Systém monitorování hashtagů

Tibor Pánik

Bakalářská práce
2019

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Tibor Pánik**
Osobní číslo:	**A16487**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační a řídicí technologie**
Forma studia:	**Kombinovaná**
Téma práce:	**Systém monitorování hashtagů**

Zásady pro vypracování

1. Vytvořte platformu pro analýzu sociální sítě Twitter.
2. Navrhněte databázi pro uchování dat.
3. Naprogramujte systém pro zpracování textu.
4. Vyhodnoťte zjištění statisticky v různých kontextech.
5. Reprezentujte finální produkty.

Rozsah bakalářské práce:
Rozsah příloh:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. RA, S., PUJARI, J., SHREENIVAS BHAT, V., DIXIT, A. Procedia Computer Science: Timeline Analysis of Twitter User, 2018, vol. 132, 157-166
2. SYMEONIDIS, S., EFFROSYNIDIS, D., ARAMPATZIS, A. Expert Systems With Application: A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis, 2018, vol. 110, 298?310
3. RATHAN, M., VISHWANATH, R., HULIPALLEDA, K. R., VENUGOPALB, L. M. Patnaik. Applied Soft Computing: Consumer insight mining: Aspect based Twitter opinion mining of mobile phone reviews, 2018, vol. 68, 765-773
4. SHRAVAN KUMAR, B., VADLAMANI, R. A survey of the applications of text mining in financial domain, Knowledge-Based Systems, 2016, vol. 114, 128?147
5. Tweepy. Tweepy [online]. Dostupné z: http://www.tweepy.org/
6. spaCy ? Industrial-strength Natural Language Processing in Python. spaCy ? Industrial-strength Natural Language Processing in Python [online]. Copyright ? 2016 [cit. 23. 11. 2018]. Dostupné z: https://spacy.io/

Vedoucí bakalářské práce: **Ing. Bc. Pavel Vařacha, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **20. prosince 2019**
Termín odevzdání bakalářské práce: **15. května 2020**

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Ve Zlíně dne 20. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor;
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne …….

podpis

diplomanta

ABSTRAKT

PÁNIK, Tibor: Systém monitorování hashtagů. Bakalárska práca. Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky. Vedúci práce: Ing. Pavel Vařacha, Ph. D. Stupeň odbornej kvalifikácie: bakalár. Zlín : FAI UTB, 2019.
Klíčová slova: tweet, Kibana, NLTK

Cieľom bakalárskej práce je vytvorenie platformy pre analýzu sociálnej siete Twitter. Na základe dostupnej literatúri bol vytvorený skript, napísaný v programovacom jazyku Python, ktorý slúži pre zber dát. Skript má taktiež za úlohu upraviť dáta do vhodnej podoby k perzistentnému uloženiu v navrhnutom databázovom systéme. Na vybranom príklade je možné výsledky štatistického charakteru reprezentovať zvoleným vizualizačným nástrojom v rôznych formách.

Klíčová slova: tweet, Kibana, NLTK

ABSTRACT

The aim of the bachelor thesis is to create a platform for the analysis of the social network Twitter. Based on the available literature was created a script written in Python programming language, which is used for data collection. The purpose of the script is modifying the data into a suitable form for persistent storage in the designed database system. In the example of choice may the results of a statistical nature be represented with a visualization tool in various forms.

Keywords: tweet, Kibana, NLTK

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné

OBSAH

ÚVOD

Spracovanie užívateľského textu patrí medzi náročné úlohy. Žiadaný dátový výstup sa dá dosiahnuť aplikovaním rozličných postupov, nástrojov a techník. Vstupné dáta musia byť vhodným spôsobom predspracované. Ideálny postup spracovania užívateľské textu aktuálne neexistuje. Obmedzujúcimi faktormi sú potrebný čas na vykonanie potrebných úprav a komplexnosť úloh na zozbieraných dátach.

Sociálne siete sú preplnené užívateľskými dátami a rozhranie klienta umožňujú k nim prístup. Autor svoje dáta uverejňuje vo forme príspevku. Užívateľské príspevky majú najmä textovú podobu a venujú sa rôznym témam od priania k narodeninám, recenzie na film až k spravodajských informáciam od overených zdrojov. Cieľom príspevku je byť viditeľný a zdieľaný medzi užívateľmi. Príspevok obsahuje aj ďalšie charakteristické informácie.

Twitter je jedným z bohatých zdrojov užívateľských dát. Demonštratívnym príkladom ukážeme aktivitu užívateľov na celosvetovo momentálne veľmi diskutovanú tému koronavírusu.

Prvým krokom je vytvorené softvéru zachytávajúce príspevky užívateľov v reálnom čase. Komunikáciu bude zabezpečovať modul komunikujúci s API danej sociálnej siete. Knižnice, ktoré ponúkajú zdarma potrebný interface, je k dispozícií niekoľko. Zozbierané dáta budú, po vhodnom predspracovaní, perzistentne uložené. Kvôli povahe dát budú ukladané do nerelelačnej databázy. Výsledky budú reprezentované vo forme grafov a máp pomocou zvoleného nástroja.

Z výsledkov bude možné vyčítať rakcie užívateľov na zvolenú tému, s akou frekvenciou prispievajú a kde vo svete je daná téma aktuálna v danom čase.

I. TEORETICKÁ ČÁST

1 CIELE

Cieľom práce je vytvorenie systému, ktorého úlohou je zber dát zo sociálnej siete Twitter a tieto dáta následne reprezentovať. Čiastkové ciele predstavujú jednotlivé úkony, ktoré je potrebné vykonať k dosiahnutiu požadových výsledkov.

1.1 Vytvoriť platformu na analýzu sociálnej siete Twitter

Skript v jazyku Python musí zabezpečovať neustály príjem nových dát pomocou vybraného klienta Twitter API. Teoretická časť sa zaoberá rozdielmi medzi dostupnými variantmi klientov v kapitole 3 a jej podkapitolách 3.1 a 3.2. Samotný výber je riešený v praktickej časti, kapitola 8.1. Prijaté dáta musia byť vhodne predspracované technikami na spracovanie uží ateľského textu pred uložením do databáze. Dáta musia obsahovať okrem samotného textu aj ďalšie informácie, ktoré budú využité pri štatistických reprezentáciách.

1.2 Návrhnúť databázu pre uchovanie dát

Relačná databáza nie je vhodná pre neštruktúrovaných dát. Preto musí byť využitá databáza nerelačná. Nerelačné databázy nevyžadujú dodržanie schémy. Zmena v štruktúre vkladaných dát nespôsobuje problémy a je určená pre veľké datasety. Do databáze sa budú ukladať dokumenty formátu JSON. V dokumente musí byť obsiahnutý identifikátor na originálny príspevok, geodáta, text tweetu, čas vytvorenia príspevku, informácie o užívateľovi ako jeho nickname či počet sledovateľov, v akom jazyku bol tweet napísaný a výstup dát zo systému na spracovanie textu. Najpoužívanejšie riešenia pre správu nerelačných databáz sú riešené v podkapitolách 6.1 a 6.2, samotný výber v kapitole 8.3.

1.3 Naprogramovať systém pre spracovanie textu

Kroky predspracovania prijatých dát z Twitter API musia byť vykonané pre uložením do databáze. Predspracovanie musí obsahovať vytvorenie dokumentu v databázovej štruktúre. Tzn. dáta, ktoré možno odvodiť, budú odvodené a ostatné získané využitím modulov

dostupných pre jazyk Python. Technikám spracovania sa venuje celá kapitola 4. Spracovanie textu tweetu zahŕňa tokenizáciu na menšie celky a extrakcia hashtagov a mentionov, zistenie jazyka a užívateľovu lokáciu.

1.4 Vyhodnotiť zistenia štatisticky v rôznych kontextoch

Nerelačné databázy majú podporu u mnohých, zdarma dostupných, vizualizačných nástrojoch. Je potrebné vybrať ten správny na základe chcených výstupov. Keďže výstupy majú štatistickú povahu, aj vizualizačný nástroj musí ponúkať možnosti, ako takéto dáta patrične zobraziť. V podkapitole 6.3 sú objasnené a znázornené možnosti vizualizačných nástrojov. Medzi nutné funkcionality patrí filtrovanie určitého obdobia, konkrétnej hodnoty, zistenie najčastejšie sa vyskytujúcich hodnôt a zgrupovanie hodnôt. Podkapitola 8.4 popisuje dôvody zvolenia správneho nástroja.

1.5 Reprezentovať finálne produkty

Podľa možností vizualizičného nástroja zobraziť produkty prehľadnou formou. Dostupné lokalizačné dáta môžu byť zobrazené rovnako dobre na mapách ich v grafoch. Postup získania geodát je popísaný v kapitole 7. Informácie o vytvorení príspevku budú v grafovej forme zobrazovať aktivitu užívateľov počas určitého obdobia. Mentiony a hashtagy bude reprezentovať témy o ktorých užívatelia píšu v spojení so sledovanou témou v danom čase. Konkrétne príklady vizualizácií sa nachádzajú v kapitole 9.

2 TWITTER

Twitter je aktuálne jednou z najväčších a najvplyvnejších sociálnych sietí. Vo štvrtom kvartáli 2018 mal Twitter viac ako 321 miliónov aktívnych užívateľov. Podľa aktuálnych informácií za prvý štvrťrok 2019 je to už 330 miliónov [4]. Denne aktívnych užívateľov je približne 130-140 miliónov [13]. Toto číslo sa dramaticky mení pri aktuálne prebiehajúcich udalostiach vo svete. K nim môžeme zaradiť športové zápasy, voľby do parlamentu, prírodné katastrofy ai. Každú sekundu pribudne približne 6 000 nových tweetov čo je denne viac ako 500 miliónov [2].

Užívatelia si aktualizujú svoj status prostredníctvom správ nazývaných tweety. Maximálna dĺžka tweetu je 160 znakov, z toho 20 znakov je vyhradených pre identifikáciu autora príspevku. Užívatelia statusom vyjadrujú svoj názor k určitej téme, zverejňujú čo práve robia, delia sa o svoje skúsenosti. Súčasťou tweetu môžu byť aj iné elementy ako text a to hashtagy, mentiony, obrázky a odkazy na iné stránky.

Hashtag označuje identifikátor témy začínajúci znakom #. Je to forma kľúčového slova, ktorá pomáha k logickému a obsahovému zoskupovaniu príspevkov - tweetov [3].

Mention je identifikátor užívateľa začínajúci znakom @. Používa sa na označenie iného užívateľa v tweete [16].

Špeciálnym prípadom tweetu je tzv. retweet. Ide o príspevok, na ktorý sa užívateľ odkazuje. Retweet obsahuje informáciu na originálny príspevok. Označuje sa písmenami „RT“ na začiatku tweetu [15].

