Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Sledování trendů na sociální síti Twitter

Plzeň 2015 Bc. Václav Rajtmajer

Originál zadání diplomové práce.

Poděkování

Děkuji vedoucímu diplomové práce Ing. Pavlu Královi, Ph.D. za cenné rady, motivaci a vstřícný přístup.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 18. května 2015

Václav Rajtmajer

Abstract

This master thesis deals with the social network Twitter and a general event detection in real time. The main goal is to create a system for Czech News Agency (ČTK) which will be able to monitor the current data-flow on Twitter, analyze it and extract relevant events. The newly detected events then will be presented to users in an acceptable form. It was thus created a novel original experimental event detection system. Some experiments have been realized in order to find and define optimal system parametrs and to show the performance of the system. The reported precission was 50% and recall was 44,4% which is very interesting for the ČTK.

Abstrakt

Tato diplomová práce se zabývá sociální sítí Twitter a obecnou detekcí událostí v reálném čase. Hlavním cílem práce bylo vytvořit systém pro Českou tiskovou kancelář, který bude za pomoci API knihovny sledovat dění na Twitteru, tyto texty dále analyzovat a extrahovat z nich události. Nově detekované události pak bude přijatelnou formou prezentovat uživateli. Podle těchto požadavků vznikl nový systém detekující události na Twitteru, který byl dále podroben několika experimentům k určení optimálních parametrů a k demostraci vlastností programu. Výsledná přesnost byla 50% a úplnost 44,4%, což je pro ČTK přijatelné.

Obsah

Seznam obrázků

1 Úvod

V posledních letech internetová společnost vzrůstá do nebývalých rozměrů. Stovky milionů uživatelů jsou aktivní na sociálních sítích, fórech a nebo přispívají do některých z blogovacích služeb. Trendem dnešní doby je využívat služby sociálních sítí nejen k interakci se svými přáteli, ale také ke sdílení informací, názorů, myšlenek a pocitů.

Pro nás je zajímavý především Twitter[1], který je se svými miliony uživatelů v současné době leaderem mezi mikroblogujícími systémy. Uživatelé Twitteru generují krátké zprávy - tzv. tweety - a sdílejí tak informace se svým okolím. Každý uživatel vlastní profil, ve kterém můžeme najít některé osobní údaje, jako je jméno, lokaci, ale i zájmy a osobní historii. Tweet je pak asociován s jedním profilem, časovou značkou a někdy i geotagem - místem určeným GPS souřadnicemi.

Jednoduchý přístup a široké spektrum uživatelů ale vede k tomu, že takto sdílené informace obsahuje i široké spektrum témat. Můžeme zde najít bezpředmětné soukromé rozhovory, reklamu, ale i čerstvé novinky ze sportu a informace o právě probíhajícím zemětřesení.

Tento stále aktuální proud informací nabízí široké možnosti k analýzám a dataminingu. V našem případě se budeme pokoušet o automatickou detekci nově vznikajících událostí a trendů v reálném čase, které uživatelé Twitteru právě sledují. Tento zdroj by měl sloužit jako potenciální směr zájmu pro zpravodajské služby a měl by tedy zachycovat pouze trendy, které se týkají aktuálních témat ve společnosti.

V dnešní době zpravodajské služby zprostředkovávají informace od médií k lidem. Náš systém je tedy do určité míry snahou o reverzi tohoto procesu a při úspěšném splnění zadání budeme mít nástroj, který bude přenášet informace směrem od lidí k médiím. Ten přinese novou metodu získávání informací pro zpravodajské služby, která v některých případech bude rychlejší než konvenční způsoby. Tato práce byla zadána v rámci smluvního výzkumu pro Českou tiskovou kancelář.

Práce je rozdělena na teoretickou a praktickou část. V následujících kapitolách teoretické části se budeme zabývat sociální sítí Twitter, její historií, ob-

[1] www.twitter.com

sahem a možnostmi. Krátce si představíme knihovnu funkcí API a možnosti automatického čtení tweetů. Dále popíšeme nástroje pro práci se získanými daty, která budeme převádět do základních tvarů procesem lemmatizace. Předně se ale budeme soustředit na proces shlukování, který se používá pro seskupení podobných objektů do skupin tzv. clusterů. Tyto skupiny představují hledané trendy a musíme pro ně tedy zvolit vhodný název a ohodnocení relevantnosti.

V praktické části navrhneme způsoby pro získání dat a porovnáme jejich potenciály. Pomocí vybraného kanálu a Twitter API pak získáme tweety, které pomocí dříve popsaných metod dále zpracujeme do clusterů, ohodnotíme jejich důležitost a navrhneme způsob jejich reprezentace. V další kapitole podrobíme takto vzniklý systém několika experimentům, kterými nalezneme takové nastavení vstupních parametrů, které nám poskytne nejlepší výsledky.

Na závěr zjistíme přesnost a úplnost navrhnutého řešení a tyto výsledky zhodnotíme.

2 Seznámení se sociální sítí Twitter

V této kapitole se seznámíme se sociální sítí Twitter, jejími uživateli a obsahem. Dále se zaměříme na veřejnou knihovnu API, kterou Twitter poskytuje pro strojový přístup k datům a její možnosti.

Twitter vznikl v roce 2006 a od té doby jeho popularita jen roste. Myšlenka sdílení krátkých zpráv - tweetů - zažila nečekanou vlnu nadšení a dnes již není nic překvapujícího, že za jeden den uživatelé odešlou i 500 milionů tweetů [liv(2015)]. Během fenomenálního růstu sociálních médií se Twitter stal jedním z největších mikro-blogů na světě a jeho popularita i nadále roste.

Dnes má svůj účet nespočet politiků, celebrit a organizací. I díky této popularitě se mohla široká veřejnost například dozvědět, že vesmírná sonda Mars Phoenix našla v roce 2008 na Marsu led. Nebo že prezident Obama vyhrál v roce 2012 volby a bude na další 4 roky prezidentem Spojených států amerických. Tyto novinky - a mnoho dalších - byly dostupné dříve na Twitteru, než je zveřejnily běžně dostupná média. [nin(2015)].

Podstata tohoto systému a masivní objem dat upoutává pozornost mnoha výzkumných pracovníků. Sakaki použil Twitter jako detektor směru a velikosti zemětřesení a díky jeho systému dokáže v reálném čase varovat uživatele o příchozím zemětřesení rychleji, než *Japan Meteorological Agency* [Sakaki et al.(2010)Sakaki, Okazaki,, Matsuo].

V roce 2014 vznikla ve Velké Británii organizace nazvaná *Samaritans Radar*[1], která analyzovala veřejné příspěvky a hledala mezi nimi vyjádření úzkosti, deprese či sebedestruktivní povahy. Po nalezení takového příspěvku pak informovala třetí osobu, která mohla zareagovat dle svého uvážení. Tím měl být snížen počet sebevražd. Bohužel se ale aplikace setkala s neúspěchem a byla ukončena devět dní po jejím spuštění. Podle časopisu EkonTech [Busta(2015)] byla hlavním důvodem ukončení služby malá skupinka lidí, která zneužívala veřejně dostupné informace k šikanování lidí právě v jejich úzkostlivých okamžicích. Na stránkách *BBC* [bbc(2014)] je uvedeným důvodem porušování práva na soukromí, což je spjaté s velkým počtem negativních ohlasů.

Z těchto příkladů je zřejmé, jak může být těch několik málo znaků důležitých.

[1] http://www.samaritans.org/

2.1 Vymezení pojmů

Pro seznámí s touto sociální sítí je třeba zavést některé pojmy, které jsou již zažité a těžko přeložitelné. Jde převážně o části Twitteru, se kterými budeme v následujících kapitolách pracovat.

Pojem	**Význam**
Tweet	Nebo i „Status“ = jednotka informace Twitteru. Jde o 140 znaků textu, který sdílí jednotliví uživatelé.
Follower	Uživatel, který sleduje náš profil. Je tedy s námi v nějaké relaci a odebírá proto naše tweety na vlastní časovou osu (viz. „Timeline“ níže).
Friend	V překladu „sledovaný“. Je to uživatel, kterého sledujeme.
Timeline	Každý uživatel má k dispozici časovou osu, na které se objevují jeho vlastní tweety, tweety jeho přátel a tweety, ve kterých je zmíněno jeho jméno.
UserList	Jde o seznam uživatelů. V každém profilu může být nastaveno 20 seznamů po 5 000 uživatelích. Každý z těchto seznamů má pak vlastní timeline.
Stream	Informační kanál nebo-li proud tweetů, který na jedné straně plní Twitter stále aktuálními tweety a na druhé straně je pak místo pro naši aplikaci, která může tato data číst a dále zpracovávat.
REST	*Representational State Transfer* je metoda pro komunikaci se serverem pomocí jednoduchých HTTP volání.
Screen name	Je strojová přezdívka uživatele, která musí být unikátní a je nejčastěji používána v URL adrese.
Hashtag	Je klíčové slovo (často kombinace několika slov či znaků bez mezer), začínající znakem křížku ('#'). Přidáním hashtagů do tweetu mohou uživatelé kategorizovat tweety do skupin.
@	Stejně jako hashtag uvozuje klíčové slovo, znak zavináče uvozuje tzv. „screen name“.
RT	Retweet. Twitter umist'uje tuto zkratku před text tweetu, pokud se jedná o retweet. Tedy o tweet, který přebíráme od někoho jiného a sdílíme ho dál pod vlastním jménem.

2.2 Tweet

Twitter je unikátní především omezenou délkou zpráv (statusů). V dnešní době, kdy je zvykem rušit a navyšovat všemožné limity a omezení, Twitter si stále drží povolenou délku tweetu 140 znaků. Původ této restrikce najdeme už v prvních návrzích Twitteru (tehdy se ještě jmenoval Twttr) [nin(2015)]. Tento systém byl primárně vyvíjen pro malou skupinu přátel, která si mohla posíláním SMS sdílet informace, co právě kdo dělá. Právě využití mobilních telefonů přinášelo v té době omezení 140 znaků. Systém se vyvinul do něčeho většího, avšak omezení zůstalo [rea(2013)].

Je zajímavé, že i přes toto omezení je Twitter tak oblíbený nástroj pro sdílení informací.

Zajímavou vlastností každého tweetu je možnost obsahovat hashtagy (viz. předchozí kapitola). Přidáním hashtagu do textu tweetu se stává hypertextovým odkazem a slučuje tak tento tweet s dalšími, které obsahují stejný hashtag. Pokud by tento princip pochopil každý uživatel a Twitter by neomezoval délku zprávy, mohl by být každý tweet otagovaný výčtem hashtagů. Takový stav by pak neposkytoval téměř žádný prostor pro náš výzkum.

2.3 Uživatelé Twitteru

Twitter není zaměřen na žádnou konkrétní skupinu uživatelů. Účet si může vytvořit školák, politik, kadeřnice i celá organizace. Stačí vyplnit jméno, e-mail a heslo a jsme součástí této sociální sítě. Proto není divu, že je uživatelská základna Twitteru tak široká - vždyt' není problém, vytvořit si účtů hned několik.

Větší problémy nastávají, pokud si někdo chce vytvořit populární - tedy sledovaný - účet. Ve světě se neustále svádí denodenní boje o každého followera (uživatel, který sleduje můj účet - tedy sleduje můj profil a odebírá mé příspěvky) a umístění v žebříčcích sledovanosti je věcí prestiže. Vzniká tak nové místo na trhu a s ním i nové společnosti, vydělávající na „prodeji followerů“[2]. Je pak jen otázkou té dané společnosti, jestli nám za zaplacené peníze poskytne imaginární účty odkudsi z Asie nebo propaguje náš profil

[2]buy1000followers.co/, www.purchasemorefollowers.co.uk/, buytwitterfollowersreview.org, www.buycheapfollowersfast.com/twitter/

mezi reálnými uživateli, kteří by se mohli stát našimi potenciálními followery [cou(2015)].

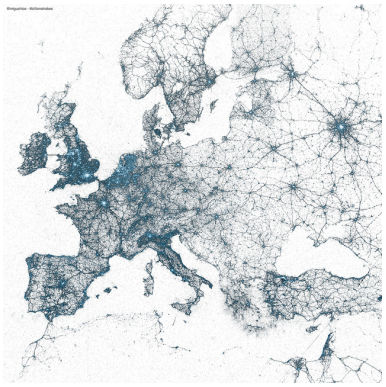
S podivem pak zjišťujeme, že v době psaní této práce jsou první tři nejsledovanější účty Twitteru Katy Perry (68 813 810 followerů), Justin Bieber (62 989 568 followerů) a Barack Obama (58 337 998 followerů). Pro Českou republiku jsou to Petr Čech (1 097 430 followerů), Tomáš Ujfaluši (358 189 followerů) a Petra Kvitová (228 254 followerů) [soc(2015)].

Dalším zajímavým faktem je, že 35% uživatelů spadá do věkové skupiny 20-24let a dalších 31% pak do skupiny 15-19let [sys(2014)].

2.3.1 Češi na Twitteru

Důležitým ukazatelem pro nás je i rozdělení uživatelů podle zemí. Není nic překvapujícího, že Česká republika se nedostává ani do první desítky zemí, která má nejvíce uživatelů Twitteru. Nejvíce uživatelů Twitteru je ze země původu této sítě - tedy USA, na druhém místě pak Velká Británie a dále Kanada, Austrálie, Brazílie, Německo, atd. [for(2014)] A jelikož se v této práci budeme zabývat pouze českou komunitou Twitteru, je pak právě tato statistika příčinou toho, že výsledky této práce nejsou nijak objemné.

