Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Bc. Marek Adamovič

Parsovanie receptov z wikipédie

Vyhľadávanie Informácií

Cvičenie: Utorok 19:00 – 20:40

Ing. Igor Stupavský 2022/2023

Popis projektu

Náš projekt parsuje recepty a ingrediencie z wikipédie. Následne umožňuje používateľovi zadať suroviny (napríklad také, ktoré má doma) a program mu vráti 3 recepty s najlepšou zhodou, kde sa tieto suroviny nachádzajú. Druhá možnosť pre používateľa je vyhľadať všetky recepty pre konkrétnu surovinu.

Program má viacero možných nastavení, vďaka čomu nemusí pri každom spustení prehľadávať celú wikipédiu (čo mu trvá okolo dvoch hodín). Je tu možnosť uloženia/nahratia vyparsovaných receptov z/do programu, vďaka čomu má používateľ výsledok do pár sekúnd.

Ako motiváciu pre náš projekt berieme, že v dnešnej dobe je veľkým problémom vyhadzovanie jedla v domácnosti. To môže byť čiastočne spôsobené tým, že ľudia nevedia, čo môžu navariť zo surovín, ktoré doma majú. Vďaka našej aplikácií vedia veľmi jednoducho zadať suroviny, ktorých sa „potrebujú zbaviť“ a o chvíľu majú návrhy receptov, ktoré s nimi môžu vyskúšať.

Súčasné riešenia problému

Aplikácia pre odporúčanie receptov [1] NTLK pre lematizáciu surovín. Tiež využíva túto technológiu na odstránenie „stop words“, teda najviac používané slová v jazyku (napríklad spojky). Autor taktiež popisuje potrebu preprocesingu dát, v ktorých sa nachádzajú nechcené interpunkčné znamienka a podobne.

Taktiež sme sa stretli [1, 2] s technológiou Word2Vec, ktorá vytvára zo slov vektory, vďaka ktorým vieme potom natrénovať modely strojového učenia, ktoré budú chápať nielen samotné slová, ale aj kontext ku nim. K tomu využíva Word2Vec aj neurónové siete.

Moje riešenie

Pre moje riešenie som zvolil skriptovací jazyk python s použitím frameworku Apache Hadoop a Spark pre paralelné spracovanie veľkých dát.

Knižnice a technológie, ktoré som v projekte použil:

- Json -> pre ukladanie / načítanie vyparsovaných dát
- Nltk -> na lematizáciu textu wikipédie
- Time -> meranie času (napríklad pre porovnanie rôznych prístupov)
- Pyspark -> paralelné spracovanie dát
- Docker -> pre ľahšie spúšťanie programu na rôznych strojoch

Projekt sme začali riešiť lineárnym spracovaním dát wikipédie (dump .bz2 súbor) a mimo dockeru. Vyhľadávali sme stránky receptov a to pomocou kľúčových slov, pri ktorých je veľká šanca, že sa jednalo o recept (ingredients, salt). Keď sme zistili, že sa jedná o recept, vyparsovali sme z neho názov a ingrediencie pomocou regexu. Keď fungovala prvá verzia programu, otestovali sme ju na menších dátach. Následne sme pridali blacklist slov, ktoré sa nemôžu nachádzať v ingredienciách, respektíve v názvoch, aby sme mali presnejšie výsledky. Ďalším krokom bolo zrýchlenie parsovania. To sme skúsili dosiahnuť pomocou python vlákien (threadov). Keďže pri parsovaní je bottleneck sústredený na CPU, nedostali sme lepší výsledok. Preto sme sa rozhodli pre multiprocessing pomocou python procesov. Využili sme návrhový vzor producer a consumer. Teda jeden proces čítal dáta z wikipédie a vyberal z nich recepty, zatiaľ čo druhý proces z nich vyberal ingrediencie a ukladal do python štruktúry. Týmto spôsobom sme dosiahli cca 40% zrýchlenie oproti pôvodnému času. Potom sme sa dozvedeli, že projekt musí obsahovať technológie pyspark alebo hadoop. Prekopali sme logiku programu, aby využíval tieto technológie a taktiež sme využili docker, aby sme ušetrili čas pri spustení projektu na iných strojoch. Oproti pôvodnému času sme dosiahli zrýchlenie až okolo 56%. Po sfunkčnení programu v kontajneri sme optimalizovali dockerfile a docker-compose súbory, aby sme museli vykonať čo najmenej operácií pre štart programu. Následne sme pridali inverzný index, ktorý nám pre každú jednu surovinu zoskupí všetky recepty, ktoré túto surovinu obsahujú. Vďaka tomu vieme používateľovi umožniť vybrať surovinu a vrátiť mu všetky recepty s touto surovinou (vyhľadávanie v inverznom indexe) V závere sme pridali unit