Dáta obsiahnuté v tweete majú vysokú informačnú hodnotu. Typickým príkladom využitia získaných dát je ich predaj tretej strane. Veľké zastúpenie tu má cielená reklama [21].

Analýza názoru z užívateľského textu na sociálnych sieťach sa používala aj na šírenie hoaxov a predpovedanie reakcie verejnosti na politickú kampaň [18]. Tieto nájdené využitia donútilo sociálne siete filtrovať, resp. blokovať príspevky na určité témy.

3 TWITTER DATA STREAMING

Data streaming označuje pojem obrovského množstva dát, ktoré sú prenášané od zdroja k cieľu a sú spracované v reálnom čase. Na úlohy, ktoré súvisia so streamingom dát, sú potrebné nemalé hardvérové prostriedky.

Každá z úloh vyžaduje určitý čas pridelený procesorom. Ak spracovanie tweetu trvá príliš dlho alebo je otvorených viac spojení v krátkom čase, Twitter spojenie uzavrie. Ďalšiu session je možné naviazať až po uplynutí určitého času. Táto situácia môže nastať pri vypadávajúcom internetovom pripojení, pri procese debuggovania alebo pri nesprávnom prístupe pri obnovení spojenia, keď sa klient pripája na stream. Nové spojenie je možné naviazať a konzumovať nové dáta až po uplynutí 15 minút. Obmedzenie sa vzťahuje aj na prekročenie množstva prijatých dát za určitý čas.

Prístup k streamu je pri všetkých knižniciach, ktoré využívajú Twitter API, už riešený štandardom OAuth. OAuth nevyžaduje zadanie užívateľského mena a hesla ale prístupové kľúče. Prístupové kľúče sa pridelia po registrácií aplikácie na Twittri. Kompletný postup [12] na získanie kľúčov sa nachádza na ich stránke. Postup zahŕňa vytvorenie developerského konta a následné zaregistrovanie vyvýjanej aplikácie. Twitter vygeneruje prístupové kľúče a využíva ich ako identifikátor cieľa.

Existuje niekoľko knižníc určených na zachytávanie dát z Twittra pre jazyk Python. K najpoužívanejším patrí Tweepy a Twarc. Prvý z dvojice ponúka vyššiu flexibilitu, druhý v poradí naproti tomu má jednoduchšie API. Oboje však patria k široko využívaným a dlhodobo podporovaným riešeniam pre streaming tweetov. Tweepy dokáže tweety nielen zachytávať, ale aj vytvárať.

3.1 Tweepy

Tweepy je široko používaný a odporúčaný klient pre ťažbu tweetov. Inštancia modulu tweepy.Stream naviaže streamingovú session a presmerováva tweety z Twittra do inštancie tweepy.StreamListener. Zachytávanie tweetov implementuje override metódy on_status, ktorý parametricky získa objekt formátu JSON.

O očakávané a neočakávané chyby sa dá postarať cez override metód on_error a on_exception [23].

Interface ponúka aj niekoľko základných metód na vytiahnutie dát z response objektu. Nie je teda nutné poznať štruktúru tweetu. Stačí zavolať metódu napr. get_user pre vrátenie užívateľa. Tieto gettery sú implementované len pre kľúče, ktoré sa v tweete nachádzajú vždy. Nie je zatiaľ podporovaný prístup k dátam, ktoré sú definované ako nepovinné.

Okrem najviac využívanej metódy GET je dostupná aj metóda POST. Vďaka tomu sa dá cez inštanciu nastaviť napr. odber na konkrétnych užívateľov.

Zaujímavou funkcionalitou je prístup k histórii cez objekt Cursor. Vďaka nemu možno iterovať prichádzajúce pole filtrovaných tweetov. Filter na Cursor umožňuje sa zamerať, medzi inými, aj na určité časové obdobie, užívateľa alebo tému.

3.2 Twarc

Jednoduchšou alternatívou k Tweepy je knižnica Twarc. Tá si vystačí s vytvorením inštancie a zavolaním metódy filter, ktorá v cykle vracia nové dáta. Nie je nutné vytvárať novú Listener triedu extendujúcu StreamListener, či potrebný override metód, ako je to v prípade Tweepy. Zaujímavou funkcionalitou je reporting priamo do html a geojson súborov. Na druhú stranu sa však horšie reaguje na error responsy API.

3.3 Štruktúra tweetu

Tweet obsahuje povinné aj nepovinné polia. Užívateľ často nemá zverejnenú svoju aktuálnu polohu alebo nepoužil v texte tweetu žiadny dodatočný element ako hashtag, či obrázok. Chýbajúce hodnoty tým menia štruktúru tweetu. Štruktúra tweetu je závislá na užívateľových nastaveniach profilu a či využíva dodatočných funkcie Twittra. Geolokačné dáta sú dostupné až po získaní povolenia od užívateľa v jeho profile. Ak túto funkcie užívateľ zapnutú nemá, ani kľúč geolokácie sa v tweete nenachádza.

Štruktúra tweetu je teda nerovnomerná. Veľká časť tweetu je nevyužiteľná a je nevyhnutná konverzia do užitočnejšej formy.

S textom tweetu sa pracuje najčastešejšie a je pravdepodobne z hľadiska analýzy najzaujímavejší. Často sa využíva k zisťovaniu názoru označovaného aj ako analýza sentimentu.

Retweet patrí medzi nadbytočnú hodnotu. Počet užívateľov, ktorí reagovali prostredníctvom retweetu už ako nadbytočnú označiť nemožno. Retweet je podobná funkcia ako zdieľanie príspevku na iných sociálnych sieťach. Môže znamenať stotožnenie sa s názorom iného užívateľa obsiahnutom v tweete alebo považovanie informácie za dôležitú. Retweetom sa k príspevku dostanú ďalší používatelia, ktorí ich ďalej zdieľajú. Množstvo zdieľaní zvyšuje aktuálnu a budúcu prestíž príspevku, témy, užívateľa a tým možno jeho tweet klasifikovať ako hodnotnejší. Počet sledovateľov užívateľa plní podobnú úlohu [18].

Twitter umožňuje užívateľom vo svojom profile pridať krátky opis označovaný ako description. Obsahuje dodatočné informácie, ktoré sa nemusia dať iným spôsobom vyčítať z profilu užívateľa. Môžu to by užívateľove záujmy, prax v pracovnom obore, hobby, pridruženosť k iným užívateľom alebo k danej téme. Aj tieto informácie môžu byť využité pri zisťovaní objektivity. Objektivita tweetu sa využíva ako parameter, ktorý vyjadruje na koľko je text tweetu subjektívny. Určité slová a slovné spojenia zvyšujú alebo znižujú mieru subjektivity.
Podrobné informácie o zložení tweetu sú uvedené na stránkach Twittru [17].

4 PREPROCESSING

Náročnosť spracovania užívateľského textu ovplyvňujú pravidlá konkrétneho jazyka, gramatika, vetná stavba atď. Často sú využíté postupy, ktoré umožňujú algoritmicky previezť dáta do vhodnejšej podoby. Záleží od povahy aplikácie, aké metódy je vhodné použiť. Postup, ktorý by všeobecne zaručil potrebnú kvalitu výstupu aktuálne nie je dostupný. Odporúčania sa rôznia takmer vo všetkých krokoch spracovanie užívateľského textu, následnosti krokov aj použitých technológiách. Kvôli obmedzeným zdrojom je kompromis medzi kvalitou a kvantitou nevyhnutný. Dáta zo streamu tečú vo veľkom množstve a vysokej rýchlosti. Softvér teda musí byť dostatočne rýchly, aby hardvér stíhal vybavovať požiadavky v rozumnom čase a zároveň garantoval minimálnu kvalitu výstupu po zbehnutí preprocessingu.

O optimálnom spôsobe spracovania užívateľského textu získaného zo sociálnych sietí sa vedú rozsiahle debaty. Štandartný postup zahŕňa odstránenie textového šumu. Veta sa rozdeľuje na jednotlivé slová a skupiny slov. Tie sa prevedú do slovotvorného základu a použitý algoritmus im priradzuje, podľa použitého slovníka, významové atribúty. Typicky sa hľadajú výrazy, ktoré označujú mená miest, štátov a osôb. Experimentovaním s poradím krokov spracovania a použitými nástrojmi na spracovanie textu docielime výstupy rôznej kvality. Nasledovné techniky [22] sú len odporúčané a je potrebné zvážiť ich použitie pre potreby aplikácie.

4.1 Tokenizácia textu

Ide o úlohu rozdelenia textu na zoznam oddelených slov, tzv. tokeny. Každý token je potom analyzovaný samostatne alebo v skupinách tokenov. Je medzi absolútnym základ om pri spracovaním textu a patrí k štandartným metódam Natural Language Processing (NLP) [7].

Nasledovný tweet bude rozdelný na slová a slovné spojenia prostredníctvom modulu Rake-NLTK a TextBlob. Tokenizácia prebehne na inak nespracovaných dátach.
Originálny tweet: „#COVID19 Safety Tip: avoid or strictly limit time spent in Closed spaces, Crowded Places and Close Contacts situations. Less is best – don’t double or triple dip into the high-risk 3-Cs!“

Výstup: ['strictly limit time spent', 'covid19 safety tip', 'close contacts situations', 'triple dip', 'risk 3', 'crowded places', 'closed spaces', 'best –', '’', 'less', 'high', 'double', 'cs', 'avoid'].

Pre porovnanie s rovnakým vstupom použitá knižnica TextBlob.

Výstup: ['covid19 safety tip', 'closed', 'crowded places', 'close contacts', 'less', '– don ’ t double', 'triple dip', 'high-risk 3-cs']

4.2 Odstránenie špeciálnych znakov a šumu

Aplikovaním regulárnych výrazov sa odstránia nepotrebné znaky ako napr. interpunkčné znamienka. Považuje sa za základnú techniku pri spracovaní užívateľské textu [10].
Zameranie aplikácie na získanie sentimentu môže odstránenie týchto znakov značne ovplyvniť. Napíklad emotikony sú tvorené kombináciou interpunkčných znamienok. Algoritmy zisťujúce sentiment dokážu emotikony, čísla a interpunkčné znamienka zohľadniť a tým spresniť a dosiahnuť tak vyššiu kvalitu výstupu [19].

4.3 Nahradenie nepotrebných výrazov

Väčšina twittrových textov obsahuje URL, hashtagy a mená užívateľov. Ich prítomnosť neobsahuje žiadnu informáciu vhodnú pre zistenie názoru užívateľa a ich výskyt sa nahrádza špeciálnymi označeniami ako URL, USER alebo ho úplne vypustiť. Dochádza k anonymizácií. Originálnu hodnotu alebo odkaz na orignálny príspevok je však stále vhodné uložiť ako referenčnú hodnotu.

Na príklade ukážeme anonymizáciu URL adresy náhradným stringom použitím regulárneho výrazu '\s+https?$'.

