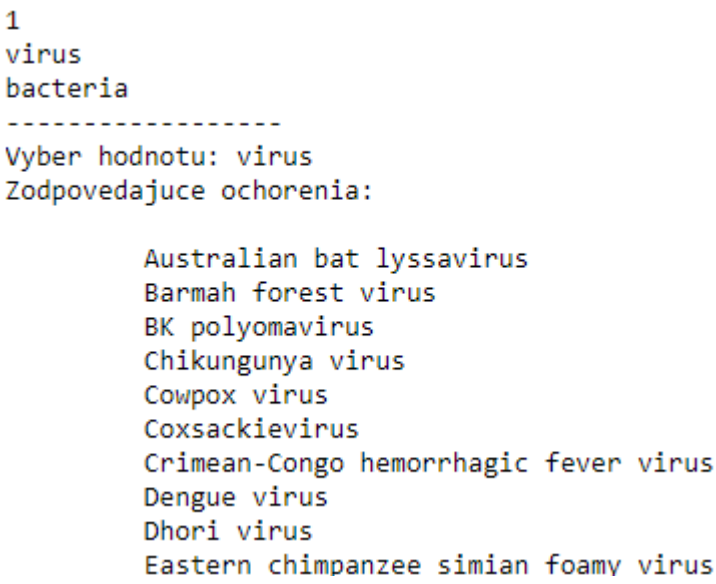
   Súbor je potrebné spustiť. Pod poslednou bunkou sa zobrazí menu.



Je potrebné si vybrať zadaním príslušného čísla. Následne je potrebné kliknúť enter.



Následne je potrebné zadať či chceme filtrovať ochorenia na základe virusu alebo bakterie (pre každú kategóriu je výber individuálny).



Po zadaní hodnoty sa zobrazia všetky ochorenia zodpovedajúce filtru.

Následne sa znova objaví menu ako na začiatku.

Návod na inštaláciu

V rámci requirements.txt sa nachádzajú všetky potrebné knižnice.

Pred spustením je potrebné stiahnuť virtuálne prostredie:

https://drive.google.com/drive/folders/1iRL_LcsA3ry0bSl38mFzP4gTY0xizszf?usp=sharing

Treba si ho aktivovať pomocou príkazu vinfenv\Scripts\activate

V ňom si následne treba spustiť prostredie jupyter notebook pomocou príkazu `jupyter notebook`. Následne si používateľ nájde, otvorí svoj súbor a spustí v prostredí jupyter notebook.

V prípade, žeby nebolo možné spustiť spark session, je potrebné Spark samostatne nainštalovať a nastaviť príslušné environment variables SPARK_HOME a HADOOP_HOME.

Stiahnutie spark: https://spark.apache.org/downloads.html