Zajímavostí je, že pro Twitter API, kterému je věnována kapitola 2.6, je Česká republika ještě málo známou. Tweety, označené autorem jako česky psané, totiž API označuje kódem „sk“ a tak se musíme vypořádat i s občasným výskytem slovenských textů. Tato chyba se už nevyskytuje u uživatelských profilů, kde se setkáváme se zkratkou „cs“.



Obrázek 2.1: Evropa z pohledu tweetů [geo(2013)].

Pro zajímavost vytvořil Twitter několik obrázků tak, že každý tweet, obsahující pozici odeslání (tzv. geotag), byl označen bodem do souřadnicového prostoru. Jak je vidět na obrázku 2.1 [geo(2013)], lze zřetelně rozeznat větší a aktivnější města Evropy.

2.4 Seznamy uživatelů

Zajímavou funkcí Twitteru je možnost seskupovat uživatelské profily do veřejných či soukromých seznamů a mít tak k dispozici data konkrétních autorů. Tato možnost je primárně určena pro implementaci časové osy více uživatelů do webových stránek, lze ji ale využívat i k našim účelům.

Nás budou na toto téma zajímat především následující omezení. Každý uživatel si může vytvořit až 20 seznamů, soukromých či veřejných a každý seznam pojme až 5 000 uživatelů.

2.5 Obsah Twitteru

V předchozích kapitolách jsme se dozvěděli, kdo se může podílet na sdílení informací a jak dlouhé zprávy může psát. Z těchto dvou témat pak přímo vyplývá další, a to - co mohou psát.

At' už se jedná o společnost, celebritu či soukromou osobu, každý má na vyjádření své myšlenky jen těch málo 140 znaků. To není dost ani na vypsání krátkého receptu. Proto si uživatelé vypomáhají různými zkratkami, odkazy, emotikony, obrázky a hlavně žargonem, specifickým právě pro sociální sítě. Dále údajně 80% aktivních uživatelů přistupuje ke svému účtu z mobilních zařízení [abo(2015)]. To přináší i překlepy a špatnou korekci mobilních zařízení. Z pohledu strojového zpracování to zanáší do zpracovávaného textu chybovost a neúplnost a komplikuje další zpracování.

2.6 Twitter API

Twitter API je veřejně dostupná knihovna funkcí, které slouží pro přístup k datům Twitteru. I přes jednoduchost Twitteru obsahuje nepřeberné množství funkcí pro správu autentizace, uživatelských účtů, tweetů, uživatelských seznamů, streamů a dalších.

Je třeba zmínit, že struktura Twitteru se neustále mění. Každou vteřinou vznikají noví uživatelé, přibývají, ale i ubývají tweety. Práce s API musí na tuto skutečnost brát ohled a počítat například s tím, že pokud získáme seznam identifikátorů nějaké skupiny uživatelů, druhý den už nemusí být všichni dostupní (mohou být smazaní či skrytí) a Twitter API pak bude vracet HTTP chybový kód.

Od května roku 2013 Twitter oficiálně uzavřel podporu REST API v1.0 a přešel na novou verzi API v1.1. Ta přináší vesměs nepatrné změny, jako je přechod z XML na JSON, zavedení OAuth autentizace a další. Ale hlavní změnou, která se úzce dotýká našeho tématu, je zpřísnění limitů. V předchozí verzi byl povolen jeden dotaz za vteřinu a ve verzi nové už je to jeden dotaz za pět vteřin [gni(2013)]. Což je změna výrazná a značně ovlivňuje směr a výsledky této práce.

Většinu API knihovny lze využívat zdarma a je přístupná pro každého registrovaného uživatele. K tomu ještě existuje jedna funkce, která je zpoplatněná a která splňuje všechny naše požadavky. Touto funkcí je Sream Firehose, což je stream všech tweetů, které prochází tímto kanálem v reálném čase. Více o Stream Firehose v kapitole 2.6.5.

Vlastní komunikace s API knihovnou pak funguje na základě HTTP dotazů, na které získáme odpověd nesoucí JSON data. Jelikož byl pro práci zvolen programovací jazyk Java, použili jsme knihovnu třetí strany - Twitter4j - která nás od této komunikace odstíní.

2.6.1 Podmínky použití

Twitter ve svých podmínkách mimo jiné píše, že další šíření dat třetím stranám lze pouze ve formě id tweetu a/nebo id uživatele. Pro šíření větších korpusů je povolen jen nezautomatizovaný způsob stahování a objem do 50 000 veřejných tweetů a/nebo uživatelských dat za den [pol(2014)].

Díky těmto restrikcím je velice náročné získat větší objem dat například na vytrénování procesu lemmatizace (viz. kapitola 3.4) a omezíme-li se jen na česká data, stažení těchto dat je velice nepravděpodobné.

2.6.2 Twitter4j

Twitter4j[3] je neoficiální Java knihovna pro Twitter API, pomocí které můžeme snadno integrovat do naší Java aplikace všechny služby Twitteru. První release byl vydán roce 2007 (15 měsíců po vzniku Twitteru) a od té doby je dále vyvíjen a šířen jako opensource pod licencí Apache License 2.0.

Díky této knihovně je každý Java programátor odstíněn od správy HTTP odkazů, hlaviček a veškeré komunikace přes tento protokol. Stačí mu jen inicializovat spojení s Twitter API, poskytnout knihovně Twitter4j uživatelské klíče a přístupové tokeny a HTTP dotazy na Twitter API již probíhají jednoduchým voláním funkcí Twitter4j. Jako výsledky takového volání pak nedostáváme textové řetězce a datové objekty, jak je tomu u HTTP volání, ale již připravený Java objekt naplněný žádanými daty.

2.6.3 Twitter Rate Limit

Tato knihovna spravuje veškerá omezení Twitteru. Je rozdělená předně na omezení pro uživatele a omezení na aplikaci. Rozdíl mezi těmito omezeními je v tom, že jeden uživatel může mít pod svou správou více aplikací.

Velká změna oproti API v1.0 je v dělení časových intervalů. Dříve se limity měřily po 60ti minutách, API v1.1 přináší změnu na 15ti minutové intervaly.

Twitter dělí funkce své API knihovny podle omezení na několik tříd. Mezi základní řadíme tyto dvě:

1. Méně časté dotazy, jako je například čtení soukromých zpráv, získání seznamu přátel, followerů, seznamů uživatelů a další. Pro tyto dotazy je přiřazen limit 15 dotazů za 15 minut.

2. Frekventovanější dotazy, jako je například výpis členů seznamu uživatelů, hledání tweetů, výpis tweetů z timeline a další. Pro tyto dotazy

[3] http://twitter4j.org

je přiřazen limit 180 dotazů za 15 minut.

Konkrétní limity jednotlivých funkcí jsou dostupné v dokumentaci API[4].

2.6.4 REST API

REST API poskytuje programový přístup ke čtení a zapisování dat do Twitteru. Mezi funkce z této knihovny patří například vytvoření nového tweetu, čtení uživatelského profilu, vyhledávání tweetů či uživatelů. V praxi všechny funkce spojuje to, že autentizovaný programátor odešle požadavek ve tvaru HTTP a jako odpověď dostává JSON pole.

V následujících vybraných funkcích, které jsou pro nás zajímavé, uvádíme teoretickou rychlost čtení. Ta je dána omezením Twitteru (tedy knihovnou Twitter Rate Limit) a je to tedy omezení maximální rychlosti získávání tweetů. Abychom ale dosáhli této rychlosti, musí Twitter obsahovat dostatečné množství dat. V tomto bodě se liší teoretická rychlost od praktické. Protože jak uvidíme v další části práce, Twitter nám sice nabízí například 12 000 tweetů za hodinu, uživatelé Twitteru ale tolik příspěvků nenapíší a tak je reálná rychlost značně nižší.

GET statuses/home_timeline

Vrací kolekci tweetů a retweetů přihlášeného uživatele a uživatelů, které sleduje. Touto metodou lze přečíst jen 800 tweetů z historie, není tedy doporučována pro ty, kteří sledují mnoho dalších uživatelů nebo sledují uživatele, kteří častěji tweetují.

Teoretická rychlost čtení je maximálně 12 000 tweetů za hodinu.

GET list/statuses

Je způsob, jak z již vytvořeného seznamu přečíst data. Číst můžeme maximálně po 100 příspěvcích a stačí nám k tomu znát jen id seznamu nebo

[4]https://dev.twitter.com/rest/public/rate-limits

vlastníka seznamu. Tak můžeme přistupovat i k seznamům pod jiným uživatelským účtem, než pod kterým jsme právě přihlášeni.

Teoretická rychlost čtení je maximálně 72 000 tweetů za hodinu.

Zajímavými vlastnostmi těchto dvou možností je, že ať už čteme tweety z home_timeline nebo ze seznamu, vždy čteme příspěvky od uživatelů, které jsme sami vybrali. Tím ztrácíme obecný vzorek a zanášíme do dat subjektivitu, ale také máme cenný nástroj, jak filtrovat nezajímavé a nežádané příspěvky.

GET search/tweets

Vrací kolekci relevantních tweetů, které korespondují se zadaným dotazem. Vyhledávat lze jedním řetězcem (Twitter si vytvořil vlastní vyhledávací syntaxi, pomocí které lze v jednom řetězci filtrovat například všechny tweety, obsahující klíčové slovo a odeslané z okolí Prahy), nebo pomocí datové struktury, ve které lze filtrovat podle lokace, jazyka nebo typu tweetu. Typ může být „populární“, „nedávný“ a „kombinovaný“.

Teoretická rychlost čtení je maximálně 72 000 tweetů za hodinu.

Výhodou této varianty je, že můžeme vyhledávat tweety od všech uživatelů Twitteru a tím neztrácíme na obecnosti.

2.6.5 Stream API

Tato část funkcí poskytuje nejrychlejší kontinuální přístup ke tweetům a je určena pro aplikace, které se jednou spustí, nastaví spojení a pak již jen čtou data v nekonečné smyčce.

Existují tři druhy streamů.

1. Public stream, který je dále dělen na „sample“, „filter“, „firehose“.
    - Sample poskytuje vzorek všech globálních dat.
    - Filter umožňuje jejich filtraci podle uživatele, klíčových slov nebo lokace.

- A jako poslední je Firehose, který poskytuje neomezený stream všech tweetů. Ten, jak již bylo zmíněno dříve, je zpoplatněn a přístup k němu není nijak intuitivní. Twitter sám nezveřejňuje ceny ani návod k získání přístupu a po marných snahách o připojení vrací jen stručnou hlášku, že přístup k tomuto streamu je podmíněn vlastnictvím speciálních práv. Twitter totiž tuto kapitolu „neomezeného přístupu k datům“ svěřil společnostem třetí strany, které pak mají práva data přeprodávat dál. Jsou to primárně společnosti Gnip a DataSift[5].

2. User stream, který poskytuje všechna data přihlášeného uživatele. Tedy příspěvky, které by přihlášený uživatel viděl na své timeline.

3. Site stream, což je verze User stream pro více uživatelů. Tato verze je určena pro servery, přes které se připojuje k Twitteru více uživatelů.

[5]DataSift ovšem oznámil konec spolupráce s Twitterem k srpnu 2015[dat(2015)]

3 Zpracování textu

V této kapitole se budeme zabývat různými možnostmi práci s textem, jeho analýzou, klasifikací a dalšími způsoby zpracování textu, které jsou potřebné pro úspěšné vyřešení této práce. Představíme si princip procesu filtrace, lemmatizace a clusteringu a jejich přínosy.

3.1 Klasifikace textu

Jde o učící algoritmy, které určují, do které z kategorií dat dané pozorování patří. Pro natrénování klasifikátoru se používá anotovaná trénovací množina, pro která jsou kategorie správně určeny. Jednotlivá pozorování jsou analyzovány do skupin kvantifikovatelných vlastností (rysů).

Algoritmus, který implementuje klasifikaci, se nazývá klasifikátor. Jako příklad takového algoritmu je kategorizování e-mailové pošty do spamu či přiřazení diagnózy pacientovi na základě jeho pozorovaných vlastností.

Ve strojové terminologii můžeme klasifikační metody dělit podle typu učení. Oblíbenou metodou je učení s učitelem (tzv. *Supervised learning*) a do této kategorie spadá i *Naivní Bayesův klasifikátor* popsaný níže.

Dalším typem klasifikace je učení bez učitele (tzv. *Unsupervised learning*). Tyto algoritmy jsou vhodné v případě, kdy nemáme k dispozici anotovaná vstupní data a rozdělují dokumenty do tříd na základě jejich vnitřní podobnosti. Nevýhodou této metody je, že nedokáže výsledné třídy automaticky pojmenovat a je zde nutný zásah uživatele. Tuto metodu použijeme a proces shlukování popíšeme níže v kapitole 3.5. Hlavním důvodem této volby je to, že nemáme k dispozici anotovaná data Twitteru, která bychom mohli použít k trénování klasifikátoru.

