Autor: Martin Kukučka

Správa k projektu

Problém - zadanie

Sparsovanie osôb wikipédie, vytvorenie jednoduchej služby 'mohli sa stretnúť?', ktorá po zadaní dvoch mien určí, či sa mohli dané osoby stretnúť (prekryv času ich života)

Motivácia

Prvotným podnetom motivácie bol fakt, že verejne dostupný nástroj s touto funkcionalitou som nenašiel a manuálne prechádzanie Wikipédie, hľadanie a porovnávanie údajov a získanie výsledkov je neefektívne. Preto vidím potenciál poskytnúť používateľovi nástroj, ktorý automaticky vyhodnotí, či sa osoby mohli stretnúť na základe prekrytia dátumov, prípadne výšku pravdepodobnosti ich stretnutia na základe prekryvu miest. Týmto nástrojom je možné zautomatizovať, uľahčiť a zefektívniť vyhľadávanie, či sa dve osoby mohli stretnúť. Nástroj tohto typu by si na stránke Wikipédie našiel uplatnenie ako nová funkcionalita, prípadne aj inde, po vhodnom prispôsobení.

Cieľ

Vytvorenie nástroja, ktorý vie spracovať vstupný dataset, vyfiltrovať potrebné údaje, vhodne údaje zaindexovať a po zadaní informácií o dvoch osobách vyhľadať a určiť, či sa dané osoby mohli niekedy počas ich života stretnúť. Informácia o možnosti stretnutia / nestretnutia bude vychádzať z datasetu z údajov:

- dátumy narodenia
- dátumy úmrtia
- lokácie miesta narodenia a miesta úmrtia, ktoré by mohli dopomôcť určiť výšku pravdepodobnosti stretnutia

Na základe týchto údajov nástroj určí, či sa osoba A mohla stretnúť s osobou B.

Existujúce riešenia

Žiadne verejne dostupné existujúce riešenia ani algoritmy som nenašiel.

Popis riešenia

Riešenie sa skladá z 3 častí:

- Parser
- Indexer
- Searcher

Parser

Parser pracuje paralelne pomocou Apache Spark. Parser prechádza v XML datasete po jednotlivých stránkach (tag <page>), ktoré spracováva.

Na spracovanie údajov bolo nutné vytvoriť list vhodných infoboxov, ktoré boli vyfiltrované priamo z datasetu enwiki-latest-pages-meta-current11.xml-p5399367p6899366.bz2 a doplnené podporovaným listom infoboxov z Wikipédie

https://en.wikipedia.org/wiki/Wikipedia:List_of_infoboxes pod sekciou Person, celkovo tak list obsahuje 240 rôznych typov infoboxov pre osoby.

Prechádzaním po stránkach sa vezmú tagy <id> a <title> a v tagu <text> sa vyhľadávajú vhodné infoboxy, z ktorých sa následne extrahujú potrebné údaje: mená, dátum narodenia, dátum úmrtia, miesto narodenia, miesto úmrtia. Údaje sa následne ukladajú do Spark Datasetu (DataFrame).

Na získanie týchto údajov bolo nutné použiť v prvom rade regulárne výrazy a Java SDK funkcie na prácu s textom.

Na konci sa už spracované údaje zo Spark Datasetu uložia vo formáte .json vo viacerých súboroch v jednom priečinku. Jeden riadok .json súboru obsahuje údaje o jednej osobe.

{"id":6212623,"title":"Cole Konrad","names":["Cole Konrad"],"birth_date":"1984-04-02","death_date":"null","birth_place":"Appleton, Wisconsin","death_place":"null"}

Obrázok 1 ukážka uloženia údajov o osobe

Indexer

Po spracovaní údajov je na rade indexer, ktorý beží na Apache Lucene a získané údaje z parsera zaindexuje. Indexer prechádza v priečinku po všetkých .json súboroch, ktoré číta po riadkoch (1 riadok = 1 osoba) a jednotlivo vytvára dokumenty, ktoré pri opätovnom spustení indexera aktualizuje na základe parametra id článku (tag <id>).