testy, ktoré kontrolujú funkčnosť funkcií starajúcich sa o vytváranie inverzného indexu a vyhľadávanie v ňom. Keďže v pysparku nevieme používať klasický input, používateľ musí zadávať svoje vstupné parametre do konštánt, ktoré sa nachádzajú navrchu súboru main.py.

Použité dáta

Pre náš projekt sme použili kompletný dump textových dát z wikipédie vo formáte *.bz2 vo veľkosti skoro 38GB (v skomprimovanej forme). Pre testovanie sme taktiež využili čiastočný dump wikipédie o veľkosti približne 200MB (v skomprimovanej forme).

Spustenie

- Stiahneme repozitár so všetkými potrebnými súbormi (https://github.com/pasavec008/vinf_zadanie1)
- Vnútri repozitáru zadáme príkaz docker compose up
- Počkáme, kým sa nám vytvorí kontajner
- Napojíme sa na kontajner (najlepšie pomocou vs code funkcionality -> attach this to running container), aby sme v ňom vedeli zadávať príkazy
- Ak chceme parsovať recepty z wikipédia dump súboru (ak nie, tento krok preskočíme), potrebujeme presunúť tento dump súbor do hadoop priečinku pomocou príkazu (súbor sa musí nachádzať v hlavnom priečinku, kde je main.py):

hadoop fs -put /vinf_recipes/<názor_súboru>.bz2 /user/root

- V main.py si môžeme nastaviť konfiguráciu v konštantách na vrchu súboru
- Spustíme main.py pomocou príkazu:

spark-submit main.py

Príklady

1. Vyparsovanie receptov z malého testovacieho súboru a vyhľadanie receptov s ingredienciami olej, zemiak a soľ:

```
#example input for parsing
USER_INPUT = ['r', 's', 'x']

# specify ingredients for recommending
RECOMMENDING_USER_INPUT = ['oil', 'potato', 'salt']
```

```
Parsing recipes from raw wiki data completed in time:  87.22  seconds.
30 recipes parsed successfully from raw wiki file.

User input s
Your 3 recommended recipes:

Recipe name:  French fries
Ingredients:  ['potato', 'oil', 'salt']
Number of matched ingredients:  3 / 3

Recipe name:  Bubble and squeak
Ingredients:  ['potato', 'cabbage']
Number of matched ingredients:  1 / 2

Recipe name:  Miso
Ingredients:  ['fermentation', 'fermented', 'soybean', 'salt', 'kōji', 'aspergillus', 'oryzae']
Number of matched ingredients:  1 / 7

User input x
```