3.1.1 Naivní Bayesův klasifikátor

Je klasifikační technika založená na Bayesově větě o podmíněných pravděpodobnostech. Pomocí tohoto klasifikátoru a anotované vstupní množině dat jsme schopni rozhodnout, do jaké kategorie (nebo do jakých kategorií) jaký

dokument spadá. To by v našem případě velmi usnadnilo zpracování a po řádném natrénování tohoto algoritmu bychom byli schopni třídit tweety do již pojmenovaných kategorií.

Je to často používaný nástroj, který sice nedosahuje nejlepších výsledků, zato je snadný na implementaci a nepotřebuje velké množství dat k natrénování. Předpokladem pro tento klasifikátor je nezávislost příznaků.

Funkčnost a vlastnosti algoritmu popisuje například Michal Hrala ve své diplomové práci [Hrala(2012)].

3.2 POS-tagging

Part of Speech (POS) tagování, je metoda, která každému tokenu (dále jen slovu) přiřadí slovní druh. Tímto nástrojem lze dojít k zjednodušení filtrace slov pro klasifikaci, protože například předložky a spojky jsou pro klasifikaci většinou nepodstatné.

Dále bychom tímto nástrojem mohli odlišovat zpracování podstatných jmen od přídavných jmen nebo sloves. Po detailnějším prozkoumání ale tento nástroj neshledáváme přínosným a k filtraci použijeme rychlejší způsob jednoduchým výčtem filtrovaných slov (viz. další kapitola.)

3.3 Filtrace

Pro další zpracování velkého objemu tweetů nebudeme potřebovat všechny. Z Twitter API stahujeme příspěvky bez ohledu na myšlenku a význam textu. V tomto kroce se soustředíme na to, abychom odfiltrovali maximum příspěvků, které se netýkají žádné události a které dokážeme jednodušše identifikovat. Dále zavedením filtrace docílíme i urychlení systému. To díky tomu, že proces lemmatizace je časově náročný a filtrací snížíme velikost dat ke zpracování.

Filtraci rozdělíme na filtrování celých tweetů a to v dalším kroce zjemníme na filtrování jednotlivých částí tweetu. K detekci události totiž nebudeme potřebovat například spojky, předložky, citoslovce, apod.

3.3.1 Filtrace celých tweetů

V této části chceme ze získaných dat odebrat ta, která jsou psaná automatem (”Líbí se mi video @YouTube od autora ...”, ”Přidal(a) jsem do seznamu videí @YouTube video ...”, atd.) a dále maximum tweetu nevýznamových, spamů a dennodenních výkřiků, které nejsou pro naši věc zajímavé.

Jednoduše jsme schopni filtrovat pouze tweety psané automatem, protože zůstávají stejně formulované a v programu je vyfiltrujeme pouhým použitím výčtu. Vzniká tak pole - volně rozšiřitelné - obsahující části tweetů, které nebudeme zpracovávat. Toto pole může v našem případě vypadat například takto:

```
private static final String[] TABU = {
  "Přidal jsem novou fotku na Facebook",
  "Líbí se mi video @YouTube",
  "Přidal(a) jsem do seznamu videí @YouTube",
  "Označil(-a) som video @YouTube"
};
```

Ostatní nevýznamové tweety můžeme filtrovat podle různých pravidel, která se snaží odhadnout podstatu nevyžádanosti a vystihnout ji nějakým strojovým pravidlem. Například tým ze skotské univerzity považoval za nevýznamové tweety obsahující 3 a více hashtagů, 3 a více odkazů na další uživatele nebo více než 1 odkaz [McMinn et al.(2013)McMinn, Moshfeghi,, Jose]. V našem případě ale nepracujeme s větším množstvím takových tweetů a tuto metodu nepoužijeme.

Texty tweetů, které nám i přes filtraci proniknou do dalších procesů zpracování, zanedbáme a jejich „znehodnocení“ proběhne až při shlukování. V tomto procesu si totiž zpravidla nenajdou dostatečně velkou množinu podobných tweetů, aby se staly důležitější součástí výsledků.

Spam na Twitteru

Tým z brazilské univerzity *Universidade Federal de Minas Gerais Belo Horizonte* se zabýval problémem spamu na Twitteru, který definovali jako příspěvěk, obsahující slova typická pro současné trendy, podobné hashtagy a URL

adresy[1], které ale vedly na nesouvisející a tedy nevyžádané stránky - spam. [Benevenuto et al.(2010)Benevenuto, Magno, Rodrigues„ Almeida]

Problém nevyžádaných příspěvků můžeme očekávat v každém informačním kanálu, který je objemný a sledovaný. Což dle našich prozatimních výsledků Twitter v České republice ještě není a tak se nebudeme zabývat ani problémem spamu.

3.3.2 Filtrace částí tweetu

Pro další zpracování můžeme již v tomto místě promazat některé části textu tweetu.

3.4 Lemmatizace

Je proces, kdy je slovo převedeno do základního tvaru - tzv. lemma. Pomocí lemmat dokážeme v následující kapitole shlukovat i slova v různých tvarech, což přináší přesnější výsledky. Díky tomuto nástroji totiž dokážeme porovnat například slova *policistu* a *policisté* a vyhodnotit je jako totéž slovo *policista*.

Tento nástroj je v některých jazycích důležitější než v jiných. Porovnáme-li například možnosti formálního tvarosloví (časování, skloňování) anglického jazyka s češtinou, vyjde čeština jako jazyk bohatší a tedy jako vhodný kandidát pro lemmatizaci. V této práci bude lemmatizace přínosným nástrojem.

Problém nastává při práci s tweety. Jednotlivá témata uživatelé popisují převážně vlastními slovy, zkratkami nebo využitím emotikon a tak i lemmatizace může být v některých příkladech nástroj nedostatečný. Bohužel nemáme v současné době k dispozici žádný nástroj natolik robustní, aby dokázal propojit slova s překlepem, gramatickou chybou nebo slangové výrazy, kterých je Twitter plný.

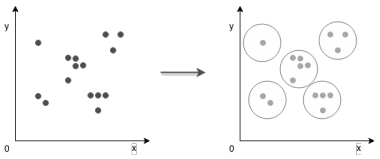
Lemmatizátor, podrobněji popsaný v kapitole 4.4, je knihovna založená na „učícím algoritmu“. Poskytnutý lemmatizátor[2] byl natrénován na korpusu

[1]URL adresy jsou na Twitteru typicky zkracovány pomocí služby tzv. „URL shortener“. Důvodem je omezení délky tweetu na 140 znaků.

[2]http://code.google.com/p/mate-tools/

PDT 2.0 [Hajič et al.(1996)Hajič, Hajičová„ Rosen]. Jazyk v oblasti sociálních médií a zejména Twitteru je odlišný od jazyka korpusu PDT 2.0. Bylo by proto vhodné lemmatizátor natrénovat na datech z Twitteru. To ale bohužel není možné, protože není k dispozici anotovaný korpus.

3.5 Shlukování



Obrázek 3.1: Princip shlukové analýzy.

Je proces organizování objektů do skupin tak, aby si jednotky ze stejné skupiny byly podobnější než objekty z ostatních skupin. Výsledkem shlukování je skupina objektů, které jsou si navzájem podobné a nejsou si podobné s objekty jiné skupiny (viz. obrázek 3.1).

Základní dělení algoritmů shlukové analýzy je:

1. Hierarchické algoritmy = Sekvence vnořených rozkladů, která na jedné straně začíná triviálním rozkladem, kdy každý objekt dané množiny objektů tvoří jednoprvkový shluk, a na druhé straně končí triviálním rozkladem s jedním shlukem obsahujícím všechny objekty. Podle směru postupu při shlukování dělíme metody na aglomerativní a divizní.

2. Nehierarchické algoritmy = Jde o algoritmy, ve kterých je nutné znát počet shluků předem.

Mezi nehierarchické metody patří typicky metoda *K-means*, která vyhledává shluky v prostoru na základě vzdáleností od centrálního prvku. Tato metoda je efektivní a velmi oblíbená, pro její použití je ale třeba dopředu znát počet hledaných shluků. V našem případě tuto hodnotu dopředu neznáme a proto tuto metodu nepoužijeme.

Divizní hierarchické algoritmy berou vstupní množinu objektů jako celek a ten pak dělí. V každém kroku dělí shluk na dva nové, které nejlépe splňují

dané kritérium rozkladu. Tento postup je výpočetně náročný a je proveditelný pro malý počet vstupních objektů. V našem případě pracujeme s velkými objemy dat a tak ani tato metoda není vhodná.

Princip aglomerativního hierarchického algoritmu spočívá v tom, že na počátku rozdělíme celek na jednoprvkové shluky. V dalších krocích pak vybíráme dva nejpodobnější shluky, tyto sloučíme a vytvoříme tak nový shluk. Tato metoda vykazuje všechny hledané vlastnosti a na rozdíl od ostatních nám nic nebrání k jejímu použití. V dalším textu detailněji popíšeme její princip.

Jelikož je v našem případě objektem tweet, musíme najít způsob, jakým počítat vzdálenosti mezi těmito textovými řetězci. Shlukování rozšíříme o další funkci, ve které vznikne nová forma reprezentace tweetu - forma číselného vektoru - s jejíž pomocí budeme schopni počítat vzdálenosti mezi jednotlivými objekty.

Proces shlukování se skládá z následujících částí.

3.5.1 Převod tweetu na číselnou reprezentaci

V této části řešíme problém, jak počítat vzdálenost mezi texty, respektive jakou metodu převodu textu na číselné vyjádření zvolit.

TF-IDF

Pro tento druh problému je nejznámější a tedy i nejpoužívanější metoda TF-IDF (term frequency–inverse document frequency). Tato metodika se používá pro hodnocení relevance při vyhledávání v textu a jak již název napovídá, toto hodnocení se skládá ze dvou částí:

1. četnost slova v dokumentu,
2. převrácená četnost slova ve všech dokumentech.

Hodnotu TF spočteme následovně:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}},$$

kde $n_{i,j}$ je počet výskytů slova t_i v dokumentu d_j, a součet ve jmenovateli vyjadřuje počet všech slov v dokumentu d_j. Hodnotu IDF spočteme jako:

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

kde $|D|$ je počet dokumentů, ve kterém hledáme a jmenovatel vyjadřuje počet dokumentů, které obsahují hledané slovo i.

Pro konkrétní výpočet se pak získané hodnoty vydělí maximem pro získání normovaného tvaru a hledaná hodnota TF-IDF se spočte jako součin těchto normovaných hodnot [Salton – Buckley(1988)Salton, Buckley].

Binární reprezentace

Jelikož pracujeme s velmi krátkými textovými řetězci, nebude pro nás metodika TF-IDF nejefektivnější. Proto si představíme alternativu pro převod textu na číselný vektor.

V našem případě máme dokument d, skládající se z množiny tweetů t a pole slov p, které obsahuje všechna slova právě zpracovávaného dokumentu. Tweety budou obsahovat vektor v, který budeme v tomto algoritmu plnit hodnotami nula nebo jedna. Jednička v případě, pokud je slovo na daném indexu v poli p v tweetu obsaženo, v opačném případě nula. Tento postup je zachycen pseudokódem v algoritmu 1.

Výsledný číselný vektor si pak můžeme představit jako pozici tweetu v N–rozměrném prostoru a v takovém prostředí již dokážeme počítat jejich vzdálenosti.

Dále je třeba zmínit možnost využití hashtagů, na které můžeme reagovat vyšší číselnou hodnotou ve vektoru. Tuto modifikaci si ale detailněji představíme až v realizační části (kapitola 4.5).

Algoritmus 1 Převod tweetu na číselný vektor

```
for all t ∈ d do
    i := 0
    for all x ∈ p do          ▷ Cyklus přes všechna slova v dokumentu.
        if t obsahuje x then      ▷ Hledáme slovo v aktuálním tweetu.
            t → v[i] := 1
        else
            t → v[i] := 0
        end if
        i++
    end for
end for
```

3.5.2 Výpočet vzdálenosti

Nyní již pracujeme s body v N-rozměrném prostoru. Abychom je mohli shlukovat do skupin, musíme znát jejich vzájemné vzdálenosti. K tomu bychom mohli použít Euklidovskou, Hammingovu nebo Kosínovou vzdálenost.

Hammingova vzdálenost

Tato vzdálenost je definována jako vzdálenost slov abecedy. Tedy počet bitů/pozic, které se musejí změnit, abychom jedno slovo změnili za druhé. Pro body A a B je definována vztahem:

$$|AB| = \sum_{i=1}^{n} |a_i - b_i|$$

Jako výsledek vrací vždy celé číslo a to v našem případě není dostatečně jemné dělení. Navíc bychom se připravili o možnost dále manipulovat s prioritou hashtagu (viz. kapitola 4.5.1).

Euklidovská vzdálenost

Výpočet této vzdálenosti je znám z elementární geometrie a vztah mezi bodem A a B je definován následovně:

$$|AB| = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

I toto vyjádření vzdálenosti pro nás není postačující. A to hlavně z toho důvodu, že se budeme vyskytovat v N-rozměrném prostoru s vysokým N, což by přinášelo vysoké hodnoty vzdáleností a zbytečně bychom je museli normovat.

