VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Pavel Rýdl

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO ANALÝZU DAT Z TWITTERU
TWITTER DATA ANALYSIS TOOL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Pavel Rýdl
AUTHOR

VEDOUCÍ PRÁCE — Ing. Zoltán Galáž
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Pavel Rýdl ***ID:*** 165417
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Aplikace pro analýzu dat z Twitteru

POKYNY PRO VYPRACOVÁNÍ:

V rámci bakalářské práce bude navržena, implementována a otestována aplikace pro automatické stahování a analýzu dat z Twitteru. Aplikace bude vytvořena v programovacím jazyku Python, bude obsahovat grafické uživatelské rozhraní a možnost generování reportů založených na technikách zpracování přirozeného jazyka (tzv. natural language processing).

DOPORUČENÁ LITERATURA:

[1] BIRD, Steven, Ewan KLEIN a Edward LOPER. Natural language processing with Python. Beijing: O'Reilly, c2009. ISBN 978-0-596-51649-9.

[2] MERTZ, David. Text processing in Python. Boston: Addison-Wesley, c2003. ISBN 0-321-11254-7.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: Ing. Zoltán Galáž
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato práce se zabývá vytvořením aplikace pro automatické stahování a analýzu dat z Twitteru založené na technikách zpracování přirozeného jazyka. Aplikace je vytvořena v programovacím jazyku Python. Pro tvorbu aplikace bylo použito vývojové prostředí Jupyter Notebook, ve kterém byla celá aplikace včetně GUI implementována. V části teorie je popsána problematika stahování dat a analýza pomocí zpracování přirozeného jazyka. V části implementace je popsáno řešení aplikace v jednotlivých krocích, jako jsou vytvoření aplikace na straně Twitteru, stahování, předpříprava, analýza dat s technikami zracování přirozeného jazyka a následná vizualizace. Implementována byla i analýza bez použití technik zpracování přirozeného jazyka. Testování probíhalo na tweetech, které obsahovali zmínku o americkém prezidentu Donaldovi Trumpovi.

KLÍČOVÁ SLOVA

analýza, Jupyter Notebook, Python, sociální síť Twitter, vizualizace, zpracování přirozeného jazyka

ABSTRACT

This work deals with the creation of an application for automatic downloading and Twitter data analysis based on natural language processing techniques. The application is created in the Python programming language. A development environment Jupyter Notebook was used for creating the application, where the entire application, including GUI, was implemented. In the section of theory are data downloading issues and data analysis by natural language processing described. In the part of implementation there is solution of the application described in several steps, such as creating the application on the Twitter's side, downloading, preprocessing, data analysis with techniques of natural language processing and following visualization. There was also a technique with no natural language processing implemented. Testing run on tweets that contained reference to US president Donald Trump.

KEYWORDS

analysis, Jupyter Notebook, natural language processing, Python, social network Twitter, visualization

RÝDL, Pavel. *Aplikace pro analýzu dat z Twitteru*. Brno, 2019, 49 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Zoltán Galáž, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Aplikace pro analýzu dat z Twitteru“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Zoltánovi Galášovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora

Obsah

Seznam obrázků

Seznam výpisů

Úvod

Počítačové zpracování přirozeného jazyka je v dnešní době součástí důležité a perspektivní disciplíny, která se dá zahrnout do oboru umělé inteligence. Jedná se o dynamicky se rozvíjející oblast informatiky, která se v dnešní době stává čím dál tím víc potřebnější. Využívá metody pro rozpoznávání obrazů, řeči nebo porozumění jazyku. Dnešní doba je dobou sociálních sítí a aplikací komunikujících skrze internet, které jsou zahlceny informacemi.

Tato práce se věnuje oblasti analýzy dat ze sociální sítě Twitter, zejména pak analýzou textu ve zprávách, na které jsou aplikovány metody zpracování přirozeného jazyka. Práce je rozdělena na dvě části, kde v první části je probrána teorie a problematika ve všeobecném pojetí. Druhá část se věnuje tvorbě aplikace v programovacím jazyku Python ve výpočetním prostředí Jupiter Notebook.

Teoretická část je rozdělena na dvě hlavní kapitoly: sociální sítě se zaměřením na Twitter a zpracování přirozeného jazyka. Kapitola o sociálních sítích shrnuje historii sociálních sítí a nastíňuje pohled na dnešní sociální sítě. Poté už se výhradně věnuje sociální sítí Twitter. Kapitola zpracování přirozeného jazyka poskytuje základní přehled problematiky, například co si pod tímto pojmem představit, kde se v dnešní době tyto metody využívají, na jakých principech pracují a problémy, kterým tyto metody zpravidla čelí.

Praktická část se zaměřuje na použité technologie a na programové části aplikace. Zpracována je zde například ukázka vytvoření aplikace na stránkách Twitter, nutná pro samotný běh programu. Stahování dat ze sociální sítě Twitter za pomoci rozhraní Twitter API. Dále pak tato práce pokračuje čištěním a předpřípravou těchto dat, které jsou nedílnou a velice důležitou součástí pro přípravu kvalitní analýzy, dále pak samotnou analýzou dat a v neposlední řadě také vizualizace dat.

V části výsledky jsou představeny ukázky zpracovaných dat a některých možných vizualizací. Textová data v podobě zpráv na sociální síti, budou zaměřena na uživatele Donald Trump, s názvem účtu „@realDonaldTrump“.

1 Teorie

1.1 Sociální sítě

Nejstarší sociální sítě existovaly již v roce 1995, jednou z takových dodnes používaných sítí je Yahoo. Tyto platformy měly především usnadnit vzájemnou komunikaci mezi lidmi prostřednictvím „chatovacích“ místností. Teprve koncem 90. let se uživatelské profily jako takové dostaly do popředí a staly se mezi uživateli oblíbenými. Tyto účty nabízeli základní správu účtu a navíc nabízeli možnost zvolit si své přátelé. Největší změna však nastala v roce 2004 při vytvoření sociální sítě Facebook. Stal se tak hlavním faktorem, změnou pro lidské životy, podnikání a celý svět.

Od posledních několika let se v sociálních médiích objevují nové aplikace s různými interakčními modely ve srovnání se sítěmi jako Facebook, LinkedIn nebo Twitter. Mezi tyto aplikace můžeme řadit Google, Pinterest, Instagram, Tinder a další.

Sociální sítě prošly v posledních letech dramatickým růstem, poskytují mimořádně vhodný prostor pro okamžité sdílení multimediálních informací mezi jednotlivci, skupinami a jejich přáteli. Sociální sítě poskytují silný odraz struktury a dynamiky společnosti. Dramatický růst sociálního obsahu generovaného uživatelem je revoluční. Sociální sítě dali možnost vzniku novým aspektům věd a novým technologiím.

Jelikož se sociální sítě neustále vyvíjejí, existuje celá řada výzkumů v sociálních sítích, které jsou v současné době zkoumány výzkumnými komunitami. Mezi takové výzkumy můžeme zařadit i analýzu vztahů a komunikace mezi členy komunity, která může odhalit nejvlivnější uživatele ze sociálního hlediska. Informace získané ze sociální sítě mohou být použity i jako užitečný nástroj v rámci bezpečnosti. Vzniknutí kybernetického dohledu nad ochranou kritické infrastruktury je dalším významným studiem.

Stejně jako v každé lidské komunitě, tak i on-line sociální sítě čelí kritickým i etickým otázkám. Ochrana osobních informací a mnoho dalších problémů vyžadují zvláštní pozornost [1, 2].

1.2 Twitter

Twitter je jednou z nejpopulárnějších sociálních sítí, tzv. mikro-blogingových služeb na světě. Jedná se o prostředek pro sdílení informací s okolním světem. Umožňuje odesílat a číst krátké zprávy, nazývané „tweety“. Uživatelé přistupují k Twitteru přes webové rozhraní, SMS, nebo prostřednictvím mobilní aplikace. Na uživateli je poté samotné rozhodnutí, jestli se zaregistruje na tuto sociální síť, nebo jestli

zůstane nezaregistrován a Twitter mu dovolí pouze čtení tweetů. Pokud se uživatel rozhodne zaregistrovat, je mu umožněno využívat více funkcí, jako zveřejňování tweetu, soukromé zprávy, sledování uživatelů a mnoho dalších.

Sociální sít Twitter uvádí, že má více než 326 milionů aktivních uživatelů k datumu 26. 10. 2018.

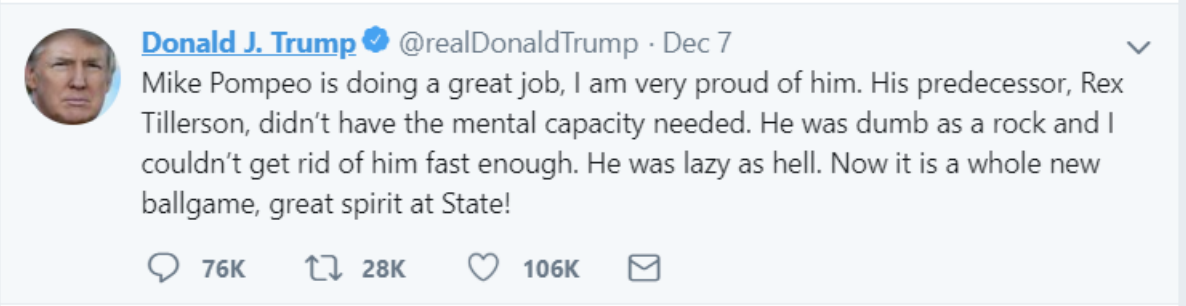
Twitter je v dnešní době velice zajímavou alternativou pro výzkum sociálního chování, protože umožňuje svobodné šíření informací mezi lidmi a skupinami. O tomto šíření informací v rámci sítě můžeme říci, že je velice podobné jako způsob šíření informací v reálném životě [1]. Díky tomu můžeme analýzou zjistit spoustu informací buď o konkrétním jedinci a jeho chování, nebo i celých skupin.

Příklad uživatelského účtu můžeme vidět na obrázku 1.1. Jedná se o uživatelský účet prezidenta Donalda Trumpa, který bude sloužit pro analýzu dat.

Obr. 1.1: Zobrazení uživatelského účtu realDonaldTrump na sociální síti Twitter.

Tweet

Zpráva, která se nazývá „tweet“, na obrázku 1.2, je textový příspěvěk dlouhý 140 znaků. Tweet může obsahovat odkaz na video, fotky, nebo další média, která jsou hostována kdekoliv na Internetu. Kvůli narůstajícím nárokům uživatelů byla v listopadu roku 2017 tato velikost navýšena až na 280 znaků. Každý tweet obsahuje atributy jako název autora, unikátní ID, časovou známku, geo. data sdílená uživatelem a mnoho dalších. K tweetu může být připojeno přes 150 dalších atributů [3].



Obr. 1.2: Zobrazení tweetu uživatele realDonaldTrump.

Způsoby stahování dat z Twitter

Samotný Twitter disponuje API [1] rozhraním, které se nazývá „Twitter API“. Umožňuje vývojářům třetích stran vytvářet software, který komunikuje s Twitter. Na svých stránkách pro vývojáře, v sekci dokumentace popisuje detailně veškeré informace, které jsou potřebné ke komunikaci [3].

Produkty, které Twitter pro vývojáře poskytuje:

- Standard Twitter API, která obsahuje REST API [2] a Streaming API, jedná se o bezplatnou verzi, vhodnou pro základní testování,
- Premium API, toto rozhraní v sobě zahrnuje lepší filtrování a analýzu dat, nebo možnost stahovat historická data až 30 dnů nazpět,
- Enterprise API, rozhraní určené pro podniky,
- dodatečné API, jako například Ads API pro reklamy a nebo rozhraní pro webové stránky.

Omezení a limity API

Většina služeb, které jsou poskytovány zadarmo, jsou nějakým způsobem omezovány a i pro využití Twitter API tomu není jinak. Použití služby Twitter API je omezeno limity, které jsou založeny na, tzv. „fair use limits“ – omezení spravedlivého použití. Standardní API má několik typů omezení. Tyto omezení se stahují jak na uživatele, tak i na samotnou aplikaci využívající službu Twitter API. Pro vyhnutí se omezením je doporučena Streaming API, kde jsou tyto omezení minimální. Pokud aplikace zneužívá tyto limity, je zařazena do černé listiny a poté už nemůže získat odpovědi od API [3].