Originálny tweet: „WHO has created a Technical Advisory Group on Behavioural Insights and Sciences for Health to broaden its existing work on behavioural science and to offer more effective health advice: https://bit.ly/3k2g42s“

Výstup: „WHO has created a Technical Advisory Group on Behavioural Insights and Sciences for Health to broaden its existing work on behavioural science and to offer more effective health advice: _URL_“

4.4 Nahradenie negácií a konverzia do slovotvorného základu

Využitie tohto prístupu nepatrí medzi široko používané. Je však veľmi užitočné. Ide o vyhľadávanie konkrétnych slov v užívateľskom texte, ktoré spôsobujú negáciu v slovných spojeniach. Tieto spojenia sú nahradené ich antonymami. Negácie sú z textu tým vylúčené. Z užívateľovho výroku „nie dobrý“ sa po aplikovaní nahradenia negácie stane „zlý“. Rovnaký postup sa aplikuje na synonymá. Algoritmus vie slová s podobným alebo rovnakým významom optimálne konvertovať do slovotvorného základu. Pomáha to k jednoduchšej klasifikácií a zoskupovaniu vetných členov. Slová s podobným významom sa zlúčia pod jeden výraz. Ide o silný nástroj pri zisťovaní frekvenčnej distribúcie.

4.5 Nahradenie čísel

Na rozdiel od nahradenia antonymami je nahradenie čísel bežná činnosť pri spracovaní užívateľského textu. Čísla sami o sebe nemajú informačnú hodnotu pokiaľ nie sú vedené v určitom kontexte. Tento krok musí nasledovať až po nahradení skratiek či slangu, keďže práve čísla môžu byť ich súčasťou. Dnešné algoritmy sú pomerne dobre vybavené. Dokážu zistiť, kedy je číslo pri spracovaní dôležité a kedy nie.

5 NATURAL LANGUAGE TOOLKIT

Skrátene NLTK, je jednou z najpoužívanejších knižníc pre prácu s textom pre jazyk Python. Je opensource, zdarma a dostupná pre Windows, Mac OS X a Linux. Na obľúbenosti jej pridáva rozsiahla dokumentácia API. Interface s prístupom k funkciám na spracovanie textu ako tokenizácia, stemming či lemmatizing je len malá vzorka toho, čo všetko NLTK dokáže. Medzi najzaujímavejšie a často využívané funkcie patrí POS Tagging.

5.1 Part-of-speech (POS) Tagging

Anglický jazyk, tak ako mnohé ďalšie, majú špecifickú vetnú skladbu. Podstatné meno, prídavné meno či prísudok majú svoje dané miesto vo vete, majú svoj tvar, podľa určitých pravidiel je ich možné identifikovať a jednoznačne určiť ich vzťah vo vete k iným vetným členom. Proces identifikácie je však občas aj pre ľudí zložitý.

Vo vetách často chýbajú niektoré vetné členy, podstatné mená majú rôzne významy, v anglickom jazyku dokonca desiatky významov, bez bližšieho poznania kontextu nie je možné jednoznačne zistiť, či skúmané slovo vystupuje ako podstatné meno alebo sloveso, nehovoriac o preklepoch v texte spôsobené ľudským faktorom, slangových slovách, slovné spojenia atď. Pre tieto a ďalšie časté problém sa snaží NLTK nájsť riešenie opravovaním textu pomocou slovníkov, machine-learning procesmi atď. Poznať, ktoré slovo vo vete vysupuje v akom význame sa totiž ukázalo ako veľmi výhodné.

POS tagging z užívateľského textu vytvára dvojice (slovo, tag). Tag pridáva dodatočnú informáciu o slove - slovný druh. Ukázalo sa, že práve táto informácia sa radí ako kľúčová pri zisťovaní sentimentu. Najväčšia hodnota je pripisovaná prídavným menám, ktoré s najväčšou váhou opisujú daný predmet, ten môže byť umocnený aj inými prostriedkami ako sú interpunkčné znamienka (napr. výkričníkom).

Na ukážke POS-Taggingu tweetu pomocou knižnice TextBlob vidieť priradenie tagov ku každému slovu vo vete. TexbBlob používa rozšírený zoznam skratiek oproti štandardom a tým presnejšie identifikuje vetnú skladbu. JJ napr. označuje prídavné meno, z angl. adjective a VB označuje sloveso, z angl. verb. Zaujímavosťou je identifikácia „https“ ako NN, teda podstatného mena.

Originálny tweet: „#COVID19 Safety Tip: avoid or strictly limit time spent in Closed spaces, Crowded Places and Close Contacts situations. Less is best – don’t double or triple dip into the high-risk 3-Cs!“

Výstup: [('WHO', 'WP'), ('has', 'VBZ'), ('created', 'VBN'), ('a', 'DT'), ('Technical', 'NNP'), ('Advisory', 'NNP'), ('Group', 'NNP'), ('on', 'IN'), ('Behavioural', 'NNP'), ('Insights', 'NNPS'), ('and', 'CC'), ('Sciences', 'NNPS'), ('for', 'IN'), ('Health', 'NNP'), ('to', 'TO'), ('broaden', 'VB'), ('its', 'PRP$'), ('existing', 'VBG'), ('work', 'NN'), ('on', 'IN'), ('behavioural', 'JJ'), ('science', 'NN'), ('and', 'CC'), ('to', 'TO'), ('offer', 'VB'), ('more', 'RBR'), ('effective', 'JJ'), ('health', 'NN'), ('advice', 'NN'), ('https', 'NN'), ('//bit.ly/3k2g42s', 'NN')]

5.2 Textblob a Rake Nltk

Obe knižnice spracúvajú text podobným spôsobom. Využívajú k analýze sentimentu slovníky určené priamo k tejto činnosti. Znižuje sa tým pravdepobonosť, že prvky textu budú nesprávne klasifikované. Tešia sa veľkej podpore zo strany komunity. Slovník dokáže zatriediť ako veľmi je užívateľský text pozitívny alebo negatívny. Vytvorenie takéhoto slovníka je náročné a to ako časovo tak aj finančne. Slovník, okrem toho, že pokrýva čo najviac slov v danom jazyku, získava aj subjektívnu a objektívnu informáciu od užívateľov v číselnom vyjadrení miery pozitivity alebo negativity (spomínaný interval od -1 po 1).

Tento prístup volia aj alternatívne knižnice, ktoré sa zaoberajú analýzou sentimentu. Slovník musí byť neustále aktualizovaný - pribúdajú nové pomenovania, nové skratky, nové slangy, emotikony ai.

Algoritmus skúma vzťahy medzi jednotlivými slovami vo vete. Hľadá medzi nimi súvisloti a berie do úvahy aj kontext. Všetky tieto aspekty majú vplyv na konečné ohodnotenie. Komplikovaná manipulácia s textom zvyšuje celkovú korektnosť výstupných údajov.

Funkcionalitu knižníc pokrýva rozsiahla avšak zrozumiteľná dokumentácia s príkladmi použitia ako k Textblob [14] tak aj k Rake Nltk [20].

Pri reálnych testoch dosahoval Rake Nltk lepšie výsledky ako Textblob. Konkrétne v POS Tagging vedel Rake Nltk lepšie identifikovať slovné spojenia, tzv. noun chunkoch. Výrazný rozdiel v ich kvalite pri zvolenej téme bol dôvodom pre zvolenie práve tejto knižnice.

Originálny tweet: „#COVID19 Safety Tip: avoid or strictly limit time spent in Closed spaces, Crowded Places and Close Contacts situations. Less is best – don’t double or triple dip into the high-risk 3-Cs!“

Meranie ukázalo veľký rozdiel v čase spracovania už na takejto malej vzorke dát. Rake-NLTK trvalo spracovanie POS-Taggingu 0.0089s a TextBlob až 0.1312s.

5.3 Analýza názoru

Ide o proces spracovania textu, kedy sa vytvorený systém snaží identifikovať názor užívateľa na danú tému. Zistenie názoru znamená ohodnotenie každého slova a skupiny slov textu. Zistenie názoru z písanej formy sa využíva rovnako často ako z tej nepísanej, avšak efektívnejšie. Využíva sa pre zistenie reálnej spätnej väzby na konkrétny produkt alebo zlepšenie kvality určitej služby. Klasifikáciu ovplyvňujú faktory ako miera subjektivity, teda v akej miere je užívateľský text subjektívny alebo objektívny. Názor sa zvyčajne zatrieďuje ako pozitívny, negatívny alebo neutrálny. Ak je väčšina slov a slovných spojení ohodnotených ako pozitívna, tak aj samotný text je ohodnotený ako pozitívny.

Algoritmus hodnotiaci názor využíva tzv. slovníky. Aby mohol byť tento algoritmus aplikovaný na text, je potrebné ho najprv predspracovať. Správnym predspracovaním je možné docieliť efektívnejšiu klasifikáciu. Predspracovaním sa vyčistia dáta, ktoré nemajú informačnú hodnotu. V priemere sa nachádza v texte tweetu takmer 40% týchto nežiadúcich informácií a ich neodstránením dochádza k skresleným záverom. Príkladmi týchto informácií sú užívateľove preklepy, slangové výrazy, skratky a čísla. Niektoré slovníky dokážu tieto slová rekonštruovať do využitelnej podoby. Zvyčajne sa kombinujú viaceré prístupy pri predspracovaní ako nahradzovanie podobných slov synonymami, nahradenie záporov s antonymami, spájanie slov či ich úplné odstránenie.

Normalizácia prebieha odstránením nadbytočných znakov a slov, ktoré samé o sebe nemajú informačnú hodnotu a nedajú sa klasifikovať ako pozitívne alebo negatívne. Normalizovaný text je rozdelený na jednotlivé slová a slovné spojenia vzniknuté ich vzájomnou kombináciou. Slová a slovné spojenia textu sú porovnávané so slovníkom. Tie majú vslovníku priradenú svoju polaritu.

Polarita charakterizuje slovo, ku ktorému je priradená, ako negatívna s hodnotou -1 a polarita s hodnotou 1 ako pozitívna. Zistenie názoru závisí aj od zvoleného slovníka. Tie sa často líšia v prístupe hodnotenia polarity. Rôzne slovníky kladú rôzny dôraz na rôzne slovné spojenia.

Aktuálnosť slovníka taktiež patrí k významným parametrom. Slovníky sú vstupné dáta tvorené ľuďmi. Polaritu, ktorú slovám priradzujú, sa v priebehu času mení. Použité výrazy nadobúdajú nové významy, stávajú sa viac či menej akceptované a záleží aj od kontextu, v ktorom sú použité.