Kosínová vzdálenost

Vybíráme Kosínovou, která se zakládá na kosínové větě. Ta má obor hodnot v intervalu $< -1; 1 >$ a není tedy třeba ji ještě nějak upravovat. Pro vektory A a B se spočítá podle následujícího vzorce:

$$|AB| = \frac{A \times B}{||A|| \cdot ||B||} = \frac{\sum^{n} a_i \cdot b_i}{\sqrt{\sum^{n} (a_i)^2} \cdot \sqrt{\sum^{n} (b_i)^2}}$$

Při samotném výpočtu pak výsledek určuje počet společných slov. Pro nulu nemají tedy dva tweety žádné společné slovo, pro jedničku mají všechna slova totožná. Jelikož vzdálenost počítáme pro tweety v lemmatizovaném tvaru, vzdálenost rovna jedné ještě nezaručuje rovnost tweetů v originálním znění.

3.5.3 Vytvoření shluku

V tomto bodě již máme vše připraveno pro vytváření shluků na základě podobnosti tweetů. Vytvoření shluku proběhne při nalezení první podobné dvojice. Avšak zde nastává problém, že tuto podobnost nemáme nikde definovanou. Jde o určení prahové konstanty (pojmenujeme ji T), která udá, zda jsou si tweety podobné či nikoliv.

Tato konstanta T výrazně ovlivní přesnost a úplnost výsledků a její nalezení budeme řešit experimentálně v praktické části práce.

Samotné vytvoření shluku navrhneme následovně. Procházíme všechny tweety v cyklu (označíme je A) a pro každý další tweet (označený B) po-

činaje následovníkem A, který ještě není v clusteru, počítáme jejich vzdálenost. Pokud nalezneme tweet B dostatečně podobný, vytvoříme nový cluster, do kterého tyto dva tweety vložíme. Pokud narazíme na tweet B, který již v clusteru je, přeskočíme ho. Pokud pak v dalším průchodu cyklem pracujeme s tweetem A, který jsme již přidali do nějakého clusteru, pokračujeme jak je popsáno výše jen, jen s tou vyjímkou, že jakmile najdeme jemu podobný tweet B, nevytváříme nový cluster, ale tweet B přidáváme do již existujícího clusteru k tweetu A.

3.5.4 Prezentování shluku

V tomto bodě máme množinu skupin tweetů, kde každá skupina vyjadřuje jednu událost. Aby byly tyto výsledky dobře čitelné, je třeba zvolit formu zobrazení, která bude co nejlépe vystihovat podstatu a důležitost události.

Zadefinujeme tedy množinu důležitých slov, které použijeme k reprezentaci události. Tyto slova najdeme díky principu výpočtu vzdálenosti tak, že hledáme slova obsažená v každém tweetu (nebo v maximu tweetů) zkoumaného shluku.

S touto množinou slov můžeme vytvořit název shluku několika způsoby. Nabízí se například použít slovní spojení, které obsahuje co nejvíce důležitých slov a má vysokou frekvenci výskytu nebo jen vybráním jednoho z nich. V našem případě použijeme jako název skupiny jeden konkrétní tweet, který obsahuje všechna důležitá slova (popř. nejvíc důležitých slov) a zároveň co nejméně slov nedůležitých. Tím docílíme toho, že nevznikne zkomolená věta či nevýznamové spojení slov, ale ve stručnosti vyjádříme podstatu události. Toto řešení bylo zvoleno i na základě preferencí České tiskové kanceláře.

3.5.5 Důležitost shluku

Další otázkou při prezentování událostí je jejich ohodnocení či přímo řazení. Možností je hned několik a jejich složitost se pohybuje od výpočtu velikosti shluku až po práci s Wikipedií (viz. například autoři systému *Twevent* [Li et al.(2012)Li, Sun„ Datta] nebo skupina z univerzity v Glasgow v jejich práci „Building a large-scale corpus for evaluating event detection on twitter“ [McMinn et al.(2013)McMinn, Moshfeghi„ Jose]). Autoři těchto dvou prací

používají on-line službu Wikipedie - tzv. Portál:Aktuality[3], podle kterého teprve rozhodnou, zda se jedná o událost důležitou, či nikoliv. To ovšem není použitelné v našem případě, protože my se snažíme detekovat události ještě dříve, než se dostanou na Wikipedii.

Pro klasifikaci důležitosti vyjdeme z původu informace o události, tedy z tweetů. Pokud jde o událost důležitou, uživatelé Twitteru na ni reagují rychlým šířením a zvýšenou aktivitou na dané téma. To má za následek výskyt objemných shluků a někdy i více podobných shluků[4]. Jednoduchý počet tweetů ve skupině tedy určí, zda se jedná o událost důležitou, či nikoliv.

[3]http://cs.wikipedia.org/wiki/Port%C3%A1l:Aktuality

[4]To záleží na vstupních parametrech systému. Pokud bude prahová konstanta T blíže hodnotě jedna, pak bude vznikat více duplicit událostí a naopak.

4 Systém na sledování trendů

V předchozích kapitolách jsme se seznámili se sociální sítí Twitter a její API, dále jsme si představili metodiky pro zpracování textu se zaměřením na text přirozeného jazyka a máme tedy připraveno vše k tvorbě vlastínho programu.

V této kapitole popíšeme samotné řešení práce, ve kterém vznikne nový systém pro sledování trendů na Twitteru. Tento program je k nalezení na přiloženém nosiči.

4.1 Stahování tweetů

Pro přístup k datům slouží několik funkcí poskytovaných Twitterem. Každá je něčím specifická a vhodná pro nějaký konkrétní problém. K řešení našeho problému bychom potřebovali objemný, souvislý přísun dat v reálném čase, filtrovaný podle jazyka. Tento popis by přesně vystihl pouze Stream „firehose“, který bohužel musíme vyřadit s ohledem na omezený rozpočet. Je tedy hledána alternativa, kterou Twitter API nabízí zdarma.

4.1.1 „Duplikáty“ na Twitter API

Je třeba zmínit jednu nevýhodu Twitter API, která se přímo týká našeho zadání. Každý uživatel má právo mazání svých příspěvků. To je funkce důležitá a v některých případech i žádaná a používaná, protože je to jediný způsob, jak opravit chybu či překlep v již odeslaném textu. Twitter API ale nedostane žádnou zprávu, že tento aktuální příspěvek je téměř stejný jako předchozí, který byl mezitím smazán a tak se často stane, že se musíme vypořádat s „duplikáty“, ve kterých je změněno jedno písmeno, slovosled či některá slova. V našem program bychom na to mohli reagovat například vyřazením staršího příspěvku. Nejde ale o jev tak častý, proto jej pro účely této práce nebudeme uvažovat.

Jako příklad zmíníme dva získáné příspěvky z filtrovaného Streamu, ze kterých první byl odeslán, zkritizován a poté smazán a nahrazen jiným - mírně upraveným textem.

- Opelka a sedm trpaslíků. Pohádka o jedné milovnici něm. aut. #OdeberZnázvuFilmuJednoPísmenoAkratcePopiš
- Opelka - pohádka o milovnici německé zn. aut. #OdeberZnázvuFilmuJednoPísmenoAkratcePopiš

Z příkladu je vidět, že jde o tentýž status od stejného autora, ale ve zpracovávaných datech se nám vyskytne jako dva nezávislé tweety.

4.1.2 Teoretická rychlost stahování

Pro přehled uvedeme tabulku se souhrnem informací z kapitoly 2.6.

Funkce	**Teoretická rychlost za hodinu**
HomeTimeline	12 000 tweetů
Search	72 000 tweetů
UserList	72 000 tweetů
FilteredStream	nezveřejněno

Dle těchto čísel můžeme vyloučit z dalšího postupu funkci HomeTimeline, která neposkytuje dostatek dat a nahradí ji bez ztráty na výkonu funkce UserList. Pracujeme tedy s metodami Search, UserList a FilteredStream.

4.1.3 Reálná rychlost stahování

Pro další zúžení výběru zjistíme reálnou rychlost stahování tweetů. Byl proveden pokus, kdy stahování probíhalo 7 dní (pro zachycení rozdílů v jednotlivých dnech) a ze získaných dat byla spočtena reálná rychlost stahování.

Výsledky byly následující.

Funkce	**Reálná rychlost za hodinu**
Search	43,5 tweetů
UserList	324,3 tweetů
FilteredStream	56,6 tweetů

Je třeba zmínit, že tato čísla vyjadřují počet stažených tweetů ještě před filtrací. Tedy všech příspěvků, zahrnujících příspěvky automaticky odesílané jinými službami (Facebook, YouTube, Instagram, atd.), reklamy a spam. Tabulka vyjadřuje rychlost stahování, kterou Twitter API poskytuje data na jeden přístupový klíč. K urychlení ale nedojde ani při zopakování experimentu na silnějším výpočetním prostředí ani při navýšení počtu přístupových klíčů. Proto musíme vyloučit možnosti paralelizace či jiných urychlovacích technik.

V dalších sekcích upřesníme průběh těchto experimentů.

4.1.4 Možné zdroje dat

FilteredStream a Search

Pro funkce FilteredStream a Search jsou stanovena kritéria filtru následujícím dotazem. Vybírány jsou pouze české tweety lokalizované na území České republiky. Pro označení lokace nabízí Twitter hned několik způsobů. Mezi ně spadá například název města a nebo využití GPS souřadnic. V příkladu níže jsme použili obdélníkový výřez v souřadnicové síti, který obsahuje Českou republiku. Díky propojení s filtrem jazyka je to metoda úspěšná.

```
FilterQuery query = new FilterQuery();
double[][] location
   = new double[][] { { 12.124470, 48.585767 },
                      { 18.827669, 50.869807 } };
query.locations(location);
query.language(new String[] {"sk"});
```

Jelikož ale tato metoda poskytuje data od obecného vzorku uživatelů Twitteru, nejsou získané tweety pro naši potřebu nejvhodnější. Jak bylo zmíněno v kapitole 2.3, 66% uživatelů Twitteru spadá do věkové skupiny 15-24 let a proto se příspěvky získané metodami FilteredStream a Search týkají převážně zájmů a aktivit této věkové skupiny. Pro zpravodajskou službu ale není důležité, jestli si lidé přejí dobré ráno, jestli mají problémy ve vztazích či co zrovna pijí.

UserList

Práce s vybranými uživateli vykazuje zatím nejlepší výsledky a je zároveň i nejkomplexnějším způsobem pro získávání dat, proto ji rozebereme obsáhleji.

Jde o možnost, která je obsažena v každém uživatelském účtu a která již byla zmíněna v kapitole 2.4. Díky této části Twitteru můžeme tedy odebírat všechny příspěvky[1], které tito uživatelé zveřejnili na svých profilech.

Díky vlastnostem této metody a její rychlosti stahování bude tato metoda vybrána k realizaci systému.

Výběr uživatelů záleží jen na konkrétním příkladu, pro který chceme systém použít. Pokud by někdo chtěl systém pro sledování trendů v zahradnictví, stačí mu přidat do seznamů uživatele, kteří se zajímají o zahradnictví. Pokud máme zájem o novinky ze světa sportu, stačí přidat do seznamu profily sportovních komentátorů, sportovních televizních stanic, či profesionálních sportovců.

Způsoby pro nalezení a výběr těchto uživatelů jsou dva.

1. Manuálně. Tzn. procházet Twitter přes jeho webové rozhraní a vybírat uživatele podle našich preferencí. Tato možnost má hlavní výhodu v tom, že se nám do seznamu nedostane neaktivní uživatel nebo člověk píšící nezajímavé texty, ale pouze ověření lidé.
2. Automaticky s využitím Twitter API (např. funkcí GET users/search). Tato možnost je rychlejší, nicméně nejsme schopni strojově kontrolovat všechna kritéria, podle kterých uživatele vybíráme. Například uživatel, který má ve svém profilu vyplněný zájem o auta, tím ještě nezaručuje, že píše jen o autech. V jeho profilu nemusí o autech padnout ani zmínka.

Algoritmů pro výběr uživatelů (ať už manuální nebo automatický) máme hned několik, následuje výčet těch nejzajímavějších.

[1]To, že získáváme všechny příspěvky, je pro nás důležitá informace. V ostatních metodách FilteredStream a Search Twitter poskytuje jen to cca 1% všech dat a tím získáváme jen vzorek, který ani nevíme, podle jakého algoritmu byl vytvořen.

- Vybrat skupinu manuálně vybraných, zajímavých lidí a přes jejich followery postupovat hlouběji přes tuto stromovou strukturu až po zastavovací podmínku. S touto metodou budeme nadále pracovat.
- Z obecného vzorku uživatelů vybírat přes různé dotazy na API uživatele podle:
  - jazyka,
  - zájmů,
  - lokace,
  - atd.

V našem případě byl tento podproblém zjednodušen díky spolupráci s Českou tiskovou kanceláří, která nám poskytla množinu pro ně zajímavých účtů. Z této množiny byla manuálně vyfiltrována část autorů, píšících jinak než česky a zbylo nám 168 uživatelských účtů.

S využitím Twitter API a jeho funkce GET followers/ids jsme pak přes noc získali id followerů těchto účtů, kterých bylo 726 623. Bohužel se nám mezi tyto uživatele opět vmísili i zahraniční autoři a účty nezajímavé. Pro jejich filtraci jsme museli znát jejich detaily (doposud jsme znali pouze jejich id). Pro funkci GET users/show už má Twitter Rate Limit ale přísnější omezení a my jsme potřebovali zpracovat velký objem dat, proto byl vytvořen druhý Twitter účet a paralelně jsme zpracovali těchto 726 623 účtů za několik dní.