[1] API (architektura rozhraní pro programování aplikací – Application Programming Interface)

[2] REST API (architektura rozhraní pro distribuované prostředí aplikací – REpresentational State Transfer. Lidé zmiňující Twitter API nebo Google API, vždy mluví o tzv. REST API, která funguje na stejném principu jako webová stránka. Klient požádá server o data přes HTTP protokol.

1.3 Zpracování přirozeného jazyka

Zpracování přirozeného jazyka (Natural Language Processing, zkráceně NLP) je odvětví umělé inteligence, které čerpá z mnoha disciplín včetně počítačové vědy, výpočetní lingvistiky a umělé inteligence, zabývá se studiem a zpracováním lidského jazyka. Softwarové nebo hardwarové komponenty v počítačovém systému analyzují lidský jazyk s cílem zaplnit mezeru mezi lidskou komunikací a počítačovým porozuměním. Využívá techniky, jako textové dolování a nebo textovou analýzu pro zpracování smysluplných informací z textu přirozeného jazyka. Cílem NLP je, aby počítačové systémy skutečně pochopily jazyk tak, jak jej chápe samotný člověk [4].

NLP lze také vnímat jako studii umělé inteligence (Artificial intelligence, neboli AI) a proto mnohé existující algoritmy a metody, včetně neuronových síťových modelů jsou využívány pro řešení problémů NLP [5].

Techniky zpracování textu zahrnují tokenizaci, normalizaci textu, nebo čištění dat. Jakmile jsou data ve standardním formátu, mohou být použity různé techniky hlubokého učení pro lepší pochopení dat. Oblíbené modelovací techniky jako klasifikace spamů, nebo hodnocení názorů (sentimentu) na sociálních sítích. Novější a složitější techniky mohou být použity například jako modelování témat, vnořování slov, nebo vytváření textu s použitím metody hlubokého učení, tzv. deep-learning.

Základní rozdělení

NLP můžeme dělit do dvou podskupin a to:

- pochopení přirozeného jazyka (Natural Language Understanding, neboli NLU),
- generování přirozeného jazyka (Natural Language Generation, neboli NLG).

Pochopení přirozeného jazyka

NLU je považována za první součást NLP a zabývá se pochopením přirozeného jazyka. Považuje se za vědu, která k řešení problému využívá oblast umělé inteligence (AI-Hard), nebo (AI-Complete). Jedná se o snahu udělat z počítače inteligentní jednotku, která by se inteligencí rovnala lidem, nebo silné umělé inteligenci [6]. Aby NLU převedla NL na užitečnou reprezentaci, vyžaduje následující analýzy:

- morfologickou analýzu,
- lexikální analýzu,
- syntaktickou analýzu,
- sémantickou analýzu,
- manipulaci s nejednoznačností,
- integrace diskurzu,
- pragmatickou analýzu.

Generování přirozeného jazyka

NLG je považována za druhou součást NLP. Je to věda, která se zabývá generováním přirozeného jazyka. NLG je definována jako proces vytváření přirozeného jazyka na výstupu stroje. Bez ohledu na přirozený jazyk, by měla být data logická. Za účelem generování logického výstupu používají mnohé systémy NLG základní fakta, nebo znalostní reprezentace. [7, 4]

Reálné využití v praxi

V dnešní době má NLP řadu využití. Mnoho lidí používá nástroje, které jsou založeny na metodách zpracování přirozeného jazyka, aniž by o tom věděli. Mezi takové můžeme zařadit například kontrolu pravopisu, automatické doplňování, tzv. predikci, spam filtry, strojový překlad, osobní asistenti jako Siri, Alexa nebo Google asistent, automatizace zákaznických služeb, monitorování sociálních sítí, automatické přehledy a mnoho dalších. Některé nejznámnější nástroje jsou uvedeny pod textem.

Vyhledávání vhodných informací

Obrázek 1.3 znázorňuje výběr vhodných informací, které se nejvíce blíží k zadanému slovu.

Kontrola pravopisu a gramatiky

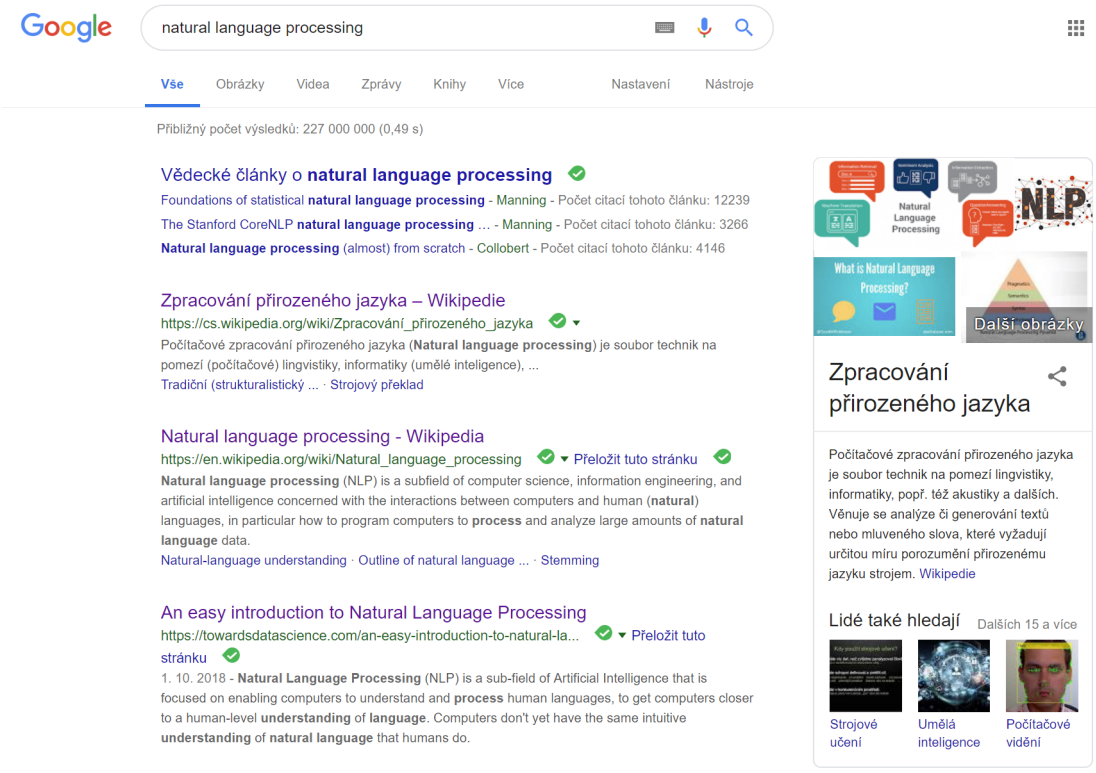
Na obrázku 1.4 je ukázáno použití kontroly pravopisu a gramatiky ve vyhledávači od Googlu a následný návrh nesprávné alternativy. Při zadání nesprávného tvaru slova, jako např. „narural“ dojde k vyhodnocení zadaného slova a opravě na tvar „natural“.

Predikce slova

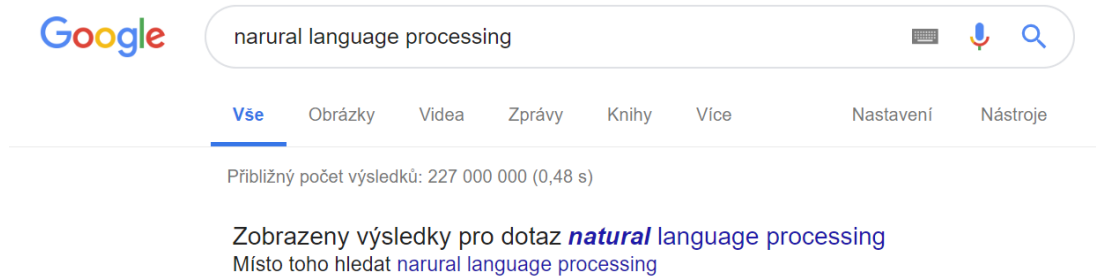
Na další ukázce 1.5 je vidět předpovídání možných alternativ, které by mohl chtít uživatel naleznout.

Strojový překlad a syntéza řečí

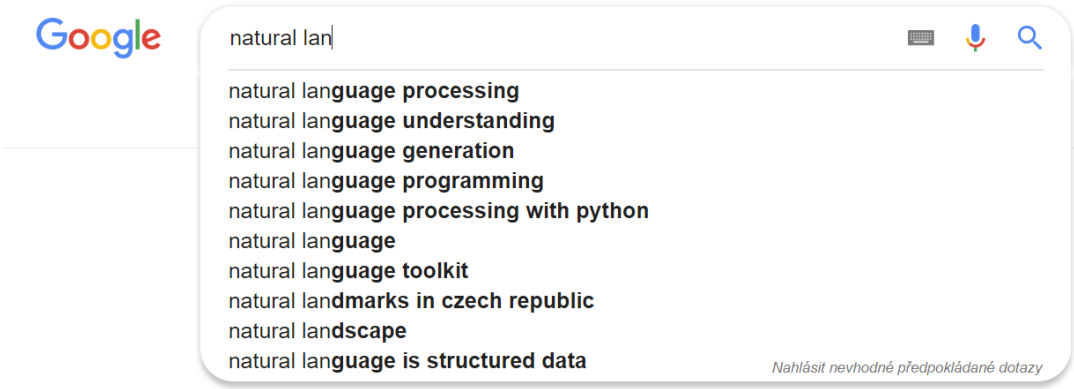
Obrázek 1.6 znázorňuje překlad z jednoho jazyka do druhého. Červeně je pak označena funkce, která umožňuje syntézu řeči.



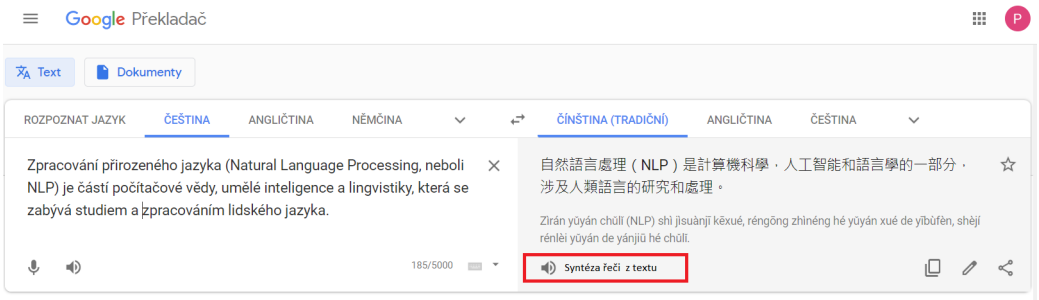
Obr. 1.3: Ukázka vyhledávání vhodných informací.



Obr. 1.4: Ukázka kontroly pravopisu a gramatiky.



Obr. 1.5: Ukázka predikce slova.



Obr. 1.6: Ukázka strojového překladu a syntézy řeči.

Problémy při zpracování přirozeného jazyka

Lidský jazyk je neuvěřitelně složitý, rozmanitý, vysoce nejednoznačný, rovněž se stále mění a vyvíjí. Lidé mají možnost se takřka nekonečným způsobem vyjadřovat a to jak ústně, tak i písemně. Proces čtení a porozumění jazyka je mnohem složitější, než se na první pohled zdá. Jazyk jako takový, je velice chudý na formální pochopení a popis pravidel, kterými se řídí. Existují stovky jazyků a dialektů a v každém z nich existuje jedinečný soubor gramatických a syntaktických pravidel. Avšak lidem nedělá problém porozumět jazyku a jsou schopni vyjadřovat, vnímat a interpretovat tyto komplikované významy. V běžném rozhovoru mezi lidmi jsou slova velmi často nečitelná, ať už ve formě různých signálu, výrazu, nebo ticha. Přesto jsme my lidé schopni těmto základním posunkům porozumět, avšak tyto vlastnosti počítači chybí. Příkladem takové nejednoznačnosti může být například interpretace homonym, kde slova mají stejnou podobu, ať už zvukovou nebo grafickou, ale naprosto odlišný význam. Jedno z takových hononym je kohoutek, může se jednat o zvíře nebo o vodovodní kohoutek. Každý si většinou dá větu do kontextu a pochopí celkový význam. Existuje celá řada těchto problémů při strojovém zpracování přirozeného jazyka.