6 NERELAČNÁ DATABÁZA

Alebo tiež NOSQL, ako alternatíva oproti relačnej (SQL), je vhodná voľba pre ukladanie napr. dokumentov štruktúry JSON s možnosťou pridania ďalších kľúčov a hodnôt narozdiel od SQL prístupu vyžadujúci vytvorenie konkrétnej definície schémy. NOSQL sa vyznačuje svojou spoľahlivosťou a zameraním na výkon pre rýchly prístup k dátam. Zvyčajne sú NOSQL databázy dostupné open-source a využívajú sa pre ukladavánie logov, záloh a pri realtime vyhľadávaní.

Medzi najpoužívanejších a najobľúbenejších predstaviteľov NOSQL databázy patrí MongoDB a Elasticsearch.

6.1 MongoDB

Veľmi elegantné open-source riešenie ukladania dát do nerelačnej databázy ponúka MongoDB. Ukladá dáta do dokumentov typu JSON. Podporou vnorených objektov zabezpečuje flexibilnosť a dynamickosť navrhnutej schémy. Jazyk query, ktorý je tiež len JSON objektom, dokáže polia radiť a vyhľadávať v nich, pritom nezáleží, ako hlboko je pole vnorené. To platí pre všetky typy dát, aj pre tie špeciálne ako je napr. geolokácia. Engine MongoDB je dostatočne komplexný, aby zabezpečil plynulý chod tranzakcií a reagoval na chyby, ktoré môžu počas tranzackie nastať. Poteší možnosť vytvorenia globalne distribuovaných clusterov s nízkou latenciou. Clusterom je možné prideliť užívateľov s rôznym typom prístupov alebo vytvoriť priamo whitelist.

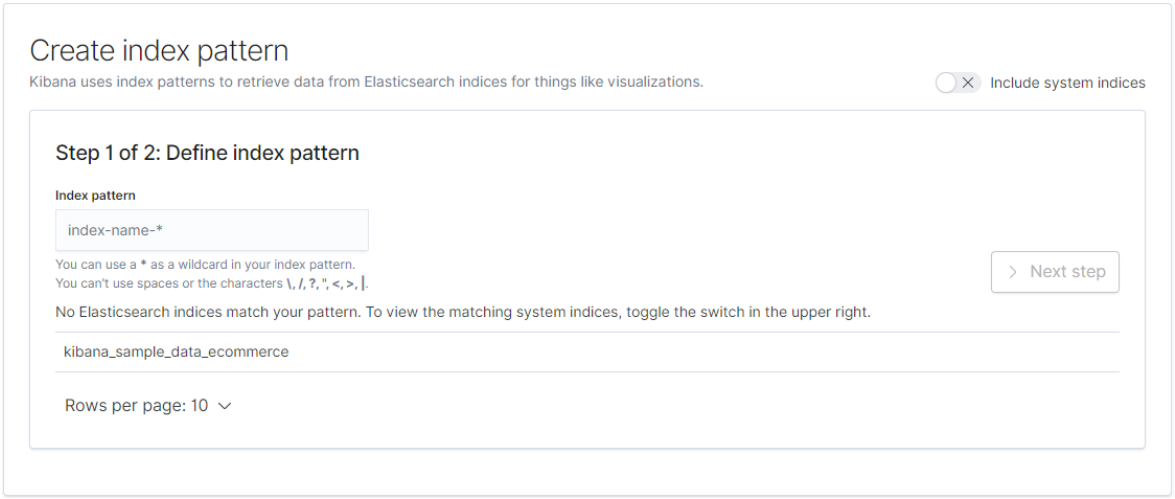
6.2 Elasticsearch

V jednoduchosti sa dá Elasticsearch popísať ako REST aplikácia, ktorá funguje ako základ pre zobrazenie dát v Kibane využívajúca middleware Logstash na predspracovanie dát a bezproblémovú komunikáciu s Kibanou. Beats je novou funkcionalitou, ktorá bola pridaná na žiadosť užívateľov. Pridáva možnosť vkladať a pracovať s celými súbormi. Elasticsearch zámerne skrýva pred užívateľmi spôsob akým pracuje. Všetky operácie vykonáva sám bez náročnej konfigurácie a riadi sa svojim mottom „it-just-works“.

S uloženými dátami chceme ďalej pracovať. Ideálne čo najjednochšie previazdať databázu s vizualizačným enginom. Elasticsearch je ideálnym riešením pre perzistentné ukladanie a vhodnejšie ako MongoDB.

6.3 Vizualizačný nástroj

Moderné riešenia pre zobrazovanie dá pre štatistické účely je k dispozícií niekoľko. Z radu vytŕčajú najviac Kibana a Grafana. Obe aplikácie poskytujú jednoduchý interface a celý rad nástrojov na analýzu dát a ich zobrazenie v rôznych tabuľkách, mapách, grafoch a to všetko v reálnom čase. Rovnako ako Elasticsearch, nepotrebuje takmer žiadnu konfiguráciu. Po spustení sa dá okamžite pracovať na užívateľom vytvorených prostrediach, tzv. dashboardoch. Dashborad reprezentuje zoskupenie výsledných vizualizácií. Kibana získáva prístup k dátam Elastisearch pod tzv. indexom.



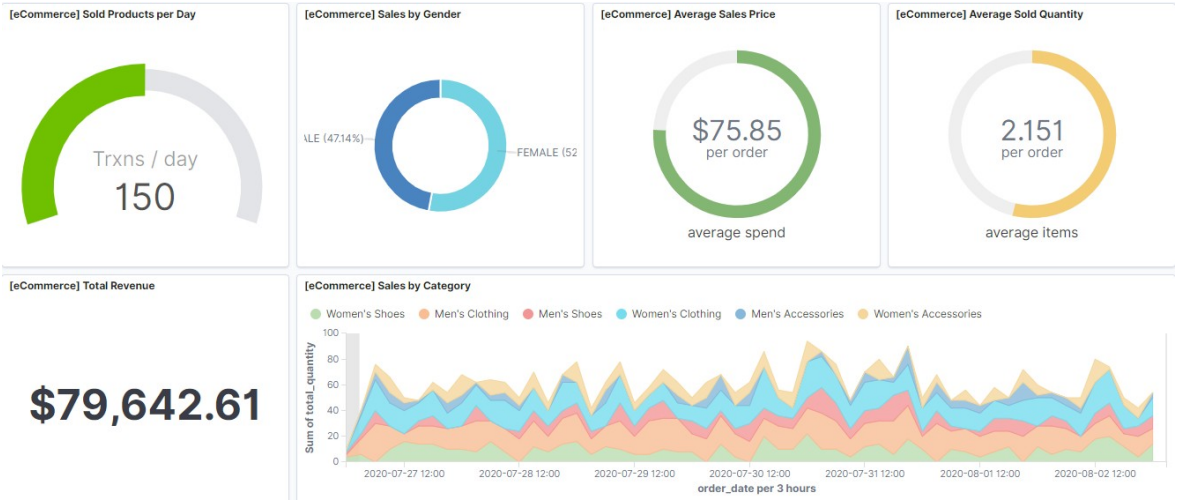
Obr. 1: Registrácia indexu

Obe platformy vedia pracovať s Elasticsearch. Grafana bola vytvorená ako fork Kibany pre prácu aj s inými backend riešeniami a time-series databázami ako defaultný Graphite či InfluxDB. Kibana ich taktiež podporuje v kooperácií s Logstashom. Výhoda Kibany je, že defaultne spolupracuje s Elasticsearch. Stačí len konfigurácia portu, kde beží Elasticsearch.

Výhodou Grafany je ďalšia vrstva bezpečnosti. Už vo svojej základnej verzii je potrebné zadať defaultné prihlasovacie údaje. Po ich zadaní nastane presmerovanie na novú masku s nastavením nového hesla.

Pre bezproblémový prísun dát je odporúčané využiť Mapping API [1]. Jednoducho sa dá vytvoriť štruktúra dát, ktoré budú z Elasticsearch odosielané Kibane. Elasticsearch má totiž problémy s null hodnotou a je vhodné túto hodnotu konvertovať na prázdny reťazec.

Silu Kibany možno vidieť na sample dátach. V každej kategórii je uvedený príklad, ako konkrétnu vizualizáciu vytvoriť a ako bude vyzerať. K dispozícii je už aj pripravený dashboard. V Grafane je pripravený prehľadný tutorial. K jeho sprevádzkovaniu treba použiť clone dát, ktorý je dostupný na Github repository.



Obr. 2: Kibana Dashboard - sample data

V oboch nástrojoch nechýba možnosť importovať a exportovať dáta z a do súboru. Pri importe podporuje Kibana CSV, TSV alebo JSON formát s obmedzenou maximálnou kapacitou 100 MB. V Grafane je to len JSON. Export dát sa nachádza pod panelom Reporting. Kibana vie dáta vyexportovať vo formáte CSV a Grafana v JSON.

7 GEOLOKÁCIA

Kibana podporuje zobrazovanie zemepisnej výšky a šírky, početnosť dát podľa regiónov a štátov vo formáte ISO 3166-1 alpha-2 a alpha-3, či dokonca podľa ich názvov. Geodáta nie sú štandartne dostupné z tweetov a je potrebné ich získať inou cestou. Koordinácie dostupné pri niektorých z nich označujú aktuálnu polohu autora, ktorý príspevok uverejňuje. Vzhľadom na zvolenú sledované tému je vhodným riešením získať tieto dáta priamo z textu tweetu. Tým síce nezistíme lokáciu autora, ale k akej oblasti je text relatívny čo je rovnako cennou informáciou.

Výber medzi knižnicami, ktoré dokážu extrahovať geolokáciou z užívateľského textu, je pomerne malý. Popularite sa tešia Mordecai a NER.

7.1 Mordecai

Projekt spaCy [9], využívajúci end-to-end open source platformu TensorFlow pre machine learning, ktorý sa zameriava na full textový geoparsing a event geocoding [8]. Extrahuje názvy miest z neštruktúrovaného textu a vráti štruktúrované geografické informácie. Mordecai k svojej práci vyžaduje nainštalovaný Elasticsearch a na ňom bežiaci Geonames gazetteer index na platforme Docker a spaCy NLP model. Využíva TensorFlow, neurálne siete Keras, ktorého API je napísané v jazyku Python a je trénovaný na dátach ďalšieho projektu spaCy Prodigy. Všetky tieto technológie sú potrebné k správnej identifikácií krajiny a jej názvu, pokiaľ je napísané v cudzom jazyku.

Mordecai k svojej práci potrebuje bežať na špeciálnej verzii Elasticsearch v kontajneri Dockeru. Ten funguje ako zovšeobecnenie operačného systému. Mordecai príjme text ako vstup a vráti pole objektov vlastného geolokačného typu. Pole obsahuje toľko objektov, koľko Mordecai nájde v texte geolokácií. Prvok poľa zahŕňa koeficient pravdepodobnosti, ako veľmi je text relatívny s nájdenou lokáciou v ňom. Jednoduchým algorimom je vybraný jediný objekt s najvyšším koeficientom. Celý tento proces je náročný najmä na operačnú pamäť a nevhodný pre real-time processing. Bol teda vytvorený middleware, ktorý získa tweety uložené v Elasticsearch a spracované Mordecaiom vráti späť.