Z těchto několika stovek tisíců lidí bylo ovšem jen 165 285 profilů označených jako český, což vyjadřuje například to, kolik mají zahraničních sledovatelů známé české účty (jako jsou například @68Jagr, @MZemanOficialni, @PrahaMesto a @historje).

I tak ale máme více uživatelů, než můžeme uložit do uživatelských seznamů. Musíme tedy vybrat mezi nimi ty pro nás zajímavé. Tento problém jsme vyřešili ohodnocením každého uživatele rankem, následným seřazením podle tohoto ranku a výběrem prvních sto tisíců.

Ohodnocení uživatelského profilu se zakládá na počtu jeho sledovatelů a na počtu jeho zveřejněných tweetů. Je hodně uživatelů, kteří Twitter používají často (mají hodně zveřejněných tweetů), ale nepíší zprávy natolik zajímavé, aby si tím získali další followery. Experimentálně byl vybrán poměr

70:30, kde počet sledovatelů má váhu 70% a počet tweetů 30%. Už z podstaty našeho systému, který má *sledovat* trendy, dáváme větší prioritu sledovanosti nad „upovídaností“. Pro normalizovaný tvar ranku ještě nalézáme přibližná maxima těchto hodnot, kterými daná čísla dělíme a získáváme tak ohodnocení uživatelského účtu v intervalu $< 0; 1 >$.

Přidání uživatelů do seznamu je pak jednoduchá operace, kterou lze opět provést manuálně i automatickým způsobem přes Twitter API. Tato volba záleží jen na osobních preferencích či počtu vybraných uživatelů a oba způsoby dosahují stejných výsledků.

Zajímavostí je, že každý uživatel, který je přidán do seznamu, je na tuto skutečnost upozorněn Twitterem. Může se stát, že pak tito lidé reagují negativně a žádají o smazání ze seznamu či vysvětlení, proč se v takovém seznamu ocitli. V našem případě tak reagovalo 34 lidí ze 99 289.

V našem případě bylo přidáno již zmíněných 99 289 uživatelů během několika hodin s využitím funkce POST lists/members/create. Uživatelů nebylo 100 000 kvůli skutečnosti, že mezi procesem získání detailů followerů a jejich následného přidání do seznamu jich několik (přesně 711) skrylo svůj účet jako soukromý nebo bylo vymazáno. Toto číslo nám poskytuje informaci o dynamičnosti struktur Twitteru a potřeby neustálé kontroly a flexibilních reakcí.

4.1.5 Algoritmus pro čtení dat

V tomto bodě již máme vybraný zdroj dat (UserList) a řešíme problém samotného získávání textů z API Twitteru. Jelikož je Twitter Rate Limit dělen na 15ti minutové intervaly, máme 15 minut na položení 180ti dotazů (princip tohoto omezení byl vysvětlen v kapitole 2.6.3). Pro přečtení všech seznamů potřebujeme 20 dotazů (máme 20 seznamů), tedy za 15 minut dokážeme přečíst všechny seznamy 9x. Samotné čtení trvá zanedbatelnou dobu, proto tedy musíme mezi jednotlivým čtením čekat $15/9 = 1,66 minut$. Tato hodnota platí ovšem pouze pro ideální stav, kdy jsou v seznamech velké toky dat a na dotaz bychom dostávali odpovědi plné nových tweetů. My ale dostáváme pouze jednotky, maximálně několik desítek tweetů na dotaz a tak čekání mezi jednotlivými dotazy není důležitý ukazatel.

Dále se setkáváme s problémem, že potřebujeme mít příspěvky řazené

podle času vytvoření a nyní pracujeme s 20ti stále rostoucími seznamy tweetů, kde u každého udržujeme ukazatel na místo, do kterého jsme tweety již zpracovali. Tento problém vyřešíme jednoduchou úvahou. Ze všech seznamů dokážeme přečíst data do 15ti vteřin (to je závislé na počtu nepřečtených tweetů, výpočetním výkonu stroje, na rychlosti internetového připojení a Twitter API). Důležitým faktem je, že jsme uživatele přidávali do seznamů seřazené, tedy do prvního seznamu jsme přidali uživatele nejsledovanější a nejaktivnější. V praxi tím docílíme toho, že rozdělení počtu příspěvků mezi jednotlivé seznamy má charakter prudce klesající křivky. Tedy že v prvních seznamech jsou nové příspěvky časté a u posledních seznamů je nový příspěvek spíše překvapením. Pokud tedy přečteme v několika vteřinách všechny seznamy, tato data seřadíme a zapíšeme, existuje jen velice nízká pravděpodobnost (experimentálně ověřeno), že mezi čtením prvního seznamu a čtením posledního vznikne nový tweet, který by měl časnější dobu vzniku, než tweet, který přečteme v posledním seznamu.

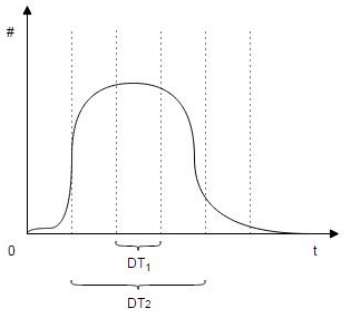
I při výskytu této nechtěné události se ale výsledky v nejhorším případě ovlivní jen tím, že se jeden tweet vůbec nezpracuje - respektive se zpracuje ve špatném časovém úseku (viz. následující kapitola). To koncový výsledek nijak neovlivní a proto se tímto problémem nemusíme dále zabývat.

4.2 Dělení dat do časových úseků

Nyní máme stažená data a je třeba je dále zpracovat. Pro tato zpracování již máme všechny nástroje připravené, otázkou ale zůstává, v jakém okamžiku tyto nástroje aplikovat. Tedy musíme zvolit konstantu DT, která rozdělí zpracovávaná data do časových úseků a tyto části pak bude filtrovat, lemmatizovat a řadit do skupin využitím shlukování.

Pro sledování nově vznikajících trendů stačí porovnávat shluky jednotlivých časových úseků. Pokud pracujeme se shlukem větším, než byl v předchozím časovém úseku, jde o růst zájmu o téma dané touto skupinou tweetů a tedy o nově vznikající trend. Na obrázku 4.1 je znázorněn výskyt nějaké události a možnosti výběru hodnoty DT. Pokud bychom zvolili malé DT (např. jako je na obrázku DT_1), vyskytne se nám událost v několika časových blocích. Ideální by bylo zvolit velikost časového úseku tak velkou, aby pokryla celou událost (viz. DT_2 na obrázku 4.1). Událost se ale může vyskytnout v půlce DT a zasáhnout tak do více úseků, proto k jejímu určení vytvoříme několik experimentů, ve kterých se pokusíme najít optimální hod-

notu.



Obrázek 4.1: Výskyt události a možné hodnoty DT.

Tento proces porovnávání s předchozími byl ale později odebrán a zveřejňovány jsou všechny shluky. A to z důvodu malého množství dat. Většinou získáme tak málo textů na společné téma, že ve výsledných shlucích již dokáže vybrat čtenář pro něj zajímavé trendy sám a není třeba je ještě dále filtrovat.

Otázkou ale zůstává určení konstanty DT, kterou je potřeba vždy zjistit experimentálně a která závisí na typu dat a úlohy (např. zúžení domény - hledáme jen zemětřesení, apod.).

Pro obecnou detekci událostí je možné spustit několik instancí programu a sledovat ty trendy, které mají dobu vzniku v řádu dní (lidé se začnou bavit o plánovaných volbách), ale i trendy vyskytnuvší se během několika minut (teroristické útoky, úmrtí slavné osobnosti, aktuality ze sportovní události, apod.)

4.3 Filtrace

V programu probíhá proces filtrace na dvou místech.

1. Při čtení textu tweetu ještě před jeho uložením do datové struktury dojde ke kontrole, zda neobsahuje část textu, kterou přidává typicky automat a kterou máme připravenou v poli. Pokud nalezneme takový řetězec (např. "Přidal jsem novou fotku na Facebook"), daný tweet rovnou přeskakujeme a do datové struktury, kterou dále budeme zpracovávat, ho vůbec nepřidáme.

2. Při vytváření objektu tweetu plníme tento objekt daty. Mimo jiné si už připravujeme z textu tweetu pole slov, která tweet obsahuje. Z tohoto pole ale vynecháváme všechny URL adresy, emotikony, předložky, spojky, spřežky, apod. Například jsou to slova {alespoň, ačkoliv, během, jakoby, jestli, prostě, všichni, zatím, atd}.

Po tomto kroku máme vytvořenou datovou strukturu množinou tweetů, které budeme dále převádět do základních tvarů pomocí lemmatizátoru a shlukovat do skupin na základě jejich podobnosti.

4.4 Lemmatizace

Proces lemmatizace patří do kategorie učících algoritmů a v přirozené řeči je velice obtížné (až nemožné) dosáhnout 100% správných výsledků.

Pro tento proces použijeme externí knihovnu (tzv. lemmatizátor), který čte slova ze souboru a do jiného souboru zapisuje slova v lemmatizovaném tvaru. Tak získáme například z tweetu na obrázku 4.2 tento proces:

Obrázek 4.2: Vzorový tweet pro znázornění lemmatizace.

1. Nejprve v jaké formě stáhneme tweet z Twitter API:

   ```
   RT @lidovky: ‚Putinovi motorkáři' se chystají do Čech.
   O Noční vlky se už zajímá policie http://ldvk.cz/NKL
   ```

2. Dále vyfiltrujeme (viz. předchozí kapitola 4.3) URL adresy, předložky, emotikony, interpunkci, ... Výsledkem je pole slov:

   ```
   [@lidovky, putinovi, motorkáři, chystají, čech, noční, vl-
   ky, zajímá, policie]
   ```

3. A tím se již dostáváme k lemmatizaci, která vrací tyto výsledky:

```
[@lidovka, putin, motorkář, chystat, č, noční, vlka, zají-
mat, policie]
```

Z tohoto postupu je vidět, že výsledky lemmatizace nejsou zcela správné a místy získáváme slova spíše deformovanější[2]. Jelikož ale lemmatizaci používáme pro obecnější sjednocování více tweetů dohromady, pak naši práci tato deformace z pravidla nepoškodí. Slovo „@lidovky“ se stále bude porovnávat se slovem „@lidovky“ a už nás nijak neovlivní, že se tomu stane přes nesprávný tvar slova „@lidovka“. Výsledky lemmatizace jsou tedy správné či neutrální a to dělá z tohoto procesu přínosný nástroj pro další řešení.

4.5 Shlukování

V kapitole 3.5 jsme si již představili všechny potřebné nástroje k této problematice a nyní popíšeme jejich implementaci.

V tomto bodě pracujeme s množinou tweetů, kde jeden tweet nám reprezentuje objekt, obsahující tyto atributy:

- id,
- datum vytvoření,
- jazyk,
- text tweetu,
- pole slov
- a pole lemmatizovaných slov.

S těmito daty a s výčtem všech použitých významových slov (všechna slova, která prošla filtrací) v lemmatizovaném tvaru stojíme před úkolem, sloučit sobě podobné tweety do skupin.

[2]Například „@lidovky“ → „@lidovka“.

4.5.1 Převod tweetu na číselný vektor

V kapitole 5 si představíme malý testovací korpus dat, získaných z Twitteru pro účely testování. Tento korpus obsahuje 8% tweetů, které obsahují jedno „důležité slovo“ (viz. kapitola 3.5.4 2x. Jelikož je toto číslo tak malé, můžeme zde použít binární reprezentaci. V porovnání s rychlostí metody TF-IDF touto volbou dojde i k značnému urychlení výpočtu.

Tento převod (znázorněný v algoritmu **??**) probíhá jako první část shlukování. Každému tweetu se alokuje pole o velikosti celkového počtu významových slov a vyplňujeme tento vektor jedničkami a nulami.

Při velkém počtu zpracovávaných tweetů je tento vektor velice řídký a většinou se skládá z řady nul, pak několik jedniček (tolik, kolik má tweet významových slov) a pak opět zbytek nul. Jedině pokud nalezneme jedničku někde v prostoru nul, pak to znamená, že v tweetu bylo použito slovo, které už použil jiný tweet a tedy že existuje nějaká podobnost.

V kapitole 2.2 byla představena důležitost hashtagu a tak pro tyto „klíčová slova“ vytvoříme následující možnost prioritizace. Při zpracovávání hashtagu vyplníme do vektoru místo jedničky nějaké číslo větší. Čím vyšší číslo, tím vyšší má hashtag prioritu. Tím můžeme docílit například toho, že při výskytu dvou tweetů, které mají společný pouze a jen tento jeden hashtag, vznikne shluk obsahující právě tyto dva tweety. To může být jev žádaný, ale také nemusí. Například hashtag „#Praha“ může být použitý v souvislosti s odložením otevření tunelu Blanka, ale zároveň tak označují tweety turisté, kteří zrovna toto město navštívili. Takto vytvořený shluk by sice poskytoval informační hodnotu, že se mluví o Praze v té a té frekvenci (například v porovnání s předchozím časovým úsekem), ale to nemusí být právě ten výsledek, který bychom chtěli. Nastavení této priority tedy záleží na osobních důvodech použití našeho programu.

Pro představu použijeme následující příklad:

Na následujícím obrázku 4.3 máme dva tweety, které byly odeslány 14. 4. 2015 dopoledne.