Stroje mohou být naprogramovány, aby chápali kód, jako například Java [3], Python [4] ..., nebo aby vyřešili matematické či logické příklady, avšak to, aby stroje pochopili přirozený jazyk je velice náročné [8, 9].

Metody strojového učení excelují v problematických oblastech, kde je velmi těžké nastavit soubor pravidel. Jazyk je symbolický a diskrétní. Základními prvky písemného jazyka jsou znaky. Znaky tvoří slova, které zase označují objekty, pojmy, události, akce, myšlenky, atd.

Příklady některých metod zpracování textu NLP:

- Tokenizace – Tokenization,
- Stematizace – Stemming,
- Lematizace – Lemmatization.

Tokenizace

Tokenizace (Tokenization) může být definována jako proces rozdělení textu na menší části, tzv. tokeny. Slovo, neboli (token) je nejmenší možná část, které stroj dokáže porozumět, nebo zpracovat. Programy, které slouží pro tokenizaci textu se nazývají tokenizery. Každý textový řetězec musí nejprve projít procesem tokenizace a až poté může být zpracován. Proces tokenizace je prvotní dělení textu na smyslupné tokeny a ta se může lišit podle potřeby NLP aplikace [10]. Například z angličtiny můžeme

[3] `https://www.java.com/en/`

[4] `https://www.python.org/`

jednoduše pouhými regulárními výrazy vyselektovat pouze slova nebo čísla. Pro Japonštinu a podobné jazyky to bude velmi náročný úkol.

Rozdělování slov můžeme dělit na:

- rozdělování podle slov – Word Tokenizer,
- rozdělování podle vět – Sentence Tokenizer,
- rozdělování podle regulárního výrazu – Regexp Tokenizer.

Stematizace

Stematizace (Stemming) je metoda nalezení kořene slova. Takovýto algoritmus se nazývá stemmer. Stematizace se používá ve vyhledávačích, kde dokáže vyhledávat slova bez ohledu na konkrétní tvar. Během tohoto procesu dochází k odstranění morfologické předpony nebo koncovky. Příkladem tohoto algoritmu může být například „muž-i, muž-e, muž-ovi“, kde výstupem bude základ slova, tedy muž [4].

Lematizace

Lematizace (Lemmatization) je velmi podobná metodě stematizace. Operace vrací základní gramatický tvar slova, tedy kořen, který se nazývá lemma. Příklad pro tento algoritmus může být například pro slova „kamínek, kamíneček, kamenný, kamenovat, atd.“, kde výstup bude slovo kámen [4].

Datové sady

Korpus je soubor písemného nebo mluveného přirozeného jazyka, uloženého na počítači, či datovém úložišti v elektronické podobě. Slouží k detailnějšímu pochopení, jakým způsobem je jazyk používán. Přesněji řečeno, korpus je systematicky uložená sbírka autentického jazyka, která se používá pro jazykovou analýzu. Elekronické korpusy slouží ke zkoumání přirozeného jazyka. V současné době je neodmyslitelným nástrojem v oblasti NLP. Aby mohli být vyvinuty aplikace NLP, potřebujeme korpus, který je napsán nebo namluven z přirozeného jazykového materiálu. Tento materiál, nebo údaje z něho se používají jako vstupní data. Z těchto dat se posléze snažíme zjistit fakta, které napomáhají vývoji aplikace NLP. Některé aplikace NLP mohou používat jeden korpus pro vstup, jindy se používají vícenásobné korpusy podle potřeby aplikace. Velikostí korpusu se dá do značné míry ilustrovat, jakým způsobem lidé používají jazyk. Existují korpusy, které mohou obsahovat více než 100 miliard slovních pozic [4].

Pomocí korpusu se provádějí některé statistické analýzy, například frekvenční distribuce, společné výskyty slov, atd. Textová data se schromažďují z písemných informací. Existuje celá řada takových zdrojů, které lze použít k získání písemné

informace, jako jsou novinové články, knihy, e-mailové zprávy, webové stránky, blogy a mnoho dalších. Dnešní dobu můžeme označit za digitální svět, ve kterém je velké množství textové informace všude kolem nás.

Předběžné zpracování datové sady

Mezi základní a společné problémy nezpracovaného textu patří:

- nekonzistentní názvy,
- chybějící data,
- duplikátní data.

Všechny tyto nedostatky by měli být odstraněny. Jeden ze způsobů je data nezahrnovat do celkové analýzy, druhý způsob je pokusit se data opravit.

Práce s nezpracovaným textem

Při práci s nezpracovanými daty se musí předpokládat, že ne všechny texty budou užitečné pro extrahování. Ve skutečnosti je pravděpodobné, že do souboru dat bude vloženo větší množství nepotřebných dat, tudíž celkový výsledek bude méně účinný. Pro odstranění nevyhovujícího textu jsou zapotřebí provést základní kroky [11].

Formátování je krok sloužící pro vygenerování datové sady. Měl by být zvolen tak, aby vyhovoval požadavkům aplikace, nebo se volí podle zkušeností pracovníka. Nejčastějším formátováním se kterým se lze setkat je formát JSON nebo data ve formátu CSV.

Čištění, pokud má množina dat chybějící hodnoty, je zvykem tyto záznamy dat odstranit nebo nahradit záznamy nejbližší vhodnou hodnotou. Zbytečné atributy dat lze také odstranit. Dále se odstraňují data, které jsou zbytečné pro budoucí korpus.

Transformace dat, tato fáze je v zásadě manipulace s daty, kde se mohou použít některé z kódovacích metod, nebo vektorové techniky.

Parsování dat

Parsování, které je také označováno jako syntaktická analýza, je jedním z úkolů NLP. Je definován jako proces zjišťování, zda je znaková sekvence napsaná v přirozeném jazyce v souladu s definovanými pravidly ve formální gramatice. Termín parsování byl odvozen z latinského slova pars (oration is), což znamená část řeči. Parsování může být rozděleno do dvou kategorií a to top-down a bottom-up [10].

Rozpoznávání slovních druhů

Rozpoznávání slovních druhů (Post of Speech, neboli POS Tags) vysvětluje část projevu, jakým způsobem je slovo používáno ve větě. Označováním části řeči je jedním z mnoha úkolů NLP. Je definován jako proces, při kterém se konkrétním slovům ve větě přiřazují jednotlivé značky. Tato značka identifikuje, zda je slovo podstatné jméno, sloveso, atd [10].

Rozpoznávání názvů entit

Rozpoznávání názvů entit (Name Entity Recognition, neboli NER Tags) je jednou z částí metody POS Tagging. Její úkol je pojmenování entity v reálném světě skutečným jménem. Pro příklad Francie, Donald Trump, Twitter budou označeny jako, Francie–město, Donald Trump–člověk, a Twitter jako společnost. Pro značkování NER je typický výstupní štítek, který identifikuje entitu ve větších kategoriích (osoba, organizace, umístění, atd). Vytvoření značky NER vyžaduje velké množství anotovaných dat [10, 11, 12].

Modelování témat

Tato metoda se snaží ve velkém množství nestrukturovaného textového obsahu najít vhodné téma podle kontextu. Primární úkol je identifikovat takto vzniklá témata podle dosavadních témat, které obsahuje korpus. Tento problém může být řešen mnoha různými cestami. Typicky se pro tento problém používá LDA (Latent Dirichlet allocation) a LSI (Latent sematic indexing). Klíčové faktory, které rozhodují o získání dobrých témat jsou: kvalita zpracování textu, rozmanitost témat, volba vhodného algoritmu, počet témat vygenerovaných algoritmem a konečné ladění algoritmu [13, 14].

Vnoření slov

Vnoření slov (neboli Word Embedding) je moderní přístup k reprezentaci textu při zpracování přirozeného jazyka. Poskytuje tzv. hustou vektorizaci slov, která se snaží zachytit význam slov. Vnořování slov poskytuje lepší výsledky než je tomu například u metod TF-IDF, které mají za výsledek velké řídké vektory, popisující převážně dokumenty, ale neberou v úvahu význam slov. Algoritmy, jako jsou Word2Vec a GloVe využívají modely neuronových sítí pro řešení problému zpracování přirozeného jazyka. Word2Vec je jednou z nejpopulárnějších technik, pomocí neuronové sítě. Byl vytvořen Tomášem Mikolovem v roce 2013. Metoda může být řešena pomocí dvou modelů, oba tyto modely zahrnují neurální sítě. První model je skip-gram, druhý

model je Bags-of-word (neboli CBOW). Oba tyto modely mají své výhody a nevýhody. Skip-gram funguje lépe s malým množstvím dat. CBOW je rychlejší a má lepší reprezentaci slov [15, 14].

Sumarizace textu

Sumarizace je užitečná zejména proto, že zestručňuje informace pro snadnější analýzu. Místo přečtení celého množství textu se vyberou pouze ty informace, které skutečně potřebujeme. Mezi dva hlavní typy přístupů a pohledů k shrnutí textu, tzv. sumarizace textu se řadí:

- extrakční shrnutí,
- abstraktní shrnutí.

Extrakční shrnutí

Princip této metody spočívá v tom, že získává klíčová slova nebo věty beze změn z původního textu.

Mezi extrakční shrnutí patří například algoritmus Textrank inspirovaný algoritmem PageRank od Google, který pomáhá identifikovat klíčové věty. Myšlenka tohoto algoritmu je, že věta, která je nejvíce podobná ostatním větám při průchodu bude pravděpodobně nejdůležitější věta v dané pasáži. Pomocí takové myšlenky lze vytvořit graf věty, kde se spojí všechny věty, které jsou si podobné a z nich se pak vytvoří souhrn.

Další z algoritmů patřících do skupiny extrakčních shrnutí patří algoritmus **Term Frequency – Inverse Document Frequency** (neboli TF–IDF) používá se k určení významu slova v dokumentu. Základní algoritmus vypočte frekvenci slova v dokumentu, které je vynásobené číslem logaritmické funkce počtu dokumentů obsahujících tato slova, nad celkovým počtem dokumentů v datové sadě. Pomocí tzv. důležitosti každého slova může poté vypočítat i důležitost každé věty. Za předpokladu, že věty, které dosáhnou největšího počtu se vyhodnotí jako důležité a jsou tudíž vhodné pro shrnutí dokumentu, tvz. sumarizací. Tento algoritmus se skládá ze dvou částí, které jsou popsány níž [10].

Term Frequency – TF reprezentuje normalizaci výskytu slova s ohledem na velikost korpusu. Udává frekvenci slova v daných dokumentech. Je to poměr počtu, kolikrát se slovo objeví v dokumentu, ve srovnání s celkovým počtem slov v tomto dokumentu. Zvyšuje se, jak se zvyšuje počet výskytů tohoto slova v dokumentu. Každý dokument má svůj TF. Je definována jako:

$$tf_{i,j} = \frac{n_{i,j}}{\sum kn_{k,j}}, \tag{1.1}$$

kde $n_{i,j}$ je počet výskytů slova t_i v dokumetu d_j. Jmenovatel reprezentuje součet počtů výskytů všech slov v dokumentu d_j, tzn. jeho délku.

Inverse Document Frequency – IDF používá se k výpočtu vzácných slov napříč všemi dokumenty v korpusu. Slova, která se v korpusu vyskytují jen zřídka, mají vysoké skóre IDF. Jinými slovy reprezentuje nedůležitost slova. Čím častěji se slovo vyskytuje v dokumentu, tím méně je důležité.

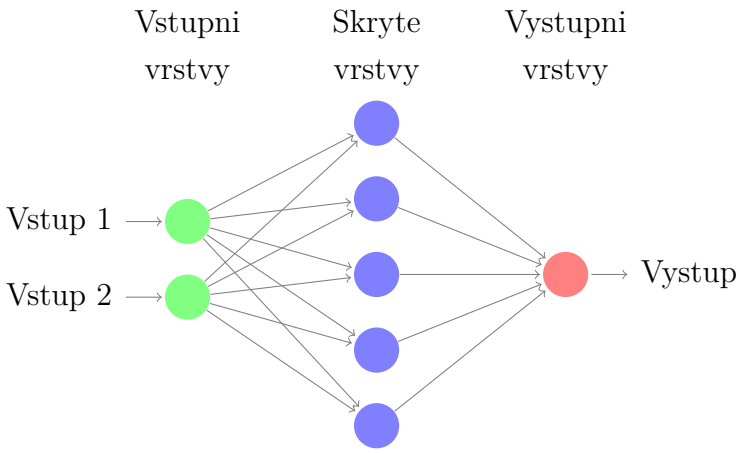
$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}, \tag{1.2}$$

kde $|D|$ je velikost databáze dokumentů, tzn. počet všech dokumentů, ve kterých je hledáno a $|\{j : t_i \in d_j\}|$ je počet dokumetů, které obsahují dané hledané slovo i.