Mordecai dosahuje výborné výsledky čo sa týka kvality. Pri ostrej prevádzke nie je ale dostatočne rýchly. Twitter Data Stream zachytáva každú minútu stovky nových tweetov.

Predspracovanie textu zaberie určitý čas a ani asynchrónne vykonávanie operácií na dátach a implementácia vlákien nedokázali zabezpečiť minimálnu stabilitu v kombinácií s Mordecai. Nie je teda vhodný na tento účel a skôr sa hodí na batch datasety.

7.2 DeepPavlov

V asociácií s machine learning je NER – Named Entity Recognition - technikou vyberania informácií pre identifikáciu a klasifikáciu prvkov v texte. NER využívajú firmy ako Google v jeho emailovej službe pri detekovaní emailových adries, kontaktov, letov atď. Gmail potom ponúka návrhy napr. pre pridanie záznamov do kalendára. Ide o veľmi silný a rýchly nástroj pre taggovanie vetných členov v 104 podporovaných jazykoch. Medzi nimi aj český a slovenský jazyk.

DeepPavlov NER model podporuje okrem vyhľadávania tagu LOC (lokácií) aj ďalších 18 tagov ako ORG (oragnizácie) či PERSON (osobnosti) [11]. Celkovo ich je teda 19.

Z modelov je na výber predtrénovaný alebo vlastný. Predtrénovaných modelov je niekoľko a plne postačujúce na bežné úlohy. Na trénovanie alebo spustenie modelov môže byť použitý ako procesor [5] tak aj grafický čip. GPU varianta DeepPavlov je podporovaná len na grafických kartách NVIDIA [6]. Obe varianty vyžadujú buď inštaláciu DeepPavlov alebo jeho image bežiaci cez Docker.

II. PRAKTICKÁ ČÁST

8 SKRIPT

Python ponúka niekoľko modulov pre zachytávanie dát z Twitter API. Medzi široko používané patria Tweepy a Twarc. Pre svoju jednoduchú implementáciu a kvalitnú dokumentáciu sa stali obľúbenými.

Dáta zachytené klientom budú v skripte predspracované. Predpracovanie bude zahŕňať očistenie od nepotrebných hodnôt použitým regulárnych výrazov. S očistenými dátami sa dá jednoduchšie pracovať v ďalších krokoch. V tých budú texty tweetov rozdelené na tokeny a tie znova pospájané do slovných spojení. Ďalšími krokmi budú zistenie jazyka použitím modulu langdetect a zistenie geolokačných dát a polaritu príspevku.

Takto pripravené dáta budú uložené do nerelačnej databáze. Backendové riešenia ako Elasticsearch a MongoDB sú ideálnymi prostredníkmi pre ukladanie dát a zároveň ich API pre poskytovanie ďalším službám.

8.1 Výber Twitter API klienta

Počas testovania knižníc a rôznych postupov sa vyskytli problémy s výkonom.

Spočiatku používaný stream klient Twarc spoľahlivo dokázal získavať nové dáta a jednoducho sa používal. Problém nastal pri vypadnutom internertovom spojení. Pri streamingu dát je potrebné zabezpečiť, aby takáto situácia nenastávala často a pokiaľ nastane, z takejto chyby sa spamätať. Počas výpadku spojenia nie sú dostupné žiadne ďalšie dáta. Twarc však neponúka žiadny interface na prácu s chybami a handling errorov sa stal nezmyselne komplikovaný. Prechod na Tweepy bol teda logický krok. Override metód on_exception a on_error poskytuje spôsob, ako sa s takýmito chybami vysporiadať.

Knižnica Tweepy vyžaduje overenie OAuth kľúče pre odchytávanie tweetov. Získanie kľúčov je záležistosť pár minút o niekoľkých krokoch. Na stránke Twittru je potrebné zaregistrovať si novú aplikáciu a vygenerované kľúče autentifikuje inštancia Tweepy. Ďalej je len potrebné zadať kľúčové slovo a odchytávanie jednotlivých tweetov môže začať. V tejto práci sú zvolené skupinou slov spojené s témou koronavírusu, a to konkrétne „coronavirus“, „covid19“, „corona“ a „sarscov2“.

8.2 Spracovanie užívateľského textu

Pred perzistentným uložením je potrebný preprocessing. Dáta nemajú jednotnú formu. Obsahujú rôzne kľúče a hodnoty nevhodné na štatistickú analýzu. Po drastickej redukcii z takmer 300 rôznych kľúčov objektu ostala len desiatka. Zahŕňajú najdôležitejšie informácie ako je identifikátor tweetu, čas, text tweetu, hashtagy a informácie o autorovi tweetu. Ostatné dáta sú odvodené z týchto hodnôt.

Retweety napriek tomu, že ich označujeme ako redundatné, nie sú odfiltrované. Stream dokáže zachytiť len približne 1% všetkých tweetov a je teda veľmi nízka pravdepodobnosť, že bude súčasne zachytený originálny tweet a zároveň retweet.

Knižnica TextBlob obsahuje funkcie na vrátenie hodnoty polarity a subjektivity po zadaní parametra, ktorým je práve užívateľský text. Hodnota polarity určuje jako veľmi je tweet negatívny či pozitívny. Tieto informácie sa získavajú z nenormalizovaného textu. Zisťovanie názoru užívateľov na konkrétnu tému má veľmi silné zastúpenie pri analýze dát zo sociálnych sietí. Pri zvolenej téme koronavírusu sa však nepodarilo zobraziť dáta v informačne hodnotnej forme.

TextBlob dokáže, cez jednoduchý interface, vrátiť slovné spojenia, ktoré uzná za vhodné. Rôznymi kombináciami sa dá docieliť rôznej kvality výsledných fráz. Preto sú uložené ich vylepšené verzie použitím lemmatizácie. Rake NLTK, ako alternatívna knižnica, lepšie dokázala extrahovať tieto frázy.

Kvôli nedostatočnému počtu tweetov obsahujúcich koordinácie užívateľov bol zvolený iný postup pre získanie geolokácie. Vzhľadom na tému koronavírusu je porovnateľne vhodné vedieť lokáciu získať z užívateľského textu. Spomedzi voľne dostupných nástrojov je knižnica Mordecai a DeepPavlov najlepšou voľbou, za ktorú sa draho platí výpočetným výkonom. Mordecai si vyčleňuje pre seba približine 3 GiB operačne pamäte. DeepPavlov dokonca 4,5 GiB. Kvôli nestabilite Mordecai pri vysokom zaťažení bol zvolený pre extrakciu lokácií DeepPavlov aj za cenu vyšších hardvérových požiadaviek.

8.3 Výber databázového riešenia

Zachytené tweety sú objekty formátu JSON. Tweet, kvôli svojej nerovnomernej povahe, nie je vhodné ukladať do relačnej databázy. Pre perzistnentné uloženie dát bola zvolená NOSQL databáza manažovaná Elasticsearch.

Nový dokument z predspracovaných dát je poslaný do nakonfigurovanej inštancie Elasticsearch.

Dáta sa do Elasticsearch ukladajú pod indexami, s ktorými vie ďalej pracovať Kibana. Podobne ako Elasticsearch, beží lokálne v prehliadači defaultne na porte 5601. V jej Index Management interfacu sa indexy uložené v Elasticsearch dajú mazať alebo upravovať. Elasticsearch automaticky rozoznáva niektoré typy dát aako sú dátumy, geolokácia či text a vie podľa nich priradiť príslušné možné operácie pre vizualizáciu, či sortovanie dát.

Použitý hardvér nebol výkonovo dostatočný, aby zvládal taký nápor dát a výpočtových operácií v dlhšom časovom horizonte. Skript sa teda presunul z lokálneho na externé serverové riešenie, kde beží na operačnom systéme Ubuntu vo verzii 18.04.

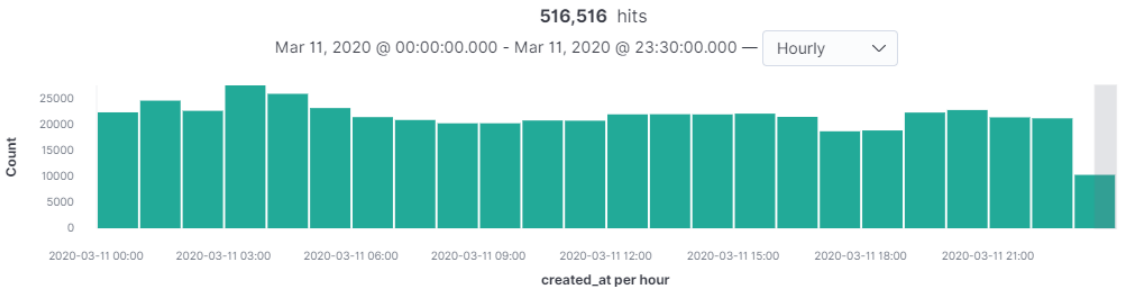
8.4 Výber vizualizačného nástroja

Grafana aj Kibana sú vo svojej triede vrchol user-friendly prostriedkov k zobrazovaniu big data. Oboje riešenia podporujú Linux, Mac, Windows aj Docker. Obe ponúkajú možnosti filtrácie dát, vizualizácie time-series grafmi aj podporu pre geodáta. Kibana má však navrch v spolupráci s Elasticsearch a teda aj v zjednodušenej konfigurácií. Grafana ako odnož Kibany bola vytvorené pre možnosť pracovať aj s inými backendami ako Elasticsearch pomocou rôznych pluginov. Pokiaľ nie je Kibana spojená s open-source riešeniami ako SearchGuard, jej dashboardy sú prístupné verejnosti.

Kibana, navzdory chýbajúcemu login modulu, je vhodnejšie riešenie pre výstupy, ktoré chceme dosiahnuť. Rozhodujúcim faktorom je jej spoľahlivosť pri spolupráci s Elasticsearch.

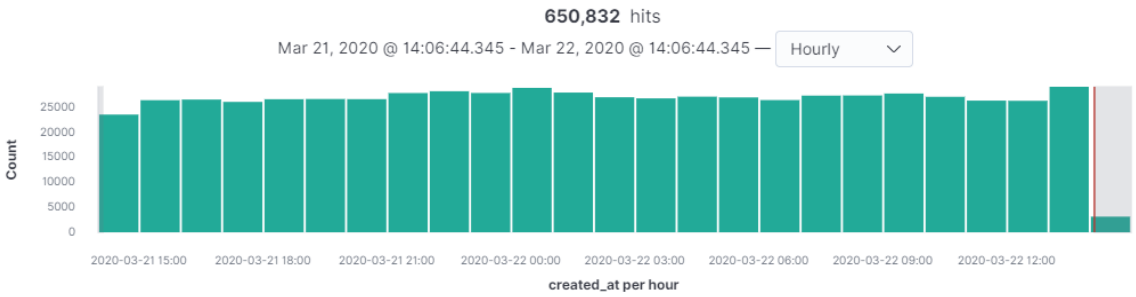
9 VÝSLEDKY

11.03.2020 bolo počas celého dňa odchytených približne 516 tisíc geoparsovaných príspevkov. Nasledovné grafy zobrazujú zvyšujúci trend v počte nových tweetov. Hodinový prírastok bol približne na rovnakej úrovni v oboch porovnávaných dňoch.