Oba se týkají totožného tématu a oba se nám vyskytly ve stejném časovém úseku. Pokud bychom pracovali pouze s těmito dvěma texty, dostali bychom následující pole všech slov v lemmatizovaném tvaru:

Obrázek 4.3: Dva podobné tweety pro znázornění shlukování.

`{zadržený, muž, pokusit, utéct, policist, zkolabovat, zemřít, zásah, policejní, eskorta, útěk}`

Budeme-li se nyní držet algoritmu, převedeme první tweet na vektor:

`{1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0}`

a druhý tweet bude reprezentovat vektor:

`{0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1}`.

4.5.2 Výpočet vzdálenosti

Tyto dva vektory dosadíme do vzorce kosínové vzdálenosti a dostaneme tak následující výpočet:

$$distance = \frac{A \times B}{||A|| \cdot ||B||} = \frac{4}{\sqrt{7} \cdot \sqrt{8}} \doteq 0.53$$

4.5.3 Vytvoření shluku

Tato vzdálenost je následně porovnána s naší prahovou konstantou T a je-li větší[3], pak vytváříme shluk, do kterého tyto dva tweety řadíme.

[3]Vzdálenost rovna nule znamená žádnou podobnost a jedna znamená totožnost

4.6 Prezentace výsledků

Nyní máme tweety rozdělené do skupin, které umíme pojmenovat (viz. kapitola 3.5.4). Pokud zvolíme jeden konkrétní tweet ze skupiny za reprezentanta celé skupiny, získáme tím strohý popis celé skupiny. Strohost je totiž základní vlastnost každého uživatele Twitteru[4].

Jak tyto výsledné shluky pak seřadíme záleží na konkrétním využití. V našem případě nás předně zajímají události, o kterých se mluví nejvíce. Tohoto zobrazení docílíme tím, že budeme preferovat skupiny obsahující největší počet tweetů. Tím odsuneme do pozadí šum, který se nám může do výsledků dostat a který se bude typicky skládat ze skupin, obsahující jen pár tweetů.

[4]Opět zde narážíme na omezení délky textu na 140 znaků.

5 Experimentální ověření

V této kapitole podrobíme nově vzniklý program několika experimentům, ve kterých budeme hledat ideální nastavení pro potřeby zpravodajské služby a zároveň budeme sledovat vlastnosti (rychlost, přesnost, úplnost) systému. Těmto vlastnostem je věnována kapitola 6.

5.1 Korpus

Pro všechny níže popsané pokusy byl použit korpus, který vznikl sledováním UserListů (popsáno v kapitole 4.1.4) a který nalezneme na přiloženém nosiči. Stažená data z období 13. 4. 2015 15:00 - 15. 4. 2015 15:00 čítají 15 856 tweetů.

5.2 Manuální detekce událostí v korpusu

Z korpusu vytvoříme seznam událostí, které se v textu objeví. Zde ale narážíme na problém, že již z principu Twitteru[1] není z textu vždy poznat, zda se jedná o událost. Například tweet pana Petra Mosese z 13. 4. (na obrázku 5.1) na první pohled poskytne dojem, že se něco děje ohledně Ukrajiny, imigrace a Evropské unie. To bylo v této době téma aktuální a velice sledované. Až po navštívení zkrácené URL adresy ale zjišťujeme, že je tento příspěvek o jedné epizodě interaktivního televizního pořadu Hyde park „70 let od smrti Adolfa Hitlera“.

Obrázek 5.1: Tweet obsahující zavádějící událost.

Hodnotíme-li pouze relativně aktuální události, nesoucí místo a čas, získáváme například:

[1] Příčinou je opět omezení délky textu na 140 znaků.

{Odchod do důchodu - změna, Úmrtí Güntera Grasse, Vladimír Růžička - úplatek, „Putinovi motorkáři“, Páté místo Honzy Pilaře v rally, Bobby McFerrin - koncert, Bitcoin, Soutěž o nejkrásnější dítě, Obnova těžby zlata v Česku, Nová kybernetická zbraň v Číně, ...}

Abychom tyto události získali i mezi výsledky z programu, musí být zmíněné vícekrát než jednou. To se často ale nesetkává s realitou (například koncert Bobbyho McFerrina, který je naplánován na 7. 6. 2015 v Praze, najdeme v námi nalezených tweetech pouze jednou; totéž se zmínkou o nové kybernetické zbrani Číny, atd.) a tak tyto události vyhodnocujeme jako nediskutované a řadíme je v tomto časovém úseku mezi nedůležité události, se kterými nebudeme nadále počítat.

Pro zjednodušení vybereme jen ty nejdůležitější - tedy ty, které mají v textu více zmínek než 6. Dostáváme tak ze vzorku dat následující výčet událostí[2]. Další události již nemají statisticky dostatečný počet zmínek.

Událost	**Počet tweetů**
Kauza Vladimíra Růžičky	258
Miloš Zeman a jeho kauza Peroutkových spisů	182
Pokus o odvolání pražské primátorky	156
Kauza Baroš&Berbr	52
Nové Mapy.cz	50
Apple Watch	32
Motorkářská skupina ”Noční vlci”	32
Pokuta pro Google	30
Muž utekl z policejního auta a zemřel	29
Kniha roku „Magnesia Litera“	28
Kandidatura Hillary Clinton na prezidentku	28
Kauza Jany Nagyové	25
Policista zaklekl muže, který šel na červenou	25
Vojenské operace v Jemenu	22
Utonutí 400 uprchlíků v Libyi	17
Policejní zásah v sídlech ROP	15
Zeman vyznamená studenta Petra vejvodu	15
Kauza Marka Půčky - „Smíchovský řidič“	12
Britové zahnali ruské bombardéry	11

[2]Zde narážíme na zajímavou možnost rozšíření a to sledování výskytu hashtagů. V tomto případě bychom narazili na hashtag #ZkažNázevFilmuHitlerem, který se během těchto sledovaných dvou dní vyskytl 307x. Nejedná se ale o událost, protože nezaujímá čas a místo.

Pohřešovaná 13ti letá holčička	11
Putin nejvlivnější osobností světa	10
Stažení tanků z východní Ukrajiny	9
Americký policista zastřelil omylem černocha	9
Islámisté znásilnili 9ti letou jezídku	8
Marco Rubio ohlásil kandidaturu do Bílého domu	8
Zemřel zpěvák Percy Sledge	8
Soud propustil strážníka, který zastřelil 2 členy ochranky	7
Islámský stát ztratil v Iráku již 1/4 území	7

5.3 Určení prahové konstanty T

Dále spouštíme program s těmito vstupními daty a s různými parametry a snažíme se najít tyto parametry takové, abychom dostali co nejbližší až stejný seznam, jako byl vytvořen manuálně. Naším algoritmem je ale prakticky nereálné strojově slučovat příspěvky, nemající jediné společné slovo (viz. níže). Musíme tedy počítat s tím, že námi získané výsledky nebudou nikdy stoprocentní.

- „PŘEHLEDNĚ: Otazníky v kauze Růžička. Kde jsou peníze? A jak z toho ven?“
- „Trénink u starýho Růžičky. Táto, dej si prachy do roličky, zpívá Ruda z Ostravy | iSport.cz“
- „Odvedení pozornosti od hry hokejového nároďáku done. Růža může děkovat. Kdo ale tehdy mohl tušit že bude repre koučem @tondablanik @BlanikZ“

S parametry začínáme na následujících hodnotách:

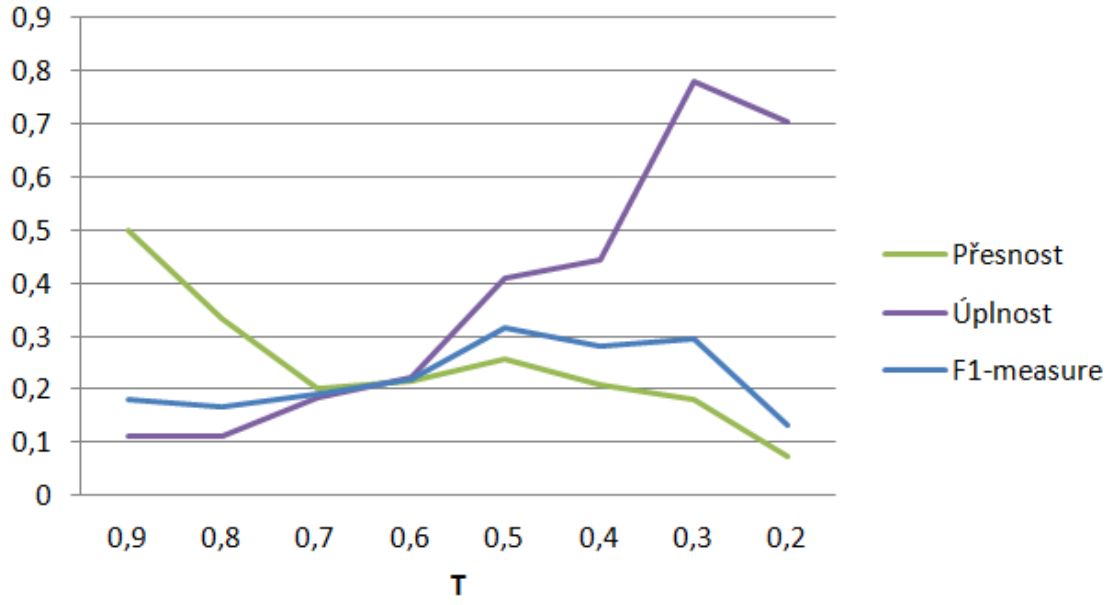
Časový interval DT	1h
Prahová konstanta T	0,9
Hodnota hashtagu pro převod textu na vektor	2
Minimální velikost shluku	4

A budeme měnit hodnotu T po desetinách až k 0,2. Je logické, že pro vysokou hodnotu T budou výsledné události velice malé (převážně 2 tweety).

Přesnost takových pokusů se bude na svém maximu a úplnost na svém minimu. Při T blíže nule bude zase shluková analýza „benevolentnější“ a bude tak shlukovat i tweety, které nijak nesouvisí, čímž docílíme mnoha velkých (špatně definovaných) událostí. Přesnost tím klesne k 0% a úplnost poroste.

Minimální velikost shluku zde použijeme jako ochranou pomůcku proti malým „nevýznamovým“ událostem a snížení přesnosti. Ve výsledcích nalézáme velké množství událostí tvořených dvěma či třema tweety a bohužel jsou mezi nimi i události pro nás podstatené. Většina z nich jsou ale soukromé události, „duplikáty“, reklama, atd. a jejich započtení do výsledků by značně ovlivnilo ukazatel přesnosti.

Výsledek pokusu znázorníme grafem na obrázku 5.2.



Obrázek 5.2: Graf přesnosti, úplnosti a F_1-measure pro $DT = 1$.

Nejzajímavější je hodnota pro $T = 0,5$, kde ukazatel F_1-measure dosahuje svého maxima, což je $31,4\%$.

Z grafu je vidět, že až k hodnotě $T = 0,3$ úplnost roste. Její zlom je dán především faktem, že vzdálenost $0,2$ a menší již nalézáme i mezi nesouvislými tweety a tak vznikající události nejsou definovány tweety stejného tématu a nejsou tedy započteny jako událost.

Mezi takový shluk můžeme řadit například událost:

[4] Ruské bombardéry poletovaly u Británie, zahnaly je až stíhačky RAF prostřednictvím @iDNEScz { Jak tady vysvětlují slovenské úřady, za víza pro ruský gang motorkářů zodpovídá vstupní země, tedy Polsko., Rusko varovalo před internetovými vtípky. A hrozí za ně trestem: Úřad, který dohlíží na ruský internet, varoval... #TechnetczInternet, Ruské bombardéry poletovaly u Británie, zahnaly je až stíhačky RAF prostřednictvím @iDNEScz, Každá piata pesnička v našom rádiu má byt' slovenská }

V pokusech narážíme na zajímavý fenomén Twitteru a to tweety s hashtagem #ZkažNázevFilmuHitlerem. Takových je ve vzorových datech celkem 307 a vytvářejí často nejobsáhlejší události ve výpisech programu. Pro jejich eliminaci bychom mohli odebrat ze sledovaných uživatelů ty, kteří tento hashtag použijí. Tím se přestane vyskytovat v analyzovaných datech a tedy i ve výsledcích. Do budoucna je doporučená manuální úprava uživatelských seznamů a jejich neustálá údržba.

Další možností, jak se těchto nevyžádaných tweetů vyvarovat, je změna hodnoty hashtagu zpět na jedničku. Tím bychom ale nevyužili plný potenciál a tak upřednostníme cestu vyšší chybovosti nad nižší úplností[3].