Abstraktní shrnutí

V porovnání s extrakčním shrnutím se líší tím, že abstraktní shrnutí se více blíží tomu, co obvykle lidé očekávají při shrnutí textu. Je to proces, který se snaží pochopit originální dokument a přeformulovat tento dokument na mnohem kratší text se zachycením klíčových bodů a přitom zachovat stejný význam. Abstrakce textu se provádí především pomocí koncepce umělých neuronových sítí [10].

Umělé neuronové sítě jsou výpočetní počítačové systémy, inspirované biologickými neuronovými sítěmi. Takové systémy se učí z příkladů a to obvykle bez předchozích znalostí daného problému, například kontrola spamů v emailu. Umělé neuronové sítě se skládají z umělých neuronů, nazývajících se jednotky, obvykle uspořadáné v sérii vrstev. Na obrázku 1.7 je zobrazena nejběžnější architektura modelu neuronové sítě [10].



Obr. 1.7: Model neuronové sítě[16].

2 Implementace

2.1 Technologie

Mezi dvě hlavní technologie, které se využívají pro zpracování přirozeného jazyka se řadí zejména Python a Java. Obě tyto technologie disponujou řadou kvalitních knihoven. Pro Javu můžeme uvést například tyto knihovny: Freeling [1], OpenNLP[2], LingPipe[3] a další. Pro Python jsou to knihovny jako NLTK[4], Gensim[5], spaCy[6], Scipy[7], Scikit-learn[8] a další.

Pro tuto aplikaci byl zvolen programovací jazyk Python, který se velmi často používá pro věděckou analýzu. Výhody Pythonu jsou dostatečná rychlost implementace a rozšiřitelnost kódu. Python se řadí mezi vysoko–úrovňové skriptovací jazyky. Byl navržen počítačovým programátorem Guido van Rossum.

Konečná aplikace byla vyvíjena v Jupyter Notebook. Projekt Jupyter existuje pro vývoj tzv. otevřených standardů a služeb pro interaktivní výpočetní techniku napříč desítkami programovacích jazyků.

Jupyter Notebook

Jupyter notebook je otevřený software webové aplikace, umožňující vytváření a sdílení dokumentů. Lze v něm kombinovat spustitelný programovací kód, formátovaný text, matematické operace, grafy, atd.

Gensim

Gensim je otevřená softwarová knihovna pro zpracovávání přirozeného jazyka se zaměřením na modelování témat. Společnost Gensim byla vyvinuta a je vedena českým výzkumným pracovníkem pro zpracování přirozeného jazyka Radimem Řehůřkem a jeho společností RaRe Technologies[9]. Nejedná se o univerzální knihovnu pro výzkum jako je NLTK, ale je zaměřena na modelování témat a podporu implementace Word2Vec pro učení slovních vektorů z textu.

[1] `http://nlp.lsi.upc.edu/freeling/`
[2] `http://opennlp.apache.org/`
[3] `hhttp://alias-i.com/lingpipe/`
[4] `http://nlp.lsi.upc.edu/freeling/`
[5] `http://radimrehurek.com/gensim/`
[6] `https://spacy.io/`
[7] `http://www.scipy.org/`
[8] `http://scikit-learn.org/stable/`
[9] `https://rare-technologies.com/`

Práce s daty Pandas a Dataframe

Pro manipulaci a následnou analýzu dat byla zvolena softwarová knihovna pandas. Nabízí datové struktury, tzv. DataFrame a operace pro manipulaci s numerickými tabulkami. Podobné datové struktury můžeme najít například v SQL [10] databázích nebo souborech CSV [11].

2.2 Programové řešení

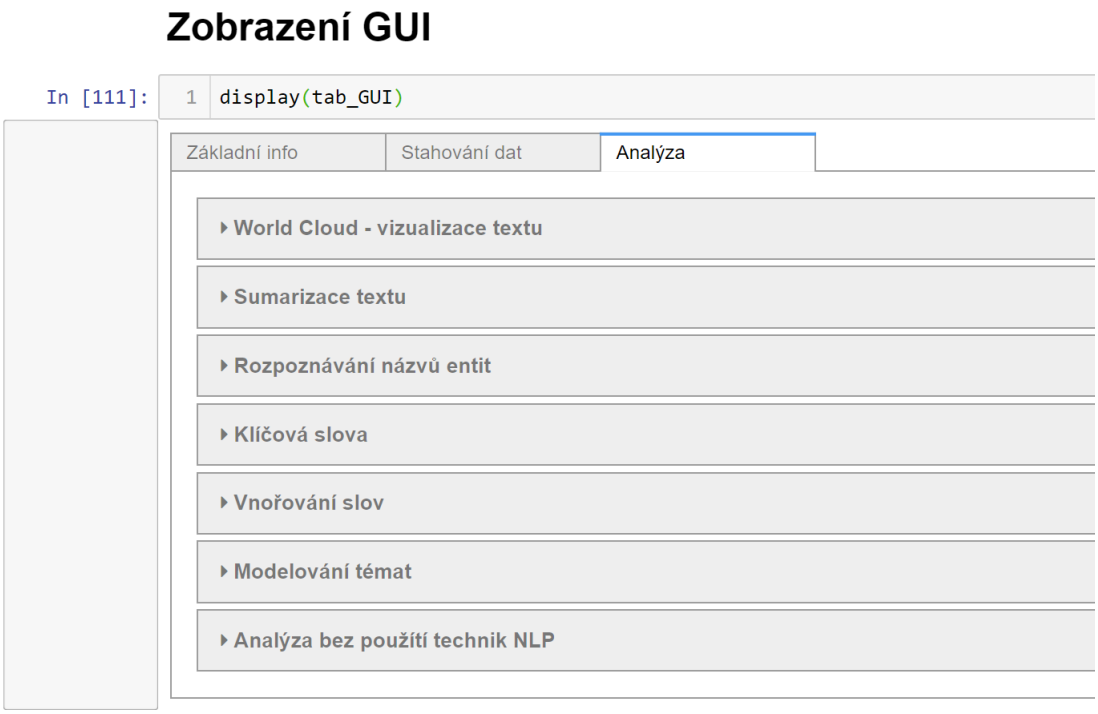
Návrh GUI

Jupyter Notebook využívá pro GUI tzv. widgety, které jsou v prohlížeči reprezentovány jako ovládací prvky, např. posuvník, tlačítko, textové pole. Jsou využívány k vytváření interaktivních grafických uživatelských rozhraní. Pro aplikaci nebylo potřebné předem připravovat návrh GUI. Jednotlivé komponenty byly implementovány nezávisle na sobě a kompletovány dohromady podle potřeby. K rozdělení hlavních prvků aplikace byly použity záložky, viz obrázek 2.1. Jednotlivé metody jsou uloženy v rozklikávacím boxu. Pro každou metodu pak byly použity komponenty tak, aby co nejvíce usnadňovali analýzu. Záložka s názvem „Základní info“ obsahuje souhrn informací ohledně velikostí souborů, počtu tweetů, atd. Záložka s názvem „Stahování dat“ obsahuje celkem tři boxy, první z boxů obsahuje ukázku stahování tweetů, kde je pevně nastavený filtr pro odchytávání tweetů. Druhý box obsahuje zobrazení tweetu ze souboru, kde je uložen v JSON podobě. Třetí box pak slouží k prozkoumání tweetů a jejich textů uložených v DataFramu. Poslední záložkou je záložka s názvem „Analýza“, obsahuje všechny implementované metody, které budou jednotlivě popsány v dalším textu, včetně interaktivního GUI.

Registrace - vytvoření aplikace na Twiteru

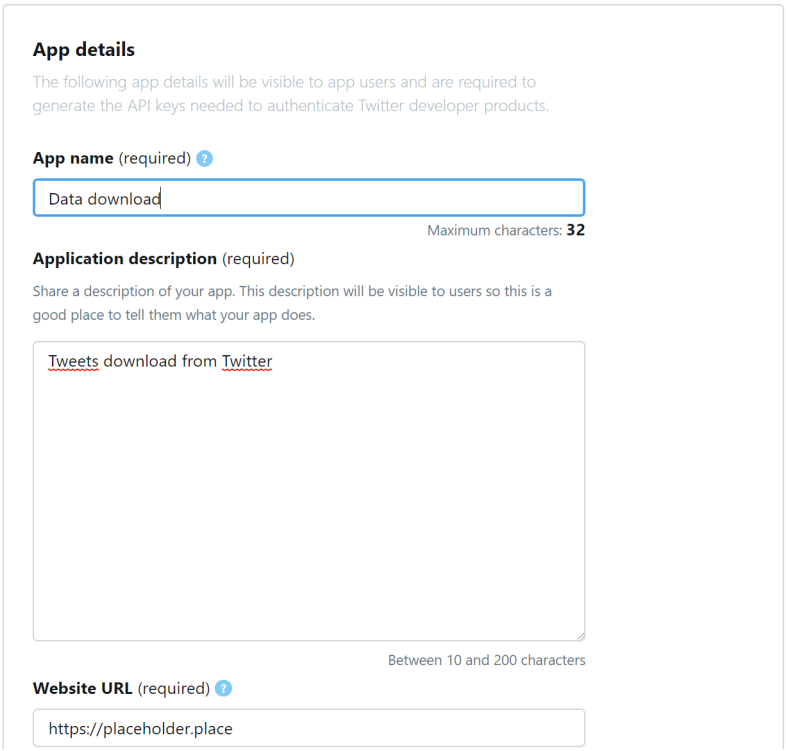
Aby aplikace mohla využívat přistup k sociální síti Twitter a taky k samotným tweetům, je nejprve nutná registrace na stránkách Twitteru [12]. Po úspěšné registraci je vytvořen klasický uživatelský účet, který umožňuje uživateli používat Twitter a jeho základní funkce. Toto však ještě nestačí, aby samotná aplikace mohla přistupovat k datům přes API, proto je potřeba vytvořit účet na stránce pro vývojáře [13]. Při úspěšném vytvoření tohoto účtu lze vytvořit novou aplikaci, v které stačí vyplnit povinné údaje 2.2, jako jsou jméno aplikace, krátký popis a URL (stačí vyplnit,

[10] (standardizovaný strukturovaný dotazovací jazyk – Structured Query Language)
[11] (hodnoty oddělené čárkami – Comma-separated values)
[12] `https://twitter.com/`
[13] `https://developer.twitter.com/`

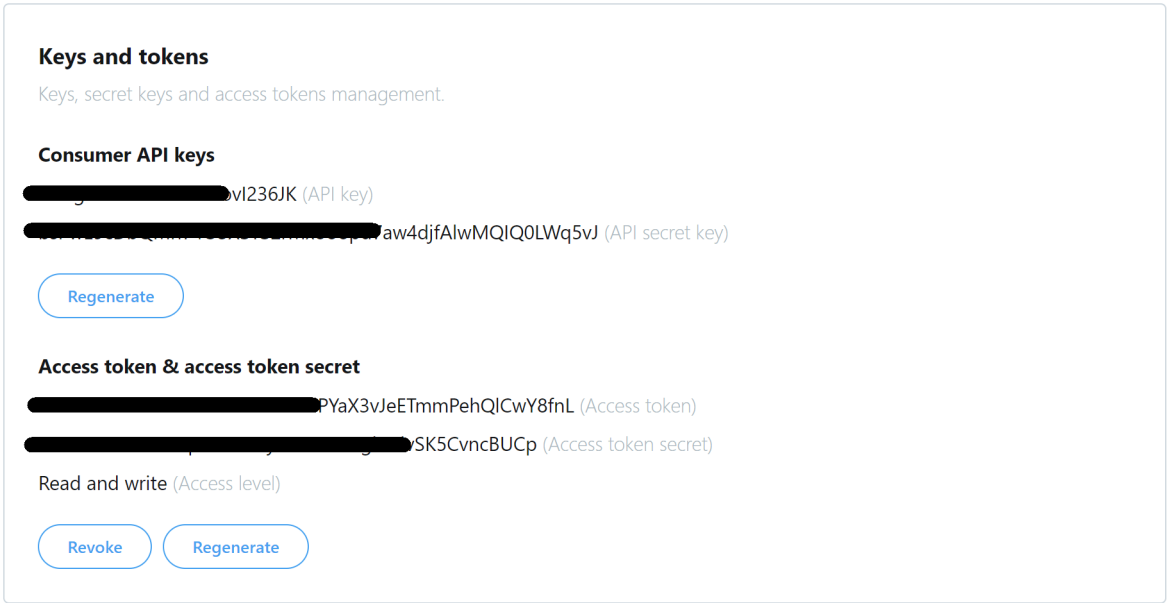


Obr. 2.1: Ukázka hlavního okna GUI.

např. https://placeholder.place) a poté se vygenerují potřebné klíče a přístupové tokeny 2.3, které slouží pro autorizaci a autentizaci pro rozhraní API. Tyto údaje pak budou využity následně při stahování dat.