2. Načítanie vyparsovaných dát z json súboru (z celej wikipédie, dokopy skoro 2000 receptov), zapísanie načítaných dát do json súboru, ich znovu načítanie, vyhľadanie receptov so surovinami olej, zemiak, soľ, vytvorenie surovinového inverzného indexu a vyhľadanie v tomto indexe (nájde všetky recepty pre jednotlivé suroviny)

```
USER_INPUT = ['f', 'w', 'f', 's', 'ci', 'si', 'x']

# specify ingredients for recommending
RECOMMENDING_USER_INPUT = ['oil', 'potato', 'salt']

# if you have specified 'is' in USER_INPUT, you also need INDEX_SEARCH_INPUT
# which are ingredients and index search will find you all foods containing that ingredient
INDEX_SEARCH_INPUT = ['salmon', 'caramel', 'potato']
```

```
User input f
1858 recipes loaded successfully from parsed json data.

User input w
1858 recipes saved successfully to json.

User input f
1858 recipes loaded successfully from parsed json data.

User input s
Your 3 recommended recipes:

Recipe name:  French fries
Ingredients:  ['potato', 'oil', 'salt']
Number of matched ingredients:  3 / 3

Recipe name:  Alu Bharta
Ingredients:  ['potato', 'oil', 'onion', 'belle', 'pepper', 'salt']
Number of matched ingredients:  3 / 6

Recipe name:  Milcao
Ingredients:  ['potato', 'chiloé', 'lard', 'salt', 'vegetable', 'oil']
Number of matched ingredients:  3 / 6
```

```
User input ci
Recipes index created success
User input si
Ingredient salmon is in these recipes: ['Gravlax', 'Kalakukko', 'Smoke and Mirrors']
Ingredient caramel is in these recipes: ['Huangqiao sesame cake', 'Salted Nut Roll', 'Street rhyme']
Ingredient potato is in these recipes: ['Afritada', 'Ajdov Kruh', 'Akhni', 'Aloo gosht', 'Aloo tama', 'Alu Bharta', 'Batata
vada', 'Bedfordshire clanger', 'Birnen, Bohnen und Speck', 'Biłgoraj pierogi', 'Bolo do caco', 'Book of Isiah', 'Boxty', 'Bu
bble and squeak', 'Caldo verde', 'Chakapuli', 'Champ (food)', 'Chelfray', 'Cholent', 'Chukauni', 'Clam chowder', 'Clapshot',
 'Coddle', 'Confirmed bachelor', 'Corn chowder', 'Croquette', 'Dabeli', 'Dhoper chop', 'Ducana', 'Duchess potatoes', 'Dumpli
ng', 'Essex Township, Clinton County, Michigan', 'Falukorv', 'Fishcake', 'French fries', 'Fritas de prasa', 'George Emery Do
rsey', 'Gnocchi', 'Goulash', 'Halušky', 'Home International Championship', 'Irish stew', 'Iron sulphide', 'Italian hot dog',
 'Japanese curry', 'Jeera aloo', 'Kal-guksu', 'Kartoffelkäse', 'Kathryn McKinley', 'Kenpi', 'Keria gula melaka', 'Khanom kha
i hong', 'Khinkali', 'Kluchy połom bite', 'Klöße', 'Kopytka', 'Kreplach', 'Kroppkaka', 'Kugel', 'Latkes', 'Leek soup', 'Lefs
e', 'Lignes Aériennes Latécoère', 'Lokshen', 'MSN (gene)', 'Masala dosa', 'Mashed potato', 'Massaman curry', 'Mfarakeh', 'Mi
chelle Fulton', 'Milcao', 'Mugoyo', 'Oromo (dish)', 'Palacio Municipal Deportes San Pablo', 'Papa a la huancaína', 'Papadam'
, 'Pasty', 'Patatnik', 'Patriotic soup', 'Pickert', 'Pininyahang manok', 'Pitepalt', 'Pommes Anna', 'Portal:Liverpool', 'Pot
ato bread', 'Potato doughnut', 'Prażonki', 'Pyzy (dish)', 'Rappie pie', 'Robert Henry Lowie', 'Rumbledethumps', 'Rösti', 'Rē
wena bread', 'Sabudana vada', 'Salt potatoes', 'Samosa', 'Schupfnudel', 'Senate bean soup', 'Sha Tin Government Secondary Sc
hool Alumni Association', "Shepherd's pie", 'Smoke and Mirrors', 'Sopa teologa', 'Stamppot', 'Steckrübeneintopf', 'Stegt flæ
sk', 'Stew', 'Styrian sour soup', 'Taramasalata', 'Tattie scone', 'Template talk:Banking panics in the United States', 'Tour
tière', 'Trial-and-error method', 'US 1st Marine Division', 'VPL Champlain', 'Vada pav', 'Venerable Thomas Tunstall', 'Watch
ing television', 'Welsh cob', 'Wikipedia:Votes for deletion/Rattle puzzle', 'Zuppa toscana']
```

Zdroje

1.https://towardsdatascience.com/building-a-recipe-recommendation-system-297c229dda7b

2.https://dl.acm.org/doi/pdf/10.1145/3428757.3429096