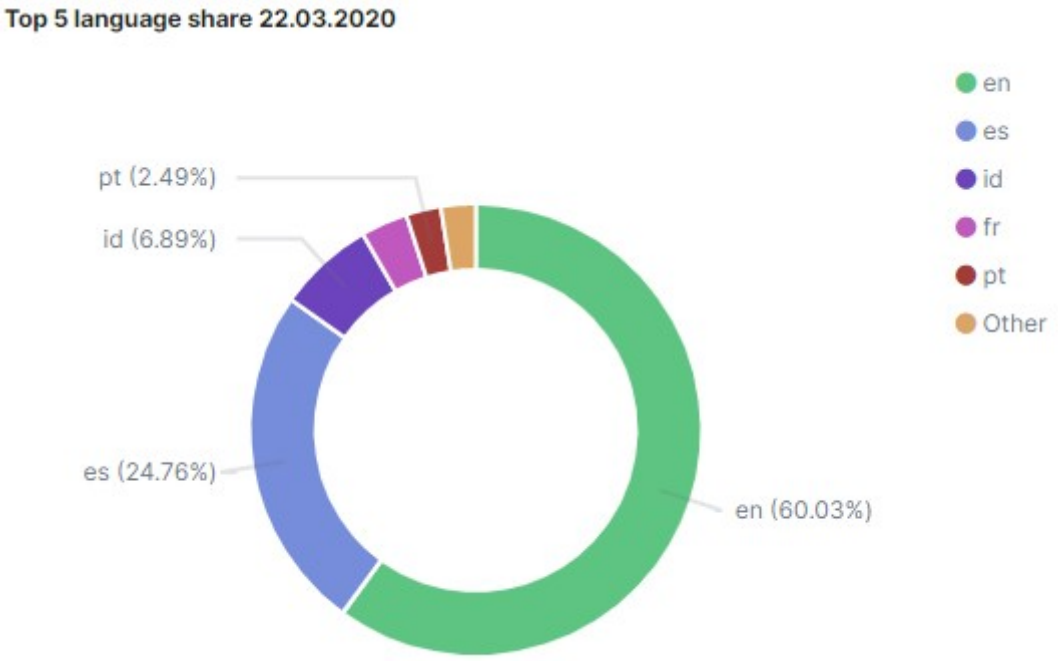
Obr. 3: Prírastok nových tweetov 11.03.2020

22.03.2020 počas sledovania trafiky, za pomerne krátke obdobie, bolo zistené zvýšenie aktivity užívateľov na Twittri pri téme coronavírusu z 516 tisíc na takmer 651 tisíc. Príčinou sú prvé potvrdené prípady napr. v Mongolsku, Indonézií a ďalších krajinách. Najviac ho však ovplyvnilo Španielsko a Taliansko. V oboch krajinách nastal prudký vývoj. Nové prípady, počas tohto obdobia, narastali v krátkom období po stovkách, tisícoch až desaťtisícoch.



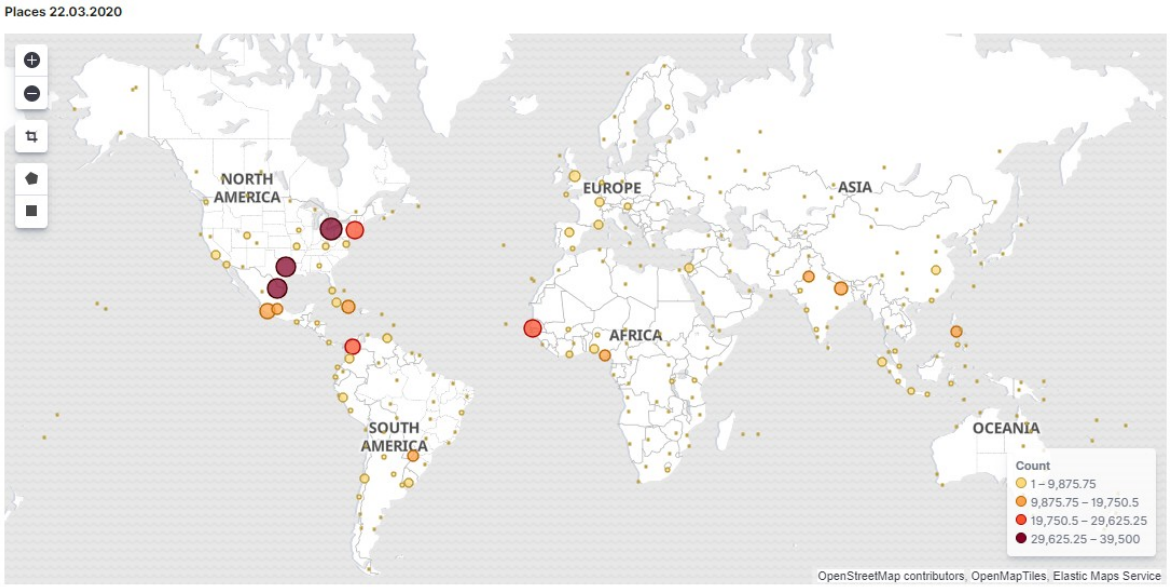
Obr. 4: Prírastok nových tweetov 22.03.2020

Španielčina sa vyzdvihla medzi najpoužívanejšie jazyky až na 2. miesto. Každý deň bol hlásený nárast nových prípadoch v Španielsku o cca 25%. Neprekvapivo je na 3. mieste India. V tento deň bolo vládou nariadené uzavretie všetkých miest, kde bol vírus zaznamenaný. V jazykoch vedia anglický jazyk, hlavne USA. V USA boliprijaté prvé opatrenia proti šíreniu nákazy. O týždeň neskôr sa krajina nachádzala na prvom mieste v rôznych rebríčkoch či už o celkovom počte nakazených, nových prípadov za 1 deň alebo úmŕtí počas 1 dňa.



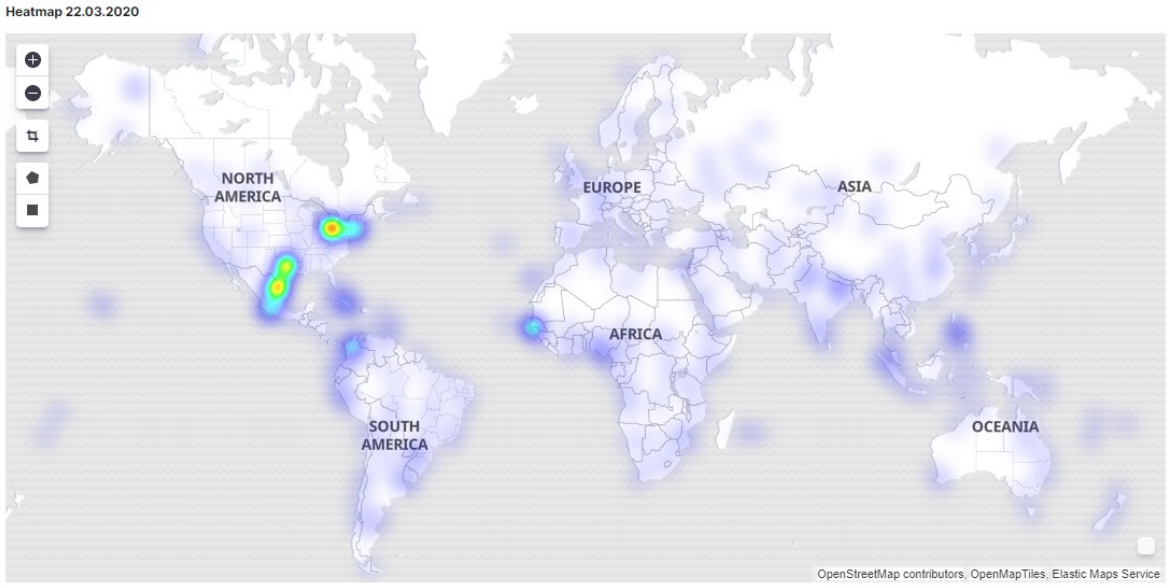
Obr. 5: Top 5 najpoužívanejších jazykov 22.03.2020

Na heatmapách sú zobrazené počty tweetov s miestami v nich spomenutými, tzn. mentiony, ktoré boli spojené s témou koronavírusu. Za všimnutie stojí podiel Číny k ostatným krajinám. V Číne v tomto období klesol denný prírastok nových prípadov na minimum, Naproti tomu v Európe sa blíži k svojmu vrcholu a v USA sa len rozbieha. V tweetoch je medzi mentionmi na prvom mieste. Na prvej heatmape zobrazuje legenda počet nových tweetov. Tmavšia farba indikuje vyššiu koncentráciu v danej krajine.



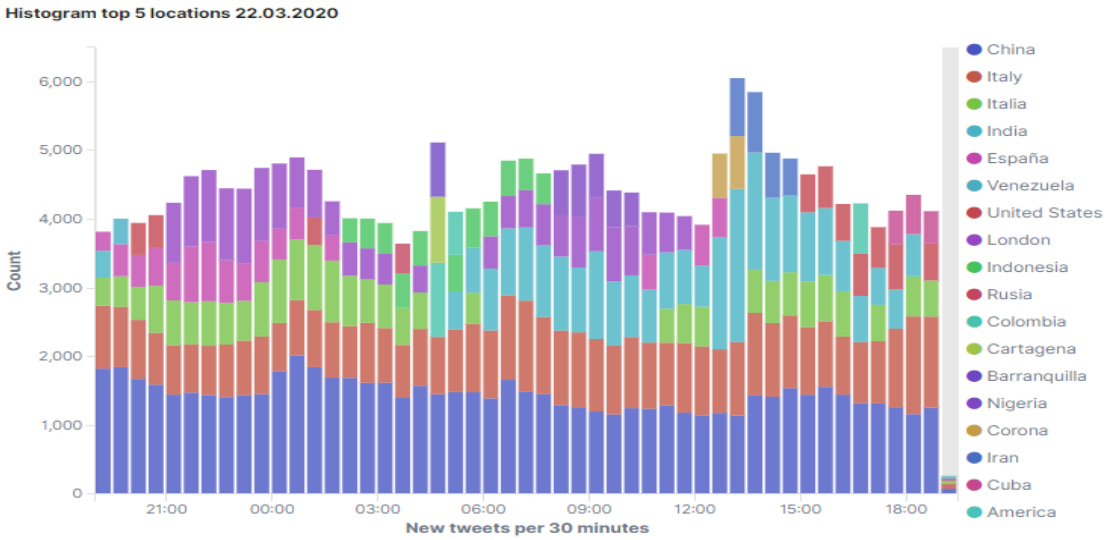
Obr. 6: Heatmapa počtu tweetov 22.03.2020

Druhá heatmapa je jednoduchším alternatívnym náhľadom rovnakých dát ako na prvej heatmape.