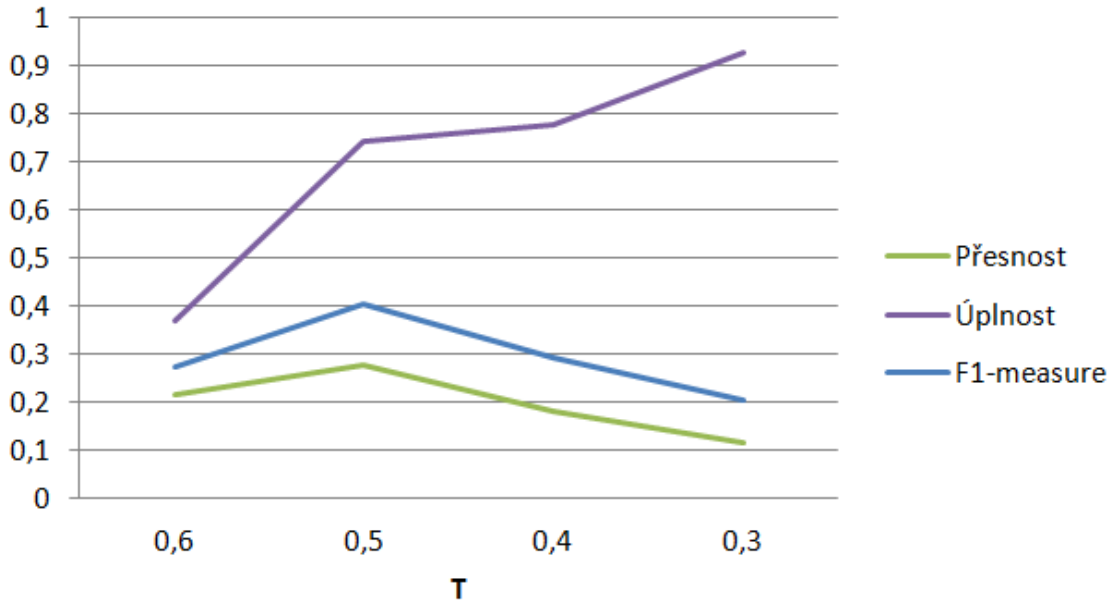
Dále si z pokusů můžeme všimnout, že se hodnota F_1-measure pohybuje ve velmi nízkých hodnotách. To si vysvětlujeme především rychlostí uživatelů a souhrou jejich reakcí. V textu totiž nemáme žádnou událost, která by „otřásla“ českým Twitterem natolik, aby byla zaznamenatelná v jedné hodině. Některé události v textu obsažené jsou sice objemné a mezi uživateli oblíbené, nejsou ale tzv. „informační bombou“, aby si o tom v jeden okamžik začalo psát více lidí. Touto úvahou docházíme k závěru, že je třeba změnit i velikost časového intervalu.

Dále pokračujeme s pokusy pro hodnoty $DT = 3$. Z předchozího pokusu je vidět, že zanedbat můžeme hodnoty T vyšší než $0,5$ a nižší než $0,3$. Hodnoty $T > 0,5$ totiž slučují převážně retweety a „duplikáty“, popsané v kapitole 4.1.1. Pro hodnoty $T < 0,3$ byl důvod neúspěchu popsán v předchozím textu.

Získáváme výsledky zobrazené grafem na obrázku 5.3.

V tomto případě dosahujeme nejlepších výsledků pro hodnotu $T = 0,5$. Přesnost $P = 27,7\%$, úplnost $R = 74\%$ a ukazatel F_1-measure $= 40,4\%$. Zkusíme tedy ještě vliv velikosti časového úseku a jako poslední pokus se

[3] V kapitole 6.2 bude podrobněji rozepsáno proč.



Obrázek 5.3: Graf přesnosti, úplnosti a F_1-measure pro $DT = 3$.

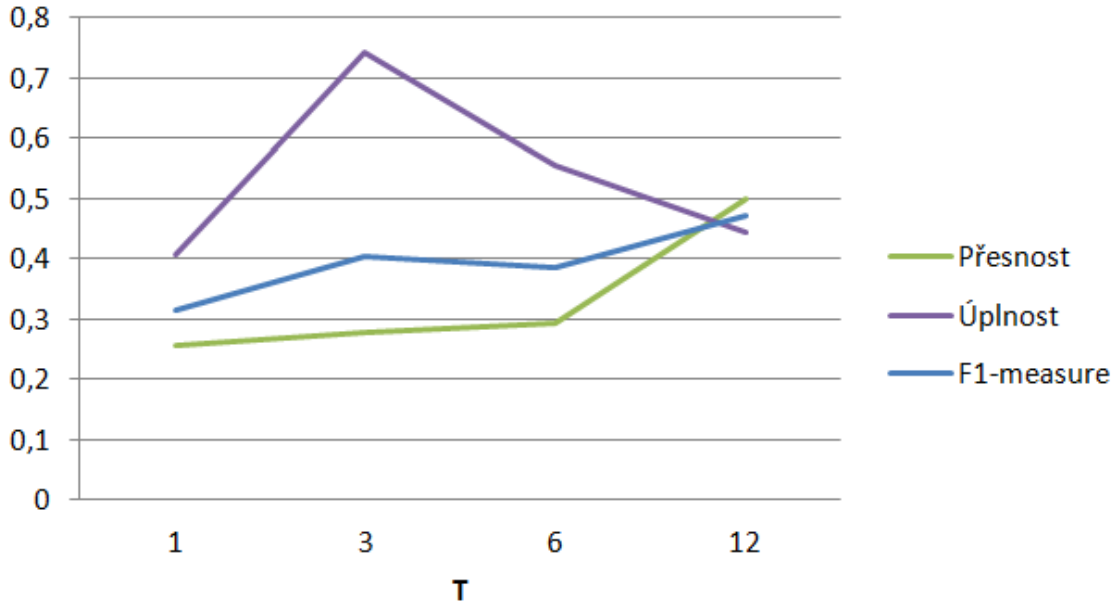
pokusíme analyzovat data pro $DT = 12$.

S vyšším počtem tweetů stoupá i počet výsledných událostí. Zvýšíme proto omezení minimální velikosti shluku na 6.

Takto velký časový úsek již znamená mnoho tweetů ke zpracování a proto i větší nároky na paměť počítače[4]. Běh programu trval 1 hodinu, 10 minut a 16 vteřin a výsledky jsou ze všech experimentů nejuspokojivější. Dosahujeme přesnosti 50% a úplnosti 44,4%. Z těchto hodnot vychází ukazatel F_1-measure na 47%.

Pro porovnání vytvoříme graf kvality výsledků v závislosti na velikosti DT při nastavení $T = 0,5$. Tento graf je znázorněn obrázkem 5.4 a můžeme z něj vypozorovat, že čim větší časový úsek nastavíme, tím přesnější výsledky získáváme. Bohužel díky limitaci zdrojů nejsme schopni určit, v jakém bodě se křivka F_1-measure v grafu zlomí v klesající.

[4]řádově stovky MB



Obrázek 5.4: Graf přesnosti, úplnosti a F_1-measure pro $T = 0,5$.

5.3.1 Chyby ve výsledcích

V těchto experimentech můžeme pozorovat, že i přes různé změny nastavení zůstávají ve výsledcích chyby, jako jsou soukromé konverzace (například událost „@Posledniskaut No to se mi snad zdá! @beneslenka“). Těmto událostem by šlo snadno předejít jednoduchým odebráním zúčastněných osob ze sledovaných seznamů. Jinak tomu je ale u příspěvků typu „dobré ráno“, které už tak snadno nevyfiltrujeme.

Dále tyto výsledky vykazují zajímavé změny oproti manuálně nalezenému seznamu událostí. Například události ze světa nám vzdálené nebo firmy s dobrou IT základnou své „události“ šíří Twittrem jednotnými texty, které další uživatelé jen retweetnou, aniž by je nějak pozměnili. Tyto události dokážeme snadno sjednotit a vyhodnotit jejich důležitost. Naopak události, které se lidí přímo týkají a které v nich vzbuzují nějaké emoce, již lidé vyjadřují vlastními slovy. Právě tyto události jsou často nejrozsáhlejší, pro nás nejdůležitější a také nejhůře detekovatelné.

5.4 Spuštění programu bez lemmatizátoru

Nyní máme přehled o vhodném nastavení systému a o jeho vlastnostech. Pokusíme se tedy využít nejlepší nalezené parametry k ověření přínosu lemmatizátoru. Spouštíme program s nastavením:

Časový interval DT	12h
Prahová konstanta T	0,5
Hodnota hashtagu pro převod textu na vektor	2
Minimální velikost shluku	6

A z algoritmu programu vynecháme lemmatizátor. Data tedy jen filtrujeme a hned na to shlukujeme.

V následující tabulce vypíšeme výsledky pokusu:

Tabulka 5.1: Porovnání výsledků programu s lemmatizací a bez.

	S lemmatizátorem	**Bez lemmatizátoru**
Doba běhu	1h 23min 7s	28min 46s
Přesnost	50%	30,3%
Úplnost	44,4%	37%
F_1**-measure**	47%	33,3%

Tím máme k dispozici ukazatel užitečnosti lemmatizátoru.

6 Vyhodnocení výsledků

V předchozích kapitolách jsme vytvořili systém, který sleduje trendy na sociální síti Twitter. Nyní se zaměříme na získané výsledky, na rychlost jejich získání, jejich přesnost a úplnost.

6.1 Rychlost zpracování

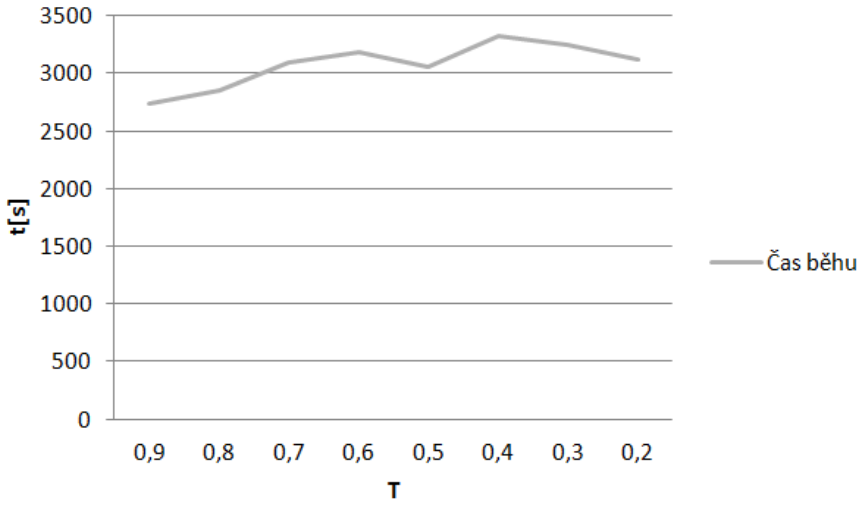
K rychlosti zpracování se úzce váže rychlost lemmatizátoru. Pokud zpracováváme rámcově tisíce tweetů, je tzv. „úzkým hrdlem“ právě lemmatizátor. Pro nás je to ale proces velice důležitý a bez něj získáváme jen zlomky výsledků.

Při spuštění programu na desetitisíce tweetů si už můžeme všimnout změny, kdy lemmatizátor stále pracuje se složitostí $O(n)$, ale proces shlukování již počítá vzdálenosti mezi každou dvojicí a jeho složitost $O(n^2)$ pak tento proces znatelně zpomalí. Také paměťová náročnost stoupá exponenciálně, protože každý tweet je reprezentován polem celých čísel, jehož velikost je ovlivněna celkovým počtem zpracovávaných slov v daném časovém úseku. To může ztížit zpracování dat například z jednoho týdne a byla by třeba optimalizace kódu nebo prostředí s vyšším výkonem, než má osobní počítač.

Rychlost zpracování je přímo úměrná velikosti aktuální množiny tweetů a tedy i velikosti časového intervalu, na který vstupní data dělíme. Pokud budeme zpracovávat data po hodině, získáme za tuto dobu méně textů než například za den a další zpracování bude rychlejší. I tak ale musíme počítat s tím, že vzorkujeme-li data po hodině, výsledky dostaneme například po hodině a 5ti minutách. Konkrétní velikost zpoždění je dána počtem tweetů a toto číslo zase ovlivňují další faktory, jako je například denní doba, průběh sportovních událostí, politická aktivita ve světě apod.

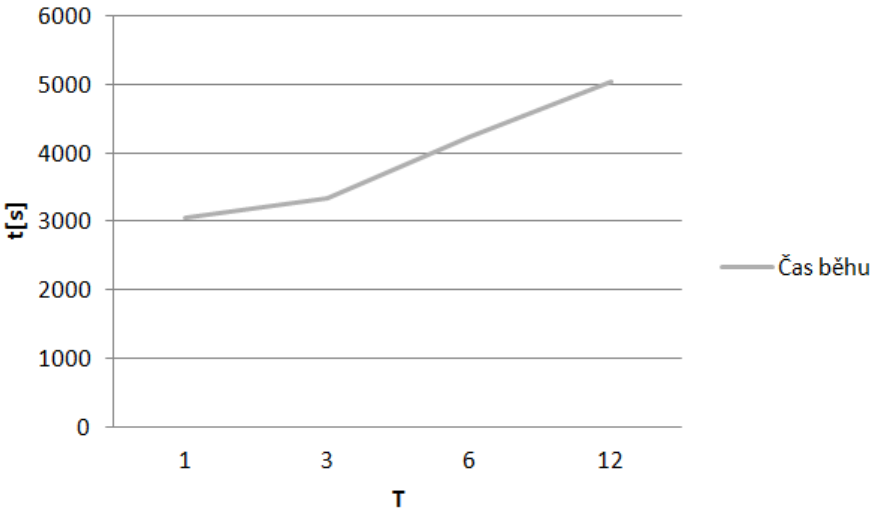
Z experimentu z kapitoly 5 zde uvedeme přehled časů, potřebných ke zpracování konkrétních dat. Tyto časy závisí na konkrétních datech a rychlosti výpočetního prostředí, proto jsou zde uvedeny jen pro znázornění závislosti rychlosti na nastavení.