Obr. 2.2: Ukázka vytváření aplikace na Twitteru - detail.



Obr. 2.3: Ukázka vytváření aplikace na Twitteru - přístupové tokeny.

Příprava stahování a ukládání dat

Existují dva způsoby stahování dat. První varianta je stahování dat zpětně v čase a druhá varianta je stahovaní dat v přímém přenosu, tzv. ze streamu.

První varianta, kde se data stahují zpětně je vhodná na analýzu dotyčného, kterému patří daný uživatelský učet. Výsledkem zde budou tweety napsané, nebo retweetnuté z jeho vlastního účtu. K této první variantě však patří nevýhody, které byly uvedené v teoretické části a to takové, že při běžné verzi Twitter API jsou k dispozici přibližně jen týden stará data. Stahování dat v přímém přenosu se hodí při analyzování tzv. sledujících, kde výsledkem jsou tweety napsané lidmi, kteří ve svém tweetu označují daného uživatele. I tato metoda má nevýhody, kterými může být například to, že jsou k dispozici jen tweety, které byly odchycené v daný den a čas. Oproti první metodě není svázana stejnou limitací. Limity této metody jsou opět uvedené v teoretické části.

Pro aplikaci bylo vhodné ubírat se druhou variantou a to stahováním dat v přímém přenosu pomocí knihovny Tweepy.

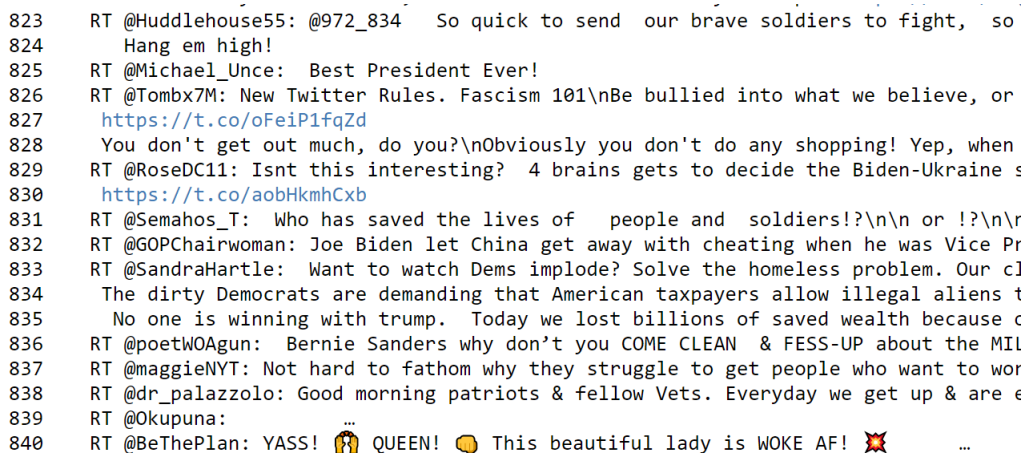
Při používání streamovacího rozhraní API služby Twitter, je třeba dbát na omezení rychlosti. Pokud klient překročí omezený počet pokusů o připojení k streamovanému API v určitém časovém okně, obdrží HTTP [14] chybu 420. Doba, kterou musí klient čekat po obdržení chyby se exponenciálně zvyšuje při každém neúspěšném pokusu. K chybě by však nemělo dojít, protože aplikace využívá jen jedno připojení, které následně i ukončí.

Filtrované data (tweety) jsou zachyceny ze streamu a následně uložena do textového souboru. Jedná se o obyčejný textový soubor s koncovkou .txt, kde jeden řádek souboru obsahuje informace týkající se jednoho tweetu ve formátu JSON. Struktura jednoho tweetu může vypadat následovně C.1. Základní struktura tweetu se nemění, avšak může se změnit počet parametrů připnutých ke tweetu, viz teoretická část. Vyzobrazený tweet byl náhodně vybrán.

Předběžné zpracování dat

Předběžné zpracování dat je jedním z nejdůležitějších kroků pro správnou analýzu dat. Jedná se o prvotní transformaci dat do srozumitelného formátu. Textová data z tweetů jsou často neúplná, nekonzistentní a nebo obsahují chyby. Všechny tyto atributy do velké míry ovlivňují výsledek. Neexistuje přesný návod, jak dosáhnout kvalitní předpřípravy dat. Každá metoda vyžaduje své specifické úpravy. Obrázek 2.4 znázorňuje nezpracovaný text v čisté podobě.

[14] HTTP (internetový prokol pro komunikaci – Hypertext Transfer Protokol)



Obr. 2.4: Ukázka textu před zpracováním.

Proto před samotným uložením do DataFramu se na každý text obsažený v tweetu použije funkce, která odstraní nepotřebné data. Pro předběžné zpracování dat byly odstraněny z textů URL linky, označení uživatelů, odstranění bílých znaků – mezer, odstranění interpunkce, odstranění emotikonů, atd. Pro tuto fázi byla použita knihovna gensim s funkcí *simple_preprocess* a vytvoření vlastních funkcí s regulárními výrazy.

V neposlední řadě proběhlo odstranění tzv. „stop words“. Jde o slova, která se vyskytují v daném textu velmi často, ale nenesou žádnou významovou informaci při analýze, například spojky, předložky, atd. Pro tyto slova neexistuje univerzální seznam a jsou zvolena v závislosti na typu analýzy. Pro aplikaci byla použita knihovna NLTK, která obsahuje předem definovaný seznam. Do tohoto seznamu se dají přidávat další nehodící se slova a tak rozšiřovat tento seznam. Existuje celá řada metod a postupů, pro předběžné zpracování dat, ale pro účely samotné analýzy textových dat z Twitteru je toto řešení dostačující. Na obrázku 2.5 je vidět finální zpracovaný text. Po finálním zpracování si lze všimnout, že se v textu stále vyskytují slova, která nemají význam. Pokud by se některé v textu opakovali víckrát, bylo by vhodné tyto slova zahrnout do „stop word“.

WordCloud

Pokud jde o jakýkoliv druh zpracování přirozeného jazyka je vykreslování pomocí „Word Cloud“ první možná varianta. Jedná se o vizualizační nástroj, kde velikost písma reprezentuje četnost v rámci daného textu. Tuto vizualizaci můžeme najít na mnoha webech, včetně blogů. Výsledkem je tzv. word cloud, poskládaný ze slov, které se nacházejí ve tweetech. Pro výslednou vizualizaci byla použitá knihovna wordcloud

```
823    quick send brave soldiers fight slow figure
824    hang em high
825    best president ever
826    new twitter rules fascism bullied believe terminated
827
828    get much obviously shopping yep go store
829    isnt interesting brains gets decide biden ukraine scandal limits yet
830
831    saved lives people soldiers
832    joe biden let china get away cheating vice president continues naively dismiss china today
833    want watch dems implode solve homeless problem closed bases housing barracks mess ha
834    dirty democrats demanding american taxpayers allow illegal aliens pour co
835    one winning trump today lost billions saved wealth th
836    bernie sanders come clean fess million campaign donations stick
837    hard fathom struggle get people want work white house
838    good morning patriots fellow vets everyday get eternally grateful corrupt politician self serv
839
840    yass queen beautiful lady woke af
```

Obr. 2.5: Ukázka textu po zpracování.

[15]. Vstupní data pro tuto funkci jsou předpřipravené texty uložené v DataFramu.

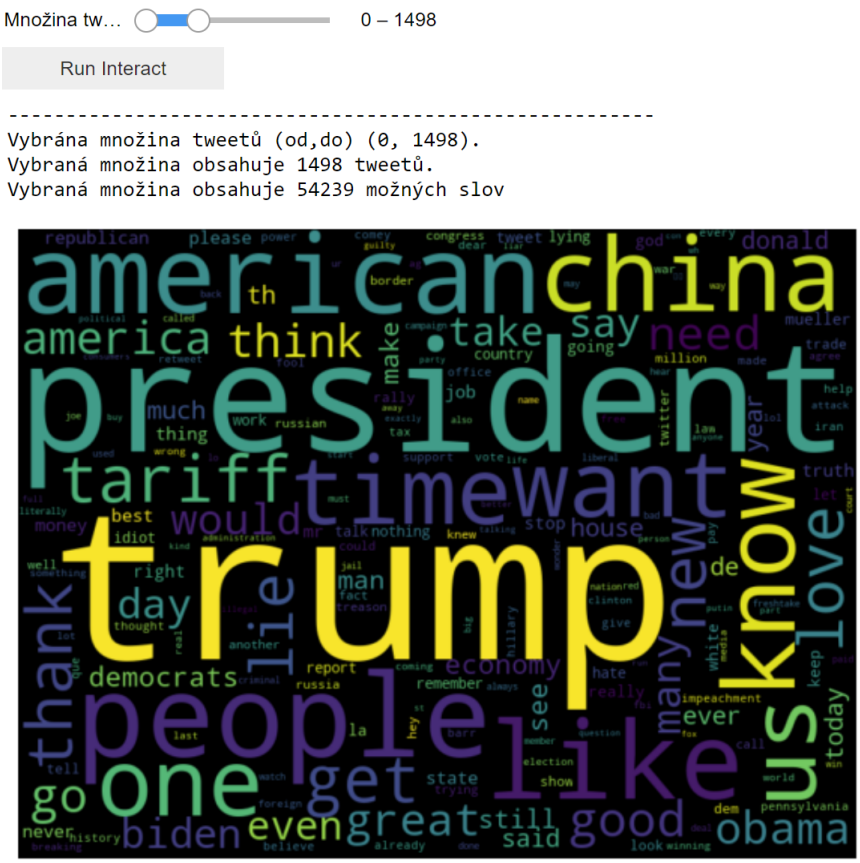
Interaktivní prostředí

Výsledný vygenerovaný obrázek 2.6 zobrazuje výstup z aplikace, kde uživatel pomocí posuvníku vybere množinu tweetů, které jsou pak vizualizovány. Včetně obrázku jsou k dispozici informace, kolik vybraná množina obsahuje tweetů a slov.

Rozpoznávání názvů entit v textu

Knihovna spaCy[16], která byla použita pro demonstraci rozpoznávání názvů entit v textu je speciálně navržena pro produkční verzi a pomáhá vytvářet aplikace, které pracují s velkými objemy textu. Jedná se o bezplatnou knihovnu pro pokročilé metody NLP s řadou funkcí včetně NER. Knihovna je napsaná v programovacím jazyku Cythonu[17]. Pojmenovaná entita je objekt reálného světa, například osoba, země, produkt, nebo název knihy. SpaCy dokáže rozpoznat různé typy entit, viz stránky[18] v dokumentu tím, že požádá model o predikci. Vzhledem k tomu, že modely jsou statistické a silně závisejí na příkladech, na kterých byly trénované, nemusí vždy tato funkce pracovat dokonale a případně potřebuje doladit. Pro analýzu byl zvolen *en_core_web_md* natrénovaný model na OntoNotes 5[19]. Knihovna umožňuje i vlastní trénování entit. Vlastní trénování entit nebylo implementováno.

[15] http://amueller.github.io/word_cloud/
[16] https://spacy.io/
[17] https://cython.org/
[18] https://spacy.io/api/annotation#named-entities
[19] https://catalog.ldc.upenn.edu/LDC2013T19



Obr. 2.6: Ukázka vizualizace slov - knihovna wordcloud.