Searcher

Searcher beží na Apache Lucene a slúži na vyhľadávanie osôb v zaindexovaných súboroch a vykonávanie hlavnej funkcionality programu - vyhľadanie, či sa dve osoby mohli stretnúť. Vyhľadanie prebieha na základe používateľom zadanej query, nájdu sa všetky vhodné výskyty a vezme sa výskyt s najväčším skóre (najrelevantnejší výskyt). Ak searcher nájde vhodné výskyty osôb v indexe, tak sa prechádza k porovnaniu dátumov a ak sa osoby podľa dátumov mohli stretnúť (existuje prekryv ich života), tak sa prechádza k porovnaniu miest, ktoré určujú výšku pravdepodobnosti stretnutia na základe podobnosti (rovnaké miesto = vyššia pravdepodobnosť). Na konci vypíše searcher výsledok vyhľadávania stretnutia osôb a čaká na ďalší vstup.

Použitý softvér

Riešenie je implementované v programovacom jazyku Java.

Na paralelné spracovanie datasetu bol použitý Apache Spark.

Na indexovanie a vyhľadávanie nad indexom bol použitý Apache Lucene.

Bližšie informácie sú v sekcií Spustenie, inštalácia softvéru, použitie softvéru.

Popis problémov ktoré sa vyskytli

Najčastejším problémom bolo extrahovanie údajov pomocou regulárnych výrazov, ktoré bolo nutné neustále upravovať, aby pokrývali väčšie množstvo rôznych typov zápisu a vhodnejšie výsledky.

V datasete sa vyskytuje viacero rôznych formátov dátumov narodenia a úmrtia (1971; 10 31, 1912; July 1875...), ktoré bolo nutné zjednotiť a podľa potreby previesť na číselný formát. Celkovo program podporuje 7 typov zápisu dátumu, ktoré vie rozpoznať. Program vie rozpoznať textový zápis mesiaca v dátume, ktorý prevedie na číselný formát.

Ďalším problémom s dátumami boli dátumy vo formáte storočia (19th century), alebo vo formáte pred naším letopočtom (675 BC), ktoré už program vie rozpoznať a vhodne spracovať.

Problémom bolo aj načítavanie XML datasetu, ktorý fungoval cez DOM parser, čiže sa celý dataset načítal do pamäte, čo bolo neefektívne a neskôr bol tento prístup zmenený na SAX parser, ktorý dataset načítaval len podľa tagu <page>. Pri paralelnom spracovaní sa dataset načítava cez Spark podľa tagu <page>.

Popis dát

Počas implementácie programu bol program testovaný na viacerých wiki dump datasetoch.

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

- enwiki-latest-pages-articles11.xml-p6899367p7054859.bz2 - 362 MB
- enwiki-latest-pages-meta-current1.xml-p1p41242.bz2 - 968 MB
- enwiki-latest-pages-meta-current11.xml-p5399367p6899366.bz2 – 3.7 GB
- enwiki-latest-pages-articles.xml.bz2 – (celá Wikipédia) – 80.6 GB

Ako finálny dataset bol zvolený dataset obsahujúci celú WIkipédiu: enwiki-latest-pages-articles.xml.bz2 o veľkosti 80.6 GB.

Spracované data sú uložené vo formáte .json. V jednom súbore je uložených viacero osôb a na jednom riadku sú uložené údaje o jednej osobe:

- id – id článku (StringField)
- title – názov článku (TextField)
- names – zoznam mien (meno, alternatívne meno, meno narodenia…) (TextField)
- birth_date – dátum narodenia vo formáte yyyy-MM-dd (StringField)
- death_date – dátum úmrtia vo formáte yyyy-MM-dd (StringField)
- birth_place – miesto narodenia (region, miesto, štát ak existuje) (TextField)
- death_place – miesto úmrtia (region, miesto, štát ak existuje) (TextField)

Ak niektorý z údajov nebol nájdený, tak je mu defaultne priradená hodnota “null”. Prípadne ak je v dátume uvedený len rok, tak datum sa uloží v tvare YYYY-01-01.