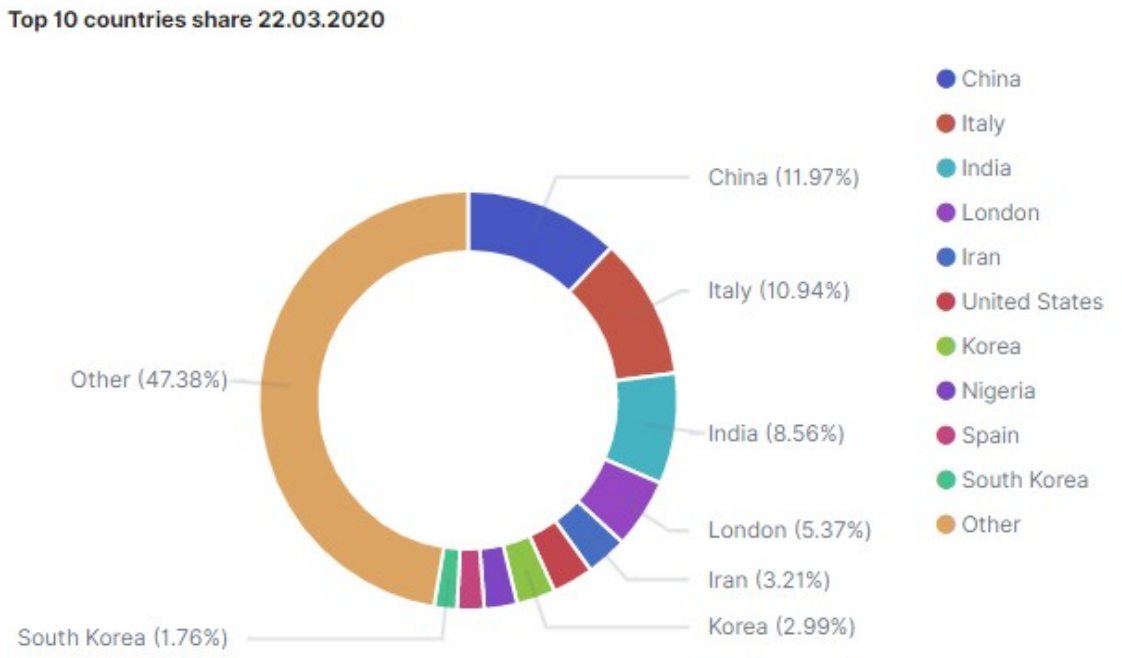
Obr. 7: Heatmapa podiel tweetov na región 22.03.2020

Histogram zobrazuje hodinovo 5 najčastejšie spomínaných krajinách počas 22.03.2020. Prvé miesto patrí Číne a vo veľmi testnom závese Taliansko. Kvôli zabezpečeniu rozumného času spracovania tweetu sa objavujú v dátach nezrovnalosti. Príkladom je zlá identifikácia slova „Corona“ vyhodnotená ako POS tag krajiny. Jedná sa však o tradičné chyby, ktoré sa vyskytujú s prácou s textom. Omylnosť modulu DeepPavlov je až prekvapivo nízka. Na čiastočné odstránenie takýchto chýb je potrebný ďalšia processing.



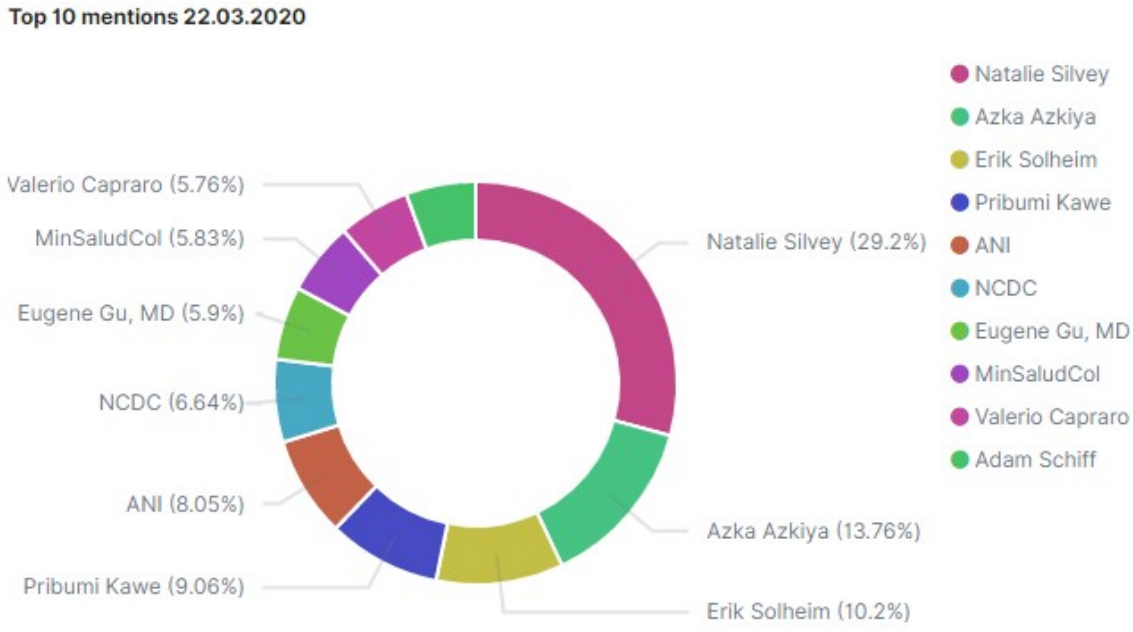
Obr. 8: Histogram top 5 najčastejších lokácií 22.03.2020

Na nasledujúcom grafe boli výrazne lepšie identifikované pri podiely až 10 krajín.

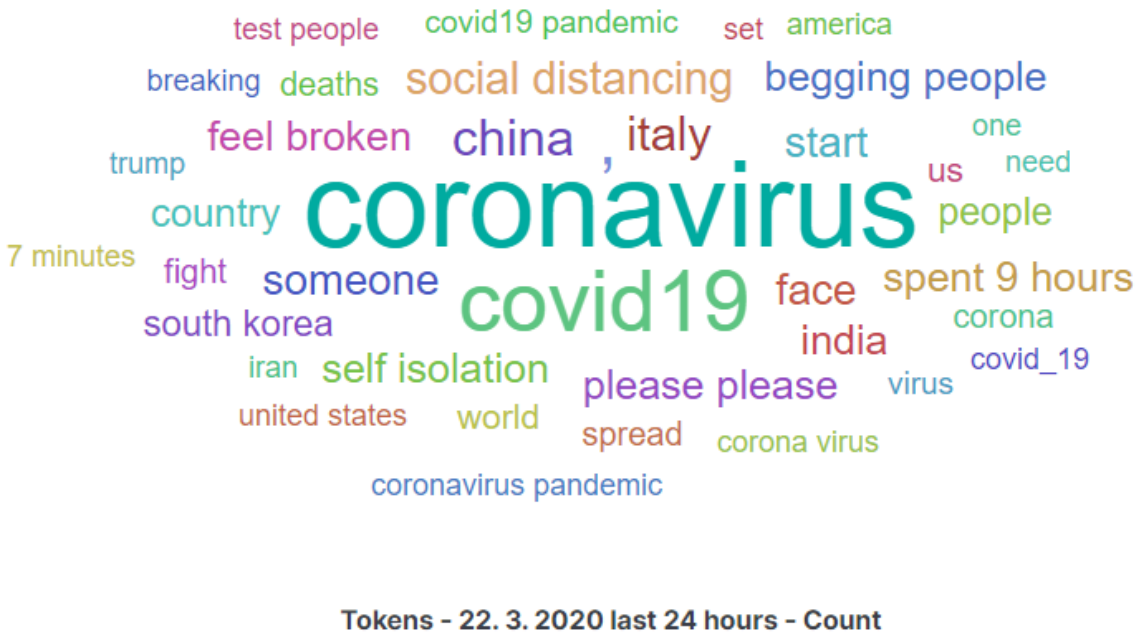


Obr. 9: Graf 10 najčastejšie označovaných krajín v tweetoch 22.03.2020

Vďaka možnostiam dodatočných úprav v Kibane, je možné do určitej miery nezrovnalosti v dátach filtrovať alebo odignorovať.



Obr. 10: Graf top 10 najčastejšie označovaných užívateľov 22.03.2020



Obr. 11: WordCloud najčastejšie používané slovné spojenia 22.03.2020

Extrahovaná lokácia v ČR za posledných 7 dní.

Obr. 12: Počet tweetov za posledných 7 dní 22.03.2020 v ČR

ZÁVĚR

Existujúce technické riešenia výrazne pomohli pri plnení cieľov tejto práce. Skript, napísaný v programovacom jazyku Python, využíva knižnicu Tweepy, ktorá po konfigurácií prístupových kľúčov, umožňuje pozorovanie aktivity užívateľov v reálnom čase. Dáta sú následne upravené rôznymi technikami pre spracovanie textu ako očistenie textu od nevýznamných slov, interpunkcie a vytvorenia logických slovných spojení.

Osvedčila sa napr. knižnica Rake Nltk, ktorá obsahuje funkcie na prácu s textom, napr. získanie slovných spojení a štatisticky ich zobraziť podľa ich početnosti. Keďže užívatelia vo veľkej miere nezverejňujú svoju geolokáciu, na získanie aspoň ich približnej polohy z príspevku bola využitá knižnica DeepPavlov. Vzhľadom na tému koronavírusu je extrakcia polohy z tweetu vhodným alternatívnym riešením. Cenou za tieto dáta sú však zvýšené softvérové a hardvérové požiadavky.

Spracované dáta sú zaslané pre perzistentné uloženie do Elasticsearch. Uložené dáta sú vizuálne prezentované v plugine Kibana spolupracujúca s Elasticsearch. Kibana využíva tieto dáta pre zobrazenie v množstve rôznych kontextoch. Najmä vizualizácia dát na mapách sa ukázala ako mimoriadne výpovedná. Koláčové grafy reprezentujú podiel zainteresovaných krajín, ktorými jazykmi užívatelia najčastejšie tweety píšu, aké krajiny sa spájajú s koronavírusom a aké slovné spojenia sa vyskytujú pri tejto téme.