Interaktivní prostředí

Po stisku tlačítka Run Interact jsou k dispozici výsledky, kde je vizualizován přímo text, nalezené entity jsou obarveny podle typu.

Vnoření slov

Vstupní data pro vytvoření instance Word2Vec, jsou uložena jako seznam v seznamu. Každý seznam obsahuje sadu tokenů (slov). Word2Vec použije všechny tyto tokeny k internímu vytvoření slovníku. Nastavení hlavních parametrů:

size - velikost hustého vektoru reprezentující každý token (slovo). Pokud jsou k dispozici jen omezené data, pak by velikost parametru měla být mnohem menší, protože pro dané slovo by existovalo až příliš jedinečných sousedů. Pokud je naopak k dispozici více dat, je vhodné experimentovat s různými velikostmi, například hodnotami mezi sto až stopadesát.

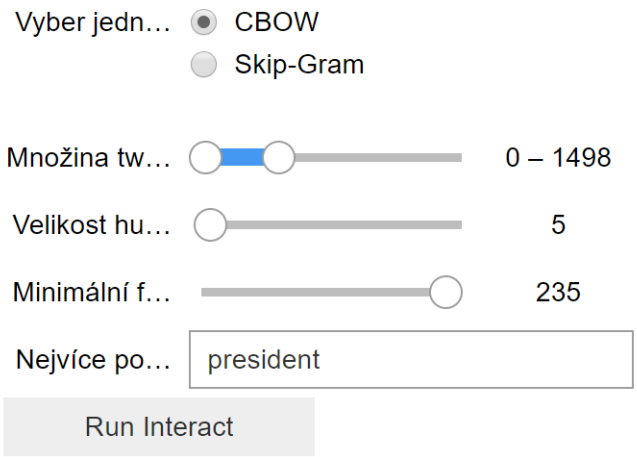
min_count - minimální frekvence počtu slov. Model ignoruje slova, která nesplňují daný počet. Slova, které se v dokumentu vyskytují málo jsou obvykle nedůležité,

proto nejsou zahrnuty do modelu.

Po takto sestavené slovní zásobě se použije funkce pro trénování modelu.

Interaktivní prostředí

Obrázek 2.7 zobrazuje základní prvky GUI. Uživatel má možnost vybrat si ze dvou modelů CBOW a Skip-gramu, dále je možnost vybrat množinu tweetů, která je přednastavena na všechny tweety uložené v souboru. Lze nastavovat dva zmíněné parametry, (*size* a *min_count*). V aplikaci jsou pojmenované tyto parametry jako velikost hustoty vektoru a minimální frekvence slov. Posledním prvkem je textové pole, které hledá podobnost mezi zadaným slovem se slovami v natrénovaném modelu. Po stisknutí tlačítka Run Interact jsou k dispozici výsledky formou grafu. Pod grafem se nachází výpis slovníku, který byl použit a slova, která jsou vektorově nejblíž slovu zadanému v textovém poli, viz výsledky.



Obr. 2.7: Ukázka interaktivního GUI - vnořování témat.

Modelování témat

Pro vytvoření funkce modelování témat byla použita knihovna Gensim s metodou LDA. Dva hlavní vstupy do LDA modelu témat jsou slovník a korpus. Knihovna Gensim vytvoří unikátní id pro každé slovo v dokumentu. Vyrobený korpus má tvar (id slova, frekvence slova), viz obrázek 2.8.

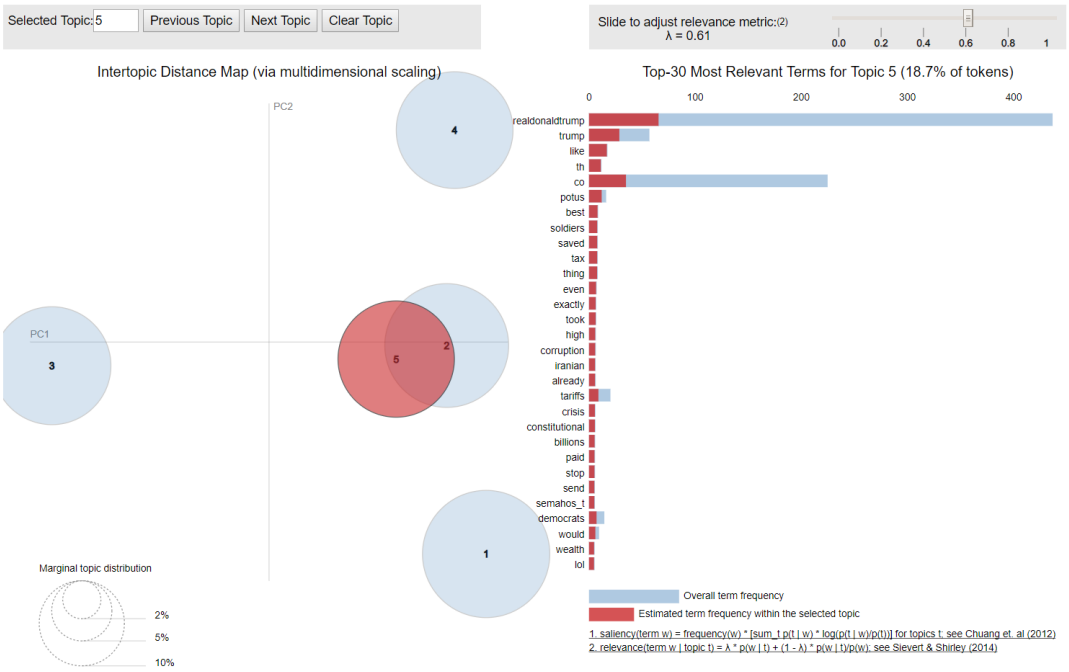
Například tvar (0,1) znamená, že slovo s unikátním id nula se vyskytuje pouze jednou v celém dokumentu. Včetně korpusu a slovníku je potřeba při vytvoření modelu uvést počet témat, která mají být extrahována z treninkového korpusu. Další možné parametry jako jsou alfa a eta, které ovlivňují řídkost témat jsou nastaveny na

```
Slovník: [['david_leavitt', 'today', 'would', 'great', 'day', 'realdonaldtrump',
--------------------------------------------------------------------------------
Korpus: [[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1),
--------------------------------------------------------------------------------
Čitelný formát: [[('david_leavitt', 1), ('day', 1), ('fuck', 1), ('giuliani', 1),
1), ('today', 1), ('would', 1)]]
--------------------------------------------------------------------------------
```

Obr. 2.8: Ukázka slovníku a korpusu včetně čitelného zobrazení korpusu.

hodnotu auto (hodnota auto znamená u alfy, že se naučí asymetricky před korpusem, u ety před daty). Parametr chunksize je počet dokumentů, které mají být použity v každém treninkovém bloku. Parametr update_every určuje, jak často mají být parametry modelu aktualizovány celkovým počtem treninkových průchodů.

Poté co je vytvořen LDA model, je dalším krokem zkoumání vytvořených témat a souvisejících klíčových slov. Pro vizualizaci této metody byla použita knihovna pyLDAvis, která dobře funguje s Jupyter Notebook. Každý kruh v grafu představuje jedno téma 2.9. Čím je kruh větší, tím je toto téma častější. Dobrý model bude mít poměrně velké, nepřekrývající se kruhy, rozptýlené po celém grafu, namísto toho aby byly seskupeny v jednom kvadrantu. Po přesunutí kurzoru nad konkrétní kruh, se zobrazí slova použitá v daném tématu s četností v dokumentu. Tato slova jsou klíčová a tvoří vybrané téma.



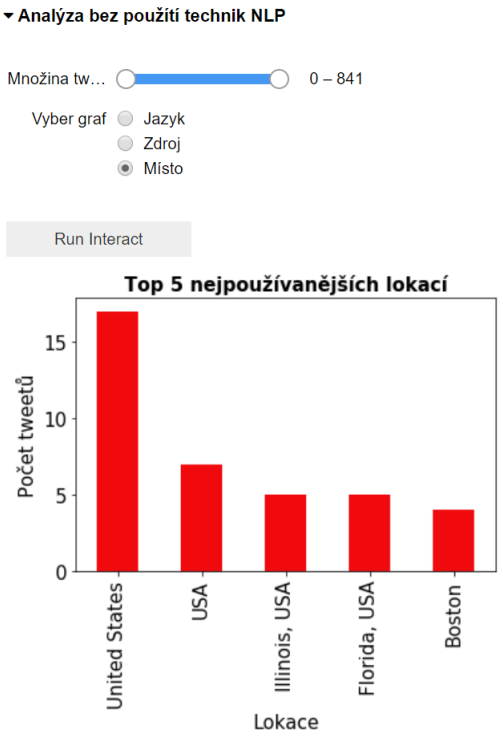
Obr. 2.9: Ukázka vizualizace modelování témat - knihovna pyLDAvis.

Analýza dat bez použití NLP

Jedná se o základní analýzu bez použití technik NLP. Pro analýzu jsou potřeba vyfiltrovaná data, uložená v DataFramu, pomocí kterého se spočítá množství každé unikátní položky a poté pomocí knihovny matplotlib jsou data zobrazovány v grafu. Výjimku tvoří například položka zdroj (v JSONu jako parametr source), který je vložen do HTML tagů a je potřeba tyto hodnoty vyextrahovat pomocí knihovny BeautifulSoup.

Interaktivní prostředí

Po kliknutí na tlačítko Run Interact je k dispozici výsledek formou grafu, viz obrázek 2.10. Uživatel má možnost vybrat množinu tweetů, které chce zahrnout do grafu a výběr zobrazovaných dat. Pro demonstraci byly implementovány pouze tři možnosti.



Obr. 2.10: Ukázka interaktivního GUI - analýza bez použití NLP.

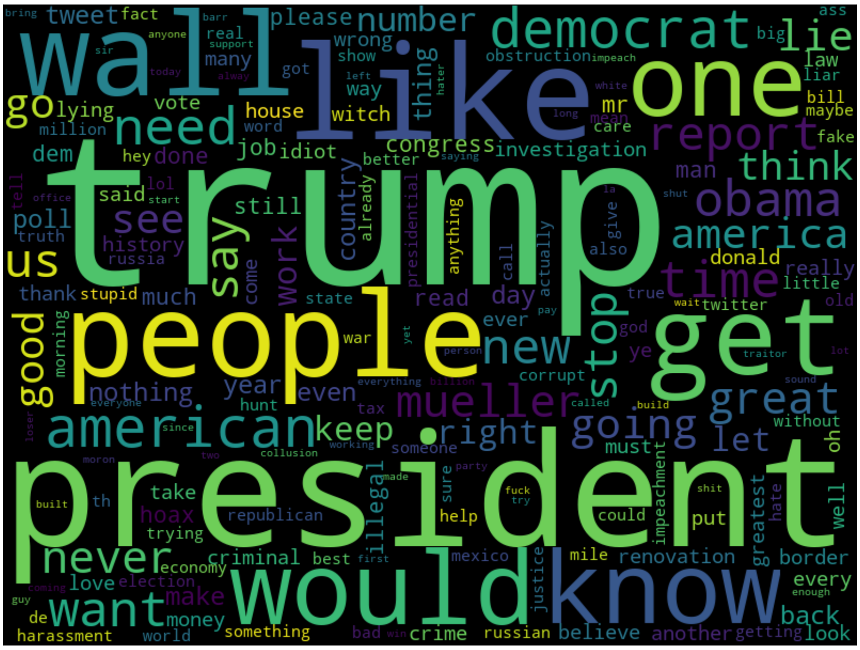
3 Výsledky

V této kapitole jsou demonstrovány výsledky aplikace. **Zpracovány a vizualizovány jsou zde skutečné texty stažené ze sociální sítě Twitter, které mohou obsahovat jak pravopisné chyby tak i vulgární slovní spojení.** Při nahrávání jakéhokoliv souboru do Jupiter Notebook je velikost souboru nastavena na výchozí hodnotu 25 MB, což přibližně odpovídá 3800 tweetům. Proto při analýze většího množství tweetů, kdy byl nahrán největší soubor o velikosti 377 MB byl webový prohlížeč otevřen přes grafický procesor NVIDIA[1] s vysokým výkonem.

Všechny následující ukázky byly vytvořeny s celkovým počtem 63 421 tweetů, které obsahovaly 1 314 411 možných slov. Stahování proběhlo dne 22. května 2019 v čase od 14:00 do 15:00.