Ukážkové dáta po spracovaní datasetu možno nájsť na stránke predmetu ako *ukážka dát* v sekcií Prílohy na stránke predmetu.

Vyhodnotenie a overenie riešenia

Testovanie riešenia prebiehalo počas celého vývoja program, vďaka čomu sa program neustále vylepšoval. Na ukážku overenia bolo vybraných pár testovacích prípadov.

Spustenie, inštalácia softvéru, použitie softvéru

Finálne riešenie sa skladá z 3 častí:

- Parser
- Indexer
- Searcher

Každú časť je možné spustiť nezávisle od seba.

Autor: Martin Kukučka

Podmienky spustenia

Na spustenie programu sú nutné:

- Java SDK 1.8.0_311
- Apache Spark 3.1.2
- Apache Lucene 7.5.0
- Scala 2.12.10
- Spark-XML 2.12-0.5.0
- org.joda.time
- org.json.simple

Parametre spustenia

| - medzi jednotlivými parametrami použité na lepšie rozlíšenie, inak predstavuje medzeru

Parser

- parametre spustenia: dataset (vo formáte XML) | priečinok na uloženie vyfiltrovaných údajov | txt súbor s vhodnými typmi infoboxov

- výstup: výstupný priečinok na uloženie vyfiltrovaných údajov

Indexer

- parametre spustenia: priečinok s vyfiltrovanými údajmi | priečinok na uloženie indexu

- výstup: výstupný priečinok na uloženie indexu

Searcher

- parametre spustenia: priečinok s indexom

- vstup: query
- výstup: výpis vhodných výsledkov vyhľadania osôb, údaje o zvolených osobách, výsledok, či sa osoby mohli stretnúť

Spustenie

Program je rozložený do štyroch súborov: Parser.java, Indexer.java, Searcher.java, Person.java a na spustenie je nutné splniť podmienky spustenia a zahrnúť potrebné knižnice. Spustenie prebieha cez vývojové prostredie, z ktorého sa dá bez problémov spúšťať jednotlivé časti zadania, prípadne je možné vytvoriť .jar súbory týchto častí a spustiť ich cez CMD. .jar súbory som nevytváral, pretože presahovali pamäťový limit na nahranie na stránku predmetu.

Časti Parser a Indexer nevyžadujú používateľský vstup, iba je nutné nastaviť potrebné parametre spustenia.

Searcher okrem parametru spustenia vyžaduje aj používateľský vstup. Searcher beží v cykle a od používateľa v cykle vyžaduje 2 vstupy / query: údaje o prvej osobe a údaje o druhej osobe. Ak používateľ nechce ďalej vyhľadávať a chce ukončiť program, tak stačí napísať *quit* do ktoréhokoľvek vstupu.

Query pracuje na základe Apache Lucene a medzi jednotlivými údajmi podporuje operátory ako AND, OR...

Autor: Martin Kukučka

Do query môžu vstupovať tieto údaje, vhodne oddelené operátormi:

- names
- birth_date
- death_date
- birth_place
- death_place

Príklad query:

- names:"carl froch" AND birth_place:"nottingham"
- names:oswaldo AND birth_place:venezuela

```
Query - first person: names:"carl froch" AND birth_place:nottingham
Total results: 1
The most relevant results:
 - Carl Froch, Score: 24.315193
Selected person: Carl Froch

Query - second person: names:oswaldo AND birth_place:venezuela
Total results: 7
The most relevant results:
 - Oswaldo Olivares, Score: 19.188406
 - Oswaldo Arcia, Score: 19.188406
 - Oswaldo Vizcarrondo, Score: 18.624403
 - Oswaldo Quevedo, Score: 18.258972
 - Oswaldo Vigas, Score: 18.14339
 - Oswaldo Peraza, Score: 17.368177
 - Oswaldo Navarro, Score: 17.368177
Selected person: Oswaldo Olivares

FIRST PERSON
Name: Carl Froch
Alternative names: Carl Froch, Carl Martin Froch, The Cobra
Date of birth: 1977-07-02
Date of death: null
Place of birth: Nottingham, England
Place of death: null

SECOND PERSON
Name: Oswaldo Olivares
Alternative names: Oswaldo Olivares
Date of birth: 1953-09-15
Date of death: null
Place of birth: Caracas, Venezuela
Place of death: null

Carl Froch / Oswaldo Olivares could meet (TRUE), if they are / were alive at the same time, (date of death doesn't exist)
with a LOW probability, locations visited during their lives are different
```

Autor: Martin Kukučka

Zhodnotenie

Riešenie bolo úspešne vypracované a splnilo stanové ciele. Úspešné dokončenie nástroja, ktorý po zadaní dvoch mien určí, či sa dané osoby mohli niekedy počas ich života stretnúť. Program podporuje spracovanie Wikipédia XML datasetu, z ktorého vie extrahovať a uložiť potrebné údaje, ktoré dokáže zaindexovať a nakoniec nad nimi aj vyhľadávať a vyhodnocovať, či sa osoby mohli stretnúť a taktiež určiť aj výšku pravdepodobnosti stretnutia.

Porovnanie top 10 výsledkov programu s Wikipédiou

Spracovanie dumpu celej Wikipédie trvalo približne dve a pol hodiny a bolo nájdených 1371359 záznamov osôb.

Porovnanie prebiehalo na 20 pokusoch zadaním rôznych mien, jednoslovných krstných mien, celých mien aj s priezviskami do programu, ktorého výsledky (top 10 výskytov) boli porovnávané s top 10 výsledkami Wikipédie. Zhoda bola v 54 prípadoch z celkového počtu 150 prípadov, program aj Wikipédia mala v top 10 výskytoch 54-krát zhodu, čo predstavuje úspešnosť 36%.

Pri zadávaní iba krstných mien (Zuzana, Albert) bola zhoda nižšia, pretože existuje väčší rozsah osôb a Wikipédia obsahuje funkcionalitu, že pri mene Albert ako prvé ponúkne meno Albert Einstein, ale v mojom programe sa dostal na vrch Albert Anker.

Pri dlhších zadaní mien aj s priezviskami (Peter Parker, Pablo Escobar) bola úspešnosť zhody vyššia, pretože sa zmenšil rozsah osôb a v takýchto prípadoch bola zhoda na vybraných vstupoch 50-100%.

Pre československé mená (Zuzana, Róbert, Vojtech) bola úspešnosť zhody takmer 50%.

Ďalšie porovnanie overovalo, či osoby uvedené na Wikipédií dokáže nájsť aj môj program. Vykonalo sa 25 testov, či osoby z Wikipédie, ktoré program nezobrazil v top 10 výsledkoch, vie nájsť. Program našiel 20 osôb a 5 osôb nevedel nájsť. Úspešnosť bola 75%.

Možné vylepšenia

- okrem vyhľadávania len nad infoboxom, by ako možné vylepšenie mohlo byť vyhľadávanie aj vo voľnom texte
- zjednotenie miest narodenia a miest úmrtia na základe štátu (napríklad pridaním externého datasetu obsahujúci mestá a k nim štát),
  t.j. Berlin, Mníchov → Nemecko, takýmto štýlom by mohla byť pokrytá väčšia časť, čo by vylepšilo porovnanie výšky pravdepodobnosti, pretože momentálne program funguje tak, že ak osoba A má lokáciu Berlín a osoba B má lokáciu Mníchov, tak sa tu nenachádza prekryv miest, a tým pádom by program zhodnotil nízku pravdepodobnosť stretnutia