Z obrázku 6.1 je vidět, že při změnách jen v hodnotách T dochází v době



Obrázek 6.1: Graf doby běhu pro $DT = 1$.

zpracování k minimálním změnám. Oproti tomu pokud změníme hodnotu DT, musíme pracovat s většími datovými strukturami a i když ve výsledku zpracujeme stejné množství dat, čas běhu programu se liší. To je znázorněno na obrázku 6.2.



Obrázek 6.2: Graf doby běhu pro $T = 0,5$.

6.2 Přesnost P

V této kapitole se zaměřujeme na část výsledků, která je určena správně. Tedy na takové události obsažené ve výsledcích, které byly nalezeny i manuálně a jejich poměr k celkovým výsledkům. Pro výpočet přesnosti slouží následující vzorec:

$$P = \frac{tp}{tp + fp},$$

kde tp jsou události určené správně a fp jsou události, určené špatně (tedy chyby ve výsledku).

V této kapitole je třeba zmínit účel vytvářeného programu. Tím je poskytování informací zpravodajské službě o potenciálních událostech, které budou po jejich detekci verifikovány a dál zkoumány manuálně právě touto zpravodajskou službou. Proto lze říci, že chybovost[1] není v našem případě striktně vzato chyba. I taková skupina tweetů, u kterých se nakonec zjistí, že se žádné události netýkají, má pro nás informační hodnotu a může (ale taky nemusí) být užitečná. Výpočet a hodnotu přesnosti výsledků ale i tak spočítáme.

Pro spočítání přesnosti použijeme data získaná experimentem v kapitole 5. Při použití nejlepšího možného nastavení nastavení jsme získali 24 výsledných událostí, z nichž je 12 určeno mimo náš ručně vytvořený seznam.

Přesnost výsledků programu, který sleduje trendy na sociální síti Twitter, je tedy 50%.

6.3 Úplnost R

Zde počítáme tweety, které jsme našli manuálním procesem, ale chybí nám ve výsledcích programu; respektive jejich poměr. Řešíme tedy otázku, jak moc úplné jsou výsledky programu. Pro výpočet slouží následující vzorec:

$$R = \frac{tp}{tp + fn},$$

[1] Chybovost = 1 - Přesnost

kde tp jsou události určené správně a fn jsou události, které nebyly určené vůbec. Tedy takové události, které v textu jsou, ale v našich výsledcích se nevyskytují.

Opět se zde opřeme o experiment z kapitoly 5, kde bylo manuálně zjištěno ze vstupních dat, kolik událostí tato data obsahují. Opět použijeme nejlepší nalezené nastavení programu a získáváme 12 správně určených událostí. Celkem jich můžeme ve zkoumaných datech nalézt 27. Poměr těchto hodnot vyjadřue úplnost výsledků.

Úplnost výsledků programu, který sleduje trendy na sociální síti Twitter je tedy 44,4%.

Takto nízké procento si vysvětlujeme především přísným nastavením minimální velikosti události, které skryje všechny shluky menší než definujeme za zajímavé (Pokud je shluk ze zanedbatelného množství tweetů, nevyhodnocujeme ho za událost). Pokud bychom definovali událost jako shluk dvou a více tweetů, ve výsledku by se začaly objevovat „duplikáty“ (viz. kapitola 4.1.1) a další nezajímavé informace (např. osobní rozhovory).

Dalším důvodem je fakt, že si lidé často píší o událostech vlastními slovy a jak bylo zobrazeno na začátku experimentu v kapitole 5.3, nemusí mít tato vyjádření jediný společný výraz. Tím například došlo k tomu, že události o kauze Růžička, či problémy pana prezidenta Miloše Zemana s Peroutkovými texty nejsou detekovány v plném objemu, ale jen jako nedůležité či okrajové události.

Úplnost jde ruku v ruce s chybovostí a pokud změníme omezení velikosti události na nižší hodnotu, získáme tím i více nevýznamových událostí, což by zlepšilo úplnost, ale zhoršilo přesnost.

6.4 F_1-measure

Metrika F1-measure patří obecně k nejpoužívanějším metrikám pro zjištění správnosti klasifikace. Tato metrika vyjadřuje harmonický průměr přesnosti a úplnosti, kde jsou váhy těchto dvou hodnot rovnocenné (tedy 50% pro přesnost a 50% pro úplnost). Výpočet je vyjádřen následujícím vzorcem:

$$F_1 = 2 * \frac{PR}{P + R}$$

Po dosazení do tohoto vzorce zjišťujeme, že F_1 hodnota pro náš systém, spuštěný s optimálním nastavením, je 47%.

7 Závěr

Cílem této práce bylo vytvořit program pro Českou tiskovou kancelář, který bude sledovat sociální síť Twitter a extrahovat z ní události v reálném čase.

Toto téma bylo rozděleno na několik dílčích podproblémů, ze kterých bylo složeno výsledné řešení. Mezi těmito podproblémy jsou: problém stahování tweetů (kapitola 4.1), problém filtrace (kapitola 4.3) a problém shlukování (kapitola 4.5). Každé dílčí řešení značně ovlivní kvalitu výsledků.

Během vývoje jsem byl nucen se několikrát vracet k prvotnímu kroku práce - získání vstupních dat. Twitter neposkytuje zdarma žádné nástroje pro stažení námi potřebného objemu tweetů a proto vzniklo několik alternativních metod, pomocí kterých by byl tento počet naplněn. Po důkladném porovnání těchto verzí a po konzultaci v České tiskové kanceláři byla nakonec vybrána metoda, která stahuje tweety z Twitter API sledováním množiny konkrétních uživatelů.

Po vytvoření zadaného systému byla velká část práce věnována testování a nalezení nejvhodnějšího nastavení programu. To vše experimentálními metodami. Při samotném testování jsem narazil na problémy s aktivitou uživatelů a tedy na malou velikost výsledků.

Testy ukázaly, že s vhodnym nastavením dosahují výsledky přesnosti 50%, úplnosti 44,4% a F_1-measure 47%.

Jelikož je Twitter v České republice relativně nepoužívaný, ale velice rychle expandující, je toto téma stále plné potenciálních studií. Tuto práci je dále možno rozšířit o práci s URL adresami, obrázky nebo například nahrazením lemmatizátoru za nástroj, který by extrahoval z jednotlivých slov jeho kořen - tzv. stemming. Dále se nabízí i práce s hashtagy, daty či další vylepšení shlukování. Velmi potřebným nástrojem pro tuto práci by byl i algoritmus pro výběr „zajímavých lidí“.

Práce již byla v dubnu 2015 prezentována v České tiskové kanceláři a je dále plánována k uplatnění v praxi.

Přehled použitých termínů a zkratek

API	Application Programming Interface je rozhraní pro počítačovou komunikaci.
Geotag	Nepovinný atribut tweetu, obsahující GPS souřadnice místa odeslání.
Tweet	Textová zpráva omezená na 140 znaků, která obsahuje kromě textu ještě další atributy (např. podpis autora, čas vytvoření, jazyk, nepovinně i geotag, ...)
Shlukování	Proces seskupování objektů do skupin na základě jejich vzájemné podobnosti.
HTTP	Hypertext Transfer Protocol je bezstavový komunikační protokol pro přenos dat na internetu.
Timeline	Časová osa uživatele, kde jsou zveřejněny jeho tweety spolu s tweety osob, které sleduje a které se o něm zmiňují.
Retweet	Zveřejnění cizího tweetu na vlastní Timeline. V textu tweetu se pak vyskytuje zkratka „RT“.
Stream	Proud tweetů, které plní Twitter aktuálními daty a nabízí je tak vývojářům.
Firehose	Plný přístup k datům Twitteru ve formě placeného streamu s tweety.
XML	Extensible Markup Language - značkovací jazyk určen především pro sdílení dat.
JSON	JavaScript Object Notation je způsob zápisu dat. Vytvořen jako nástupce formátu XML.

Literatura

[abo(2015)] *About Twitter* [online]. 2015. [cit. 22.4.2015]. Dostupné z: https://about.twitter.com/company.

[bbc(2014)] *#BBCtrending: The petition to block the Samaritans' Twitter app* [online]. 2014. [cit. 26.4.2015]. Dostupné z: http://www.bbc.com/news/blogs-trending-29881099.

[cou(2015)] *Get real Twitter followers and boost your influence* [online]. 2015. [cit. 24.4.2015]. Dostupné z: http://twittercounter.com/pages/featured.

[dat(2015)] *Twitter Ends its Partnership with DataSift* [online]. 2015. [cit. 24.4.2015]. Dostupné z: http://blog.datasift.com/2015/04/11/twitter-ends-its-partnership-with-datasift-firehose-access-expires-on-au\\gust-13-2015/.

[for(2014)] *Top Twitter Trends: What Countries Are Most Active? Who's Most Popular?* [online]. 2014. [cit. 24.4.2015]. Dostupné z: http://www.forbes.com/sites/victorlipman/2014/05/24/top-twitter-trends-what-countries-are-most-active-whos-most-popular/.

[geo(2013)] *The geography of Tweets* [online]. 2013. [cit. 24.4.2015]. Dostupné z: https://blog.twitter.com/2013/the-geography-of-tweets.

[gni(2013)] *4 Things You Need To Know About Migrating to Version 1.1 of the Twitter API* [online]. 2013. [cit. 24.4.2015]. Dostupné z: https://blog.gnip.com/migrating-version1-1-twitter-api/.

[liv(2015)] *Twitter Usage Statistics* [online]. 2015. [cit. 22.4.2015]. Dostupné z: http://www.internetlivestats.com/twitter-statistics/.

[nin(2015)] *Nine years and counting* [online]. 2015. [cit. 22.4.2015]. Dostupné z: https://blog.twitter.com/2015/nine-years-and-counting.

[pol(2014)] *Developer Agreement & Policy* [online]. 2014. [cit. 26.4.2015]. Dostupné z: https://dev.twitter.com/overview/terms/agreement-and-policy#6._Be_a_Good_Partner_to_Twitter.

[rea(2013)] *The Real History of Twitter, In Brief* [online]. 2013. [cit. 22.4.2015]. Dostupné z: http://twitter.about.com/od/Twitter-Basics/a/The-Real-History-Of-Twitter-In-Brief.htm.

[soc(2015)] *Twitter Profiles* [online]. 2015. [cit. 24.4.2015]. Dostupné z: http://www.socialbakers.com/statistics/twitter/profiles/.

[sys(2014)] *Inside Twitter by Sysomos* [online]. 2014. [cit. 24.4.2015]. Dostupné z: http://sysomos.com/uploads/Inside-Twitter-BySysomos.pdf.

[Benevenuto et al.(2010)Benevenuto, Magno, Rodrigues„ Almeida] BENEVENUTO, F. et al. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, 6, s. 12, 2010.

[Busta(2015)] BUSTA, D. Facebook v boji proti sebevraždám. *EkonTech*. 2015, 17.číslo.

[Hajič et al.(1996)Hajič, Hajičová„ Rosen] HAJIČ, J. – HAJIČOVÁ, E. – ROSEN, A. Formal Representation of Language Structures. *TELRI Newsletter*. 1996, , 3, s. 12–19.

[Hrala(2012)] HRALA, M. Automatická klasifikace dokumentů s podobným obsahem [online]. Master's thesis, University of West Bohemia, Faculty of Applied Sciences, 2012. Dostupné z: DostupnÃľzWWW<http://theses.cz/id/7satjf/>.

[Li et al.(2012)Li, Sun„ Datta] LI, C. – SUN, A. – DATTA, A. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, s. 155–164. ACM, 2012.

[McMinn et al.(2013)McMinn, Moshfeghi„ Jose] MCMINN, A. J. – MOSHFEGHI, Y. – JOSE, J. M. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, s. 409–418. ACM, 2013.

[Sakaki et al.(2010)Sakaki, Okazaki,, Matsuo] SAKAKI, T. – OKAZAKI, M. – MATSUO, Y. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, s. 851–860. ACM, 2010.

[Salton – Buckley(1988)Salton, Buckley] SALTON, G. – BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information processing & management*. 1988, 24, 5, s. 513–523.

Přílohy

První polovina výpisu programu pro $DT = 6h$ **a** $T = 0,5$ nad testovanými daty. Události jsou vypisovány ve tvaru [velikost události] název shluku {obsah skupiny}

5 Z fronty na východní Ukrajině by měly být odsunuty tanky #Ukrajina {Z fronty na východní Ukrajině mají být odsunuty tanky: Ministři zahraničí Německa, Francie, Ruska a Ukrajiny se dohodli na dalším. . . , Z fronty na východní Ukrajině by měly být odsunuty tanky #Ukrajina , Z fronty na východní Ukrajině by měly být odsunuty tanky #Ukrajina , Z fronty na východní Ukrajině mají zmizet tanky, domluvili se ministři: Ministři zahraničí Německa, Francie, Ruska a. . . #ZprávyZahraniční, Z fronty na východní Ukrajině mají zmizet tanky, domluvili se ministři}

4 Ištvan a Šlachta nás stáli miliony, ať skončí! žádá Benda { Ištvan se Šlachtou nás stáli dva miliony. Měli