3.1 Word Cloud

Výsledek pomocí metody „word cloudu“ zachycuje obrázek 3.1. Analyzovány byly tweety, které byly zaměřeny na prezidenta Donalda Trumpa. Nejčastěji se vyskytovaly slova „trump, president, like, people“.



Obr. 3.1: Výsledek metody word cloud.

[1]`https://www.nvidia.com/`

3.2 Sumarizace textu

Výsledek pomocí metody sumarizace znázorňuje obrázek 3.2, jedná se pouze o část výsledku.

```
Summary:
truly amazing president trump focused winning democrats still trying win.
wednesday may one citizen call resignation impeachment donald trump president.
going meet president trump tomorrow meeting democrats trump impeachment think.
looks like president great standing china holding responsible trade.
congressional democrats want work american people busy covering cor.
right time blow phone nonstop guilty tweets good job impeachment.
maybe let see unredacted report never happened obama corrupt president.
democrats interested putting illegals first president trump puts americans first.
hey jerk shut talk president united states like sorry ur brainwashed.
thanks president wall built quickly president trump fixing democrats.
great people president lied times since taking office.
president quite skilled telling americans think great juvenile name calling.
people like americans love nation black trump supporter.
tweets harassment good hard working american people.
point democrats like schiff nadler focus working american people attacking.
exactly type president leaders countries want morons manipulate rob americans blind.
time president trump invoke insurrection act use us military deport.
poll numbers notoriously wrong especially comes trump america president trump.
```

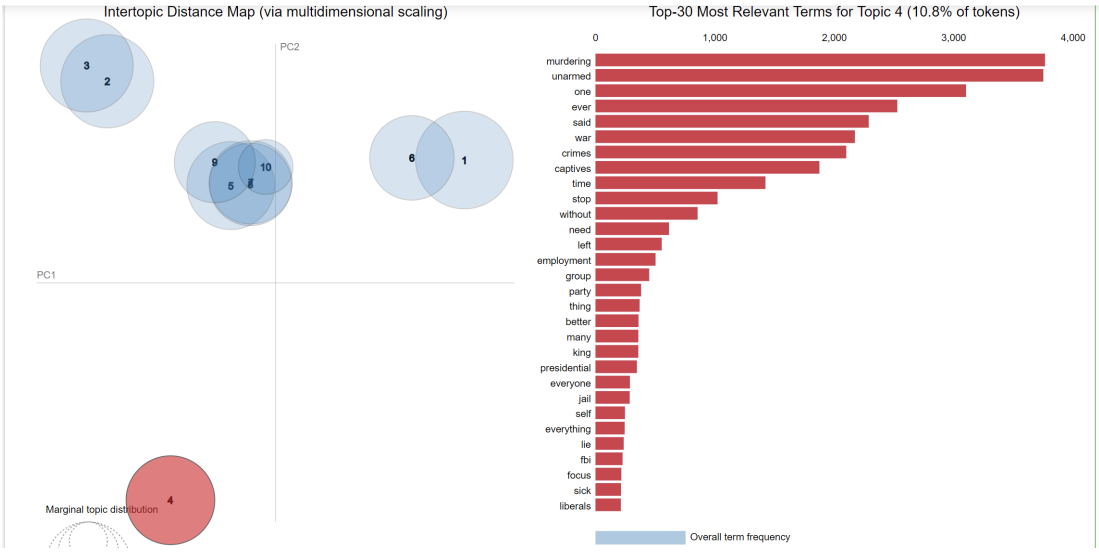
Obr. 3.2: Výsledek metody sumarizace textu.

3.3 Modelování témat

Výsledek pomocí metody modelování témat zachycuje obrázek 3.3. Nejvhodnější soudržnost tématu byla při vygenerování celkem 10 témat. Obrázek 3.4 znázorňuje vizualizaci vygenerovaného modelu, označeno je čtvrté téma.

```
Témata (výpis pouze prvních 5 slov):
0: 0.066*"support" + 0.037*"america" + 0.028*"house" + 0.027*"law" + 0.024*"deal"

1: 0.033*"use" + 0.032*"coming" + 0.027*"office" + 0.022*"rat" + 0.021*"texas"

2: 0.100*"trump" + 0.096*"president" + 0.067*"hey" + 0.036*"great" + 0.023*"even"

3: 0.085*"murdering" + 0.085*"unarmed" + 0.070*"one" + 0.057*"ever" + 0.052*"said"

4: 0.069*"girl" + 0.038*"illegal" + 0.035*"china" + 0.030*"numbers" + 0.027*"investigation"

5: 0.038*"wall" + 0.035*"want" + 0.034*"get" + 0.029*"take" + 0.020*"obama"

6: 0.084*"like" + 0.056*"people" + 0.033*"would" + 0.028*"american" + 0.027*"democrats"

7: 0.056*"mueller" + 0.055*"report" + 0.048*"justice" + 0.044*"lot" + 0.044*"collusion"

8: 0.029*"know" + 0.023*"response" + 0.021*"please" + 0.019*"yes" + 0.017*"make"

9: 0.040*"biden" + 0.037*"us" + 0.037*"going" + 0.030*"poll" + 0.029*"still"

Modelová složitost: -8.659813455391957
Soudržnost tématu: 0.40704727404341334
```

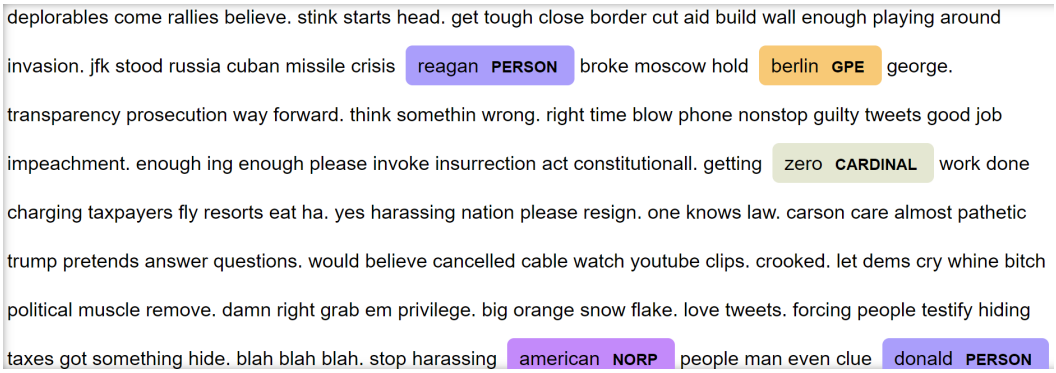
Obr. 3.3: Výsledek metody modelování témat.



Obr. 3.4: Výsledek metody modelování témat - vizualizace.

3.4 Rozpoznávání názvů entit v textu

Výsledek pomocí metody rozpoznávání entit v textu zachycuje obrázek 3.5. Knihovna spaCy [2] obsahuje modely pro tagování, analýzu a rozpoznávání entit. Jeho výhodou je, že disponuje doplňky, které umožňují vizualizovat daný text. Na přiloženém obrázku si můžeme všimnout části označených slov, které tyto natrénované modely dokázaly rozpoznat.



Obr. 3.5: Výsledek metody rozpoznávání názvů entit v textu.

[2] https://spacy.io/

3.5 Klíčová slova

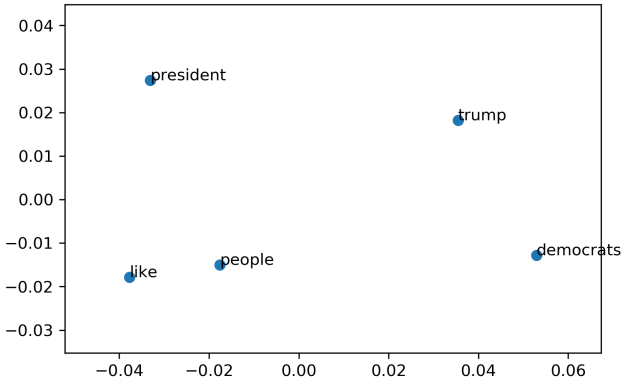
Výsledek pomocí metody klíčových slov zachycuje obrázek 3.6. Slova jsou seřazena sestupně podle hodnocení algoritmu. Lze vidět opakující se slova (tak, jak tomu bylo například u metody word cloud), která se ve tweetech objevují nejčastěji, například slova „trump, president, like, people“.

```
Keywords:
['trump', 'trumps', 'trumped', 'trumping', 'president', 'president
s', 'presidency', 'preside', 'presidence', 'like', 'likes', 'likel
y', 'liking', 'liked', 'people', 'peoples', 'peopl', 'knows', 'kno
wing', 'know needs', 'need', 'needed', 'needing', 'neede', 'democr
at', 'democratically', 'democratics', 'walls', 'walling', 'walle
d', 'presid presidential', 'want', 'wants', 'wanted', 'wanting',
'time', 'timing', 'timeli', 'obamas', 'think', 'thinks', 'new', 's
top', 'stopped', 'stops', 'stopping', 'tweets', 'tweet', 'tweetin
g', 'tweeted', 'democratic house', 'democrats got', 'american', 'a
mericans', 'americanes', 'americas', 'going', 'good', 'goodness',
'goods', 'let', 'letting', 'right', 'rights', 'rightful', 'country
```

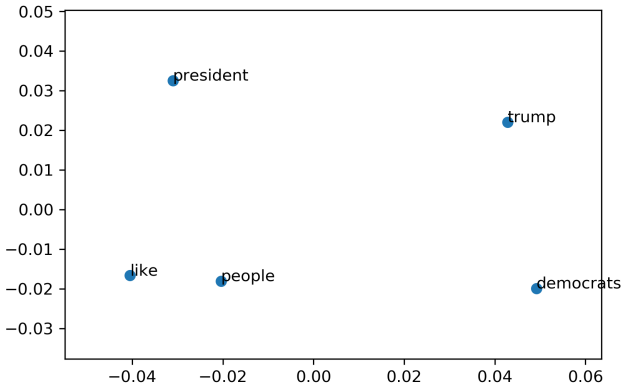
Obr. 3.6: Výsledek metody - keywords.

3.6 Vnořování slov

Výsledky pomocí metody vnořování slov zobrazují následující dva obrázky. Obrázek 3.7, vygenerovaný model typu CBOW. Obrázek 3.8 vygenerovaný model typu Skip-gram. V obou případech jsou nastaveny stejné parametry, aby bylo možné porovnat výsledky mezi sebou. Grafy znázorňují vektorovou vzdálenost jednotlivých slov v 2D prostoru.



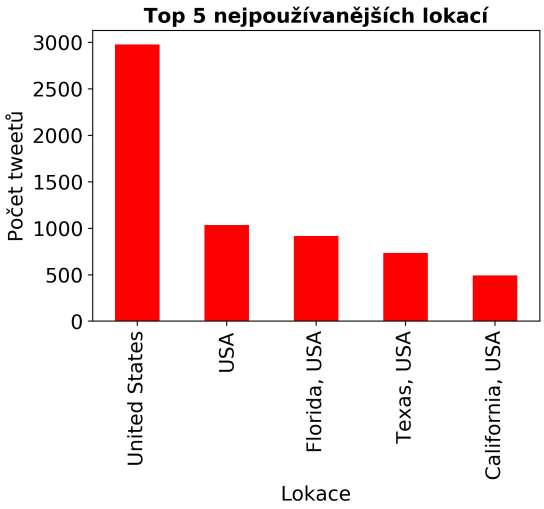
Obr. 3.7: Výsledek metody vnořování slov - CBOW.



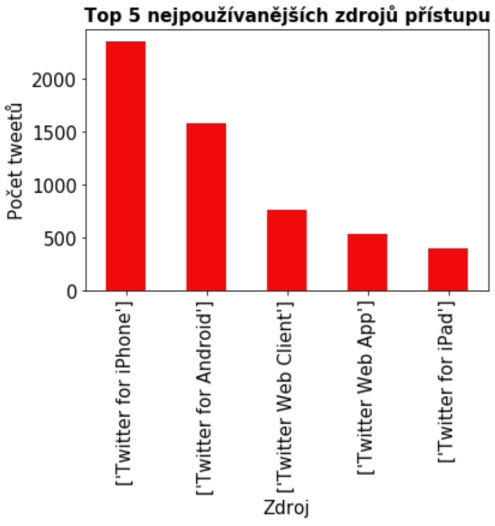
Obr. 3.8: Výsledek metody vnořování slov - Skip-gram.