Aplikovaním ďalších postupov na spracovanie užívateľského textu je možné docieliť kvalitnejších výstupov a výsledných vizualizáciách. Vizualizačný nástroj nie je bezpečnostne ošetrený. Nerelačná databáza manažovaná Elasticsearch je však dostatočne pružná. Vďaka tomu je možné dáta preniesť do iného vizualizačného nástroja, ktorý obsahuje aj bezpečnostné prvky ako napr. Grafana, alebo vytvoriť samostatný prihlasovací modul. Sledovanie iných hashtagov na sociálnej sieti spôsobí rozdiely v množstve prichádzajúcich dát. Optimalizovaním krokov predspracovania a zvýšením výpočtového výkonu možno docieliť stabilnejšieho chodu aplikácie. Príkladom je aj vytvorenie vlastného NER modulu na vlastných natrénovaných dátach.

SEZNAM POUŽITÉ LITERATURY

[1] BERMAN, Daniel. Elasticsearch Mapping: The Basics, Updates & Examples [online], 2020. Dostupné z: https://logz.io/blog/elasticsearch-mapping/

[2] BERNES-LEE, Tim. Twitter Usage Statistics [online], 2014. Dostupné z: https://www.internetlivestats.com/twitter-statistics/#

[3] Campbell, Anita. What is a Hashtag? And What Do You Do With Hashtags? [online], 2018. Dostupné z: https://smallbiztrends.com/2013/08/what-is-a-hashtag.html

[4] CLEMENT, J.. Twitter: number of monthly active users 2010-2019 [online], 2019. Dostupné z: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[5] deeppavlov. deeppavlov/base-cpu [online], 2020. Dostupné z: https://hub.docker.com/r/deeppavlov/base-cpu

[6] deeppavlov. deeppavlov/base-gpu [online], 2020. Dostupné z: https://hub.docker.com/r/deeppavlov/base-gpu

[7] GUPTA, Mohit. NLP | How tokenizing text, sentence, words works [online], 2019. Dostupné z: https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/

[8] HALTERMAN, Andy. Mordecai [online], 2020. Dostupné z: https://github.com/openeventdata/mordecai

[9] HALTERMAN, Andy. mordecai: Full text geoparsing using spaCy, Geonames and Keras [online], 2020. Dostupné z: https://spacy.io/universe/project/mordecai

[10] KOENIG, Rachel. NLP for Beginners: Cleaning & Preprocessing Text Data [online], 2019. Dostupné z: https://towardsdatascience.com/nlp-for-beginners-cleaning-preprocessing-text-data-ae8e306bef0f

[11] KONOVALOV, V.. 19 entities for 104 languages: A new era of NER with the DeepPavlov multilingual BERT [online], 2019. Dostupné z: https://towardsdatascience.com/19-entities-for-104-languages-a-new-era-of-ner-with-the-deeppavlov-multilingual-bert-1bfa6d413ea6

[12] LANE, Kin. Twitter API Authorization [online], 2019. Dostupné z: https://community.postman.com/t/twitter-api-authorization/9512

[13] LIN, Ying. 10 Twitter Statistics Every Marketer Should Know in 2019 [Infographic] [online], 2019. Dostupné z: https://www.oberlo.com/blog/twitter-statistics

[14] LORIA, Steven. TextBlob: Simplified Text Processing [online], 2020. Dostupné z: https://textblob.readthedocs.io/en/dev/index.html

[15] Merriam-Webster.com. Retweet [online], 2020. Dostupné z: https://www.merriam-webster.com/dictionary/retweet

[16] OSMAN, Maddy. Twitter Mentions: How to Find, Track & Get More [online], 2017. Dostupné z: https://sproutsocial.com/insights/twitter-mentions/

[17] PARMAR, Hiten. Example JSON response from Twitter streaming API [online], 2010. Dostupné z: https://gist.github.com/hrp/900964

[18] RA, S., PUJARI, J. SHREENIVAS BHAT, V., DIXIT, A.. Timeline Analysis of Twitter User, 2018, vol. 132, 157-166

[19] RATHAN, M., VISHWANATH, R., HULIPALLEDA, K. R., VENUGOPALB, L. M. Patnaik. Applied Soft Computing: Consumer insight mining: Aspect based Twitter opinion mining of mobile phone reviews, 2018, vol. 110, 298-310

[20] SHARMA, B Vishwas. rake-nltk [online], 2020. Dostupné z: https://github.com/csurfer/rake-nltk

[21] SHRAVAN KUMAR, B., VADLAMANI, R.. A survey of the applications of text mining in financial domain, Knowledge-Based Systems, 2016, vol. 114, 128-147

[22] SYMEONIDIS, Symeon. A Comparison of Pre-processing Techniquesfor Twitter Sentiment Analysis, 2017, vol. 10450, 396-398

[23] Getting started: API, 2020 [online], Dostupné z: http://docs.tweepy.org/en/latest/getting_started.html#api

SEZNAM OBRÁZKŮ

SEZNAM PŘÍLOH

PŘÍLOHA P 1: SKRIPT.

```
import tweepy
from tweepy import *
import re
import requests
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from elasticsearch import Elasticsearch
from dateutil import parser
import langdetect
from rake_nltk import Rake
from polyglot.text import Text
from geotext import GeoText
import preprocessor as p
from deeppavlov import configs, build_model

consumer_key = "MSjhLnSVrezk9A9hahH1KKYA1"
consumer_secret = "YPze8H6l35LtWa9leasNPud3muMqmv6qjbQIFdSgRj4OU7eVm5"
access_token = "839479598-5j5VsyhqkmAnSXYORvYiRe0yRKvRXSxE1dkYGN95"
access_token_secret = "QjhJ8IRacgJfBXsFj0LwCeVidql5rhnqXxmwgSyWA7Frk"

print("building ner model")
ner_model = build_model(configs.ner.ner_ontonotes_bert_mult, download=True)

index = "corona_v8"

class StdOutListener(StreamListener):
    def on_status(self, status):
        if hasattr(status, "retweeted_status"):
            try:
                process_tweet(status._json)
            except AttributeError as e:
                pass
    def on_error(self, status):
        return True
```

```
    def on_exception(self, exception):
        return True


def process_tweet(tweet):
    try:
        try:
            if tweet["retweeted_status"]["extended_tweet"]["full_text"]:
                text_tmp = tweet["retweeted_status"]["extended_tweet"]["full_text"]
        except:
            text_tmp = tweet["text"]
            pass
        cleantext_tmp = p.clean(text_tmp)
        if len(cleantext_tmp) > 100:
            places = GeoText(cleantext_tmp)
            cities_tmp = places.cities

            if cities_tmp:
                geodata, location = geoname_city(cities_tmp[0])
                        geodata = {item: geodata.get(item) for item in ["name", "country",
"population"]}

                if geodata is not False:
                    ner_loc, ner_org, ner_per = get_ner(text_tmp)
                    user = tweet["user"]
                    hashtags = get_nested(tweet["entities"], "hashtags")
                    if not isinstance(hashtags, list):
                        hashtags = [].append(hashtags)
                    document = {"created_at": parser.parse(tweet["created_at"]),
                                "location": location,
                                "text": text_tmp,
                                "geo": geodata,
                                "id_str": tweet["id_str"],
                                "mentions": get_nested(tweet["entities"], "user_mentions"),
```

```
                    "noun_chunks": get_noun_chunks(text_tmp),
                    "ner_per": ner_per,
                    "ner_org": ner_org,
                    "ner_loc": ner_loc,
                    "sentiment": 0,
                    "user_created_at": parser.parse(user["created_at"]),
                    "user_name": user["name"],
                    "user_description": user["description"],
                    "user_followers_count": user["followers_count"],
                    "user_verified": user["verified"],
                    "lng": detect_lng(text_tmp)}
            es.index(index=index, body=document, doc_type="_doc")
            print("new entry:", document["created_at"])
    except Exception as e:
        print("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
        print(e)
        print("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!")
        pass


def geoname_city(city_tmp):
    url_osm = "http://207.180.228.23/city?"
    geonode_params = {
        'name': city_tmp
    }

    result = requests.get(url=url_osm, params=geonode_params).json()
    if result[0]:
        try:
            location_tmp = result[0]
            location = {"lat": float(location_tmp["location"]["latitude"]),
                        "lon": float(location_tmp["location"]["longitude"])}
            return location_tmp, location
        except Exception as e:
            pass
```

```
    return False, False


def detect_lng(text):
    if len(text) == 0:
        return "unknown"
    else:
        try:
            return langdetect.detect(text.lower())
        except langdetect.lang_detect_exception.LangDetectException:
            return "unknown"

def get_nested(object, property):
    result = []
    try:
        result = object[property]
    except:
        pass
    return result


def tweet_normalization_aggressive(text):
    text = re.sub(r'\s+&amp;?\s+', ' and ', text)
    text = re.sub(r'@[A-Za-z0-9_]+\b', ' ', text)
    text = re.sub(r"\b\d\d?:\d\d\s*[ap]\.?m\.?\b", '', text, flags=re.IGNORECASE)
    text = re.sub(r"\b\d\d?\s*[ap]\.?m\.?\b", '', text, flags=re.IGNORECASE)
    text = re.sub(r"\b\d\d?:\d\d:\d\d\b", '', text, flags=re.IGNORECASE)
    text = re.sub(r"\b\d\d?:\d\d\b", '', text, flags=re.IGNORECASE)
    text = re.sub(r'\bhttps?:\S+', '  ', text, flags=re.IGNORECASE)
    text = re.sub(r'\s+https?$', ' ', text, flags=re.IGNORECASE)
    text = re.sub(r'[^\w\d\s:\'",.\(\)#@\?!/’_]+', '', text)
    text = re.sub(r'\n', ' ', text)
    text = re.sub(r'\s{2,}', ' ', text)
    text = text.strip()
    return text
```

```
def get_noun_chunks(text):
    text = tweet_normalization_aggressive(text)
    r = Rake()
    r.extract_keywords_from_text(text)
    return r.get_ranked_phrases()

def get_ner(text):
    text = Text(text)
    ners = text.entities
    ner_loc = []
    ner_per = []
    ner_org = []
    if ners:

        for item in ners:
            txt = ""
            for i in item:
                txt = txt + " " + i

            if item.tag == "I-LOC":
                ner_loc.append(txt)
            if item.tag == "I-PER":
                ner_per.append(txt)
            if item.tag == "I-ORG":
                ner_org.append(txt)

    return ner_loc, ner_org, ner_per

print("connecting to elasticsearch")
es = Elasticsearch([{"host": "173.212.230.46", "port": 9200}])

es.indices.create(index=index, ignore=400)
```

```
print("importing vader sentiment analyzer")
analyser = SentimentIntensityAnalyzer()

auth = tweepy.auth.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
listener = StdOutListener()
stream = Stream(auth, listener)
stream.filter(track=["Coronavirus", "corona", "covid19", "SARSCoV2"], is_async=True)
```