3.7 Analýza dat bez použití NLP

Výsledky pomocí metod, které nevyužívali NLP zachycují následující dva obrázky. Obrázek 3.9, na kterém můžeme vidět lokaci, odkud bylo posláno nejvíce tweetů a obrázek 3.10, z jakého zařízení byly odeslány.



Obr. 3.9: Výsledek bez použití metod NLP - lokace.



Obr. 3.10: Výsledek bez použití metod NLP - zdroj.

4 Závěr

Cílem mé bakalářské práce bylo navrhnout, implementovat a otestovat aplikaci, která umožňuje automatické stahování a analýzu dat z Twitteru, založenou na technikách zpracování přirozeného jazyka. Aplikace je vytvořena v programovacím jazyku Python, obsahuje grafické a interaktivní rozhraní, které bylo vytvořeno ve výpočetním prostředí Jupiter Notebook. Jedná se o aplikaci, která je spuštěna ve webovém prohlížeči. Disponuje velkou komunitou uživatelů, vývojářů a podporuje mnoho programovacích jazyků včetně Pythonu. Vhodně kombinuje kód, vkládání textů, výsledky výpočtů, vizualizaci a to vše v jednom formátu.

Vytvořenou aplikaci nebylo nutné předem navrhovat. Hlavní okno GUI se skládá ze záložek, které oddělují aplikaci na logické celky. Mimo jiné aplikace umožňuje ukládání vygenerovaných grafů, které byly použity v této bakalářské práci v kapitole výsledky.

Pro stahování dat byl použit soukromý uživatelský účet amerického prezidenta Donalda Trumpa (@realDonaldTrump), který je pro svou popularitu ideálním zdrojem pro analýzu dat. Tweety byly odchytávány v reálném čase pomocí knihovny Tweepy, ta umožňuje manipulaci se streamovanými daty. Aplikace je schopná stáhnout za dobu 10 sekund průměrně až 100 tweetů od různých uživatelů, kteří ve svém textu zmiňují Donalda Trumpa „@realDonaldTrump“. Rychlost stahování se odvíjí od rychlosti připojení k internetu a množství odeslaných zpráv uživatelů. Odchycené tweety se ukládají do textového souboru v JSON formátu. Jeden řádek v textovém souboru odpovídá jednomu tweetu tak, jak byl zachycen ze streamu. Během celé doby, kdy byly tweety odchytávány nedošlo k žádným komplikacím, ať už ze strany Twitter API v podobě překročení limitů, nebo ze strany aplikace v podobě neočekávané chyby. Avšak při nahrávání jakéhokoliv souboru do Jupiter Notebook, je nutno podotknout, že velikost souboru by neměla překročit 25 MB, což přibližně odpovídá 3800 tweetům. Toto číslo se však může lišit podle atributů, které jsou k tweetům připnuté.

Při nahrávání tweetů ze souboru docházelo k občasným chybám. Tyto chyby vznikaly při ukládání dat, kdy systém nestíhal data včas zapisovat a docházelo tak k ukládání nekonzistentních tweetů. Tato chyba byla později vyřešena přidáním časovače nastaveného na padesát milisekund.

Do analýzy bylo zahrnuto 63 421 tweetů s celkovým počtem 1 314 411 možných slov. Byly implementovány tyto analýzy: sumarizace textu, modelování témat, vnoření slov, klíčová slova, rozpoznávání názvů entit v textu a word cloud. Při analýze typu word cloud a analýze klíčových slov v textu vycházela nejčastěji slova: „trump, president, like, people“. Zajímavostí je, že kombinace slov se neustále opakovala v průběhu celého půlroku, kdy byla aplikace testována. Analýza pokročilých tech-

nik typu modelování témat a vnořování slov, vyžadovala hledání vhodných vstupních proměných. Ze subjektivního hlediska lze říci, že výsledky vykazují dobrou kvalitu, avšak bez znalostí souvislostí, například americké politiky, nelze s jistotou tvrdit o jak kvalitní výsledky se jedná.

Implementace aplikace zahrnovala práci jak se standardními, tak i s nestandardními knihovnami Python, umožňujícími například práci s předpřípravami textu, stahováním dat z Twitteru pomocí API, nebo zpracování přirozeného jazyka. Nejtěžší část z celého projektu byla nepochybně předpříprava a následná analýza dat, zvláště pak samotného textu, ve kterém se vyskytovaly nejednoznačnosti díky uživatelům, kteří se někdy neřídili pravidly gramatiky.

V budoucnu by mohla být tato práce zaměřena na vytvoření internetového bota, který by automatizovaně stahoval tweety od konkrétního jedince a potřebná data ukládal do vhodně navrhnuté databáze. Další fází by poté bylo vytvoření webové aplikace, například v Django frameworku, který by umožňoval uživatelům analyzovat data, zejména pak analýzu pomocí knihovny Gensim.

Literatura

[1] Dimitrios Milioris (auth.). *Topic Detection and Classification in Social Networks: The Twitter Case*. Springer International Publishing, 1 edition, 2018. URL: http://gen.lib.rus.ec/book/index.php?md5=0bf66e6e3f8c76614cb8af66ab2da7ab.

[2] Michal Krystyanczuk Siddhartha Chatterjee. *Python Social Media Analytics: Analyze and visualize data from Twitter, Youtube, GitHub, and more*. Packt Publishing, 2017. URL: http://gen.lib.rus.ec/book/index.php?md5=2a29a0ac497d9802749706cbb7e57471.

[3] Inc Twitter. online documentation, 2019. URL: https://developer.twitter.com/en/docs#.

[4] Jalaj Thanaki. *Python Natural Language Processing*. Packt Publishing, 2017. URL: http://gen.lib.rus.ec/book/index.php?md5=4FC7370281AE626D07D52F79A7857D92.

[5] Zixin Luo. *Text Summarization using Natural Language Processing*. PhD thesis, Worcester Polytechnic Institute, 2018.

[6] James Allen. *Natural language understanding*. Benjamin/Cummings series in computer science. Benjamin/Cummings Pub. Co, 1987. URL: http://gen.lib.rus.ec/book/index.php?md5=39D7E12F880DCD0FD145ED361D32E7CE.

[7] Evans R. Belz A. *Natural Language Generation*. Springer - 2011-04-02, 2004. URL: http://gen.lib.rus.ec/book/index.php?md5=06C41B2DCA2F3EE79C7FFE6DDC35AA24.

[8] Karan Jain Palash Goyal, Sumit Pandey. *Deep Learning for Natural Language Processing. Creating Neural Networks with Python*. Apress, 2018. URL: http://gen.lib.rus.ec/book/index.php?md5=13487ef9cc76ac61700a0a36c358bd95.

[9] Yang Liu Li Deng. *Deep Learning in Natural Language Processing*. Springer, 2018. URL: http://gen.lib.rus.ec/book/index.php?md5=3de0be1b9217691c2022af5c1ed296ca.

[10] Iti Mathur Deepti Chopra, Nisheeth Joshi. *Mastering Natural Language Processing with Python*. Packt Publishing, 2016. URL: http://gen.lib.rus.ec/book/index.php?md5=b30895263b21846d989498722569beab.

[11] Taweh Beysolow II. *Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*. Apress, paperback edition, 2018. URL: http://gen.lib.rus.ec/book/index.php?md5=5baa54d2d27feca6ab4e10644a14599c.

[12] Bhargav Srinivasa-Desikan [Bhargav Srinivasa-Desikan]. *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis With Python, Gensim, spaCy, and Keras*. Packt Publishing, 2018. URL: http://gen.lib.rus.ec/book/index.php?md5=531D6EA77DE7AEA459D785A196733210.

[13] Deepti Chopra Nitin Hardeniya, Jacob Perkins. *Natural Language Processing: Python and NLTK*. Packt Publishing, 2016. URL: http://gen.lib.rus.ec/book/index.php?md5=f557b0532ba9a94d25f72d73724e8187.

[14] Rajalingappaa Arumugam, Rajesh; Shanmugamani. *Hands-on natural language processing with Python : a practical guide to applying deep learning architectures to your NLP applications*. Packt Publishing, 2018. URL: http://gen.lib.rus.ec/book/index.php?md5=f5c37c265ab1bad3b4aff94d5d258f90.

[15] Sumit Pandey Palash Goyal. *Deep Learning for Natural Language Processing: Creating Neural Networks with Python*. Apress, 1 edition, 2018. URL: http://gen.lib.rus.ec/book/index.php?md5=2f7a5655f311967e9ddc6fcbdb969c10.

[16] Kjell Magne Fauske. Example neural network in tikz, 2018. URL: http://www.texample.net/tikz/examples/neural-network/.

Seznam symbolů, veličin a zkratek

AI	umělá inteligence – Artificial intelligence
API	architektura rozhraní pro programování aplikací – Application Programming Interface
CBOW	Continuous Bag of Words
DSP	číslicové zpracování signálů – Digital Signal Processing
LDA	latentní Dirichletova alokace – latent Dirichlet allocation
LSI	latentní sématické indexování – latent sematic indexing
NL	přirozený jazyk – Natural Language
NLP	zpracování přirozeného jazyka – Natural Language Processing
NLG	generování přirozeného jazyka – Natural Language Generation
NLTK	sada knihoven pro zpracování přirozeného jazyka – Natural Language Toolkit
NLU	pochopení přirozeného jazyka – Natural Language Understanding
REST API	architektura rozhraní pro distribuované prostředí aplikací – REpresentational State Transfer
TF	četnost slova v dokumentu – Term Frequency
TF–IDF	převrácená četnost slova ve všech dokumetech – Inverse document Frequency
URL	jednotná adresa zdroje – Uniform Resource Locator

Seznam příloh

A Obsah CD

K bakalářské práci je přiloženo CD s touto adresářovou strukturou:

- soubor xrydlp00.pdf - Tento dokument,
- soubor README.txt - Stručný návod k použití aplikace,
- soubor aplikace.ipynb - Jupiter Notebook s aplikací.
- soubor requirements.txt - Soubor obsahuje verze knihoven.

B Seznam použitých knihoven

Knihovna použitá pro stahování tweetů:

- Tweepy [1].

Knihovny použité pro zpracování a čištění textu:

- json [2],
- html [3],
- re [4],
- BeautifulSoup [5],

Knihovna pro uložení dat:

- Pandas [6].

Knihovny pro zpracování přirozeného jazyka:

- Gensim [7],
- NLTK [8],
- spaCy [9].

Knihovny pro vizualizaci dat:

- matplotlib [10],
- wordcloud [11].
- pyLDAvis [12].
- IPython [13].

Knihovny pro GUI:

- ipywidgets [14],

[1] http://www.tweepy.org/
[2] https://docs.python.org/2/library/json.html
[3] https://docs.python.org/3/library/html.parser.html
[4] https://docs.python.org/3/library/re.html
[5] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[6] https://pandas.pydata.org/
[7] https://github.com/RaRe-Technologies/gensim
[8] https://www.nltk.org/
[9] https://spacy.io/
[10] https://matplotlib.org/
[11] https://github.com/amueller/word_cloud
[12] https://pyldavis.readthedocs.io/en/latest/index.html
[13] https://ipython.org/
[14] https://ipywidgets.readthedocs.io/en/stable/index.html

C Příklad struktury tweetu

Výpis C.1: Příklad strukruty tweetu v JSON.

```
{
    "created_at": "Sat␣Nov␣17␣12:35:41␣+0000␣2018",
    "text": "@realDonaldTrump␣Against␣all␣odds?␣That\u2019s␣
       ridiculous",
    "display_text_range": [
        17,
        53
    ],
    "source": "",
    "in_reply_to_screen_name": "realDonaldTrump",
    "user": {
        "id": ,
        "id_str": "",
        "name": "",
        "screen_name": "",
        ...
    },
    "geo": null,
    "coordinates": null,
    "place": null,
    "contributors": null,
    "is_quote_status": false,
    "quote_count": 0,
    "reply_count": 0,
    "retweet_count": 0,
    "favorite_count": 0,
    "entities": {
        "hashtags": [],
        "urls": [],
        "user_mentions": [
            {
                "screen_name": "realDonaldTrump",
                "name": "Donald␣J.␣Trump",
                "id": 25073877,
                ...
            }
        ],
        "symbols": []
    },
     ...
    "lang": "en",
    "timestamp_ms": "1542458141281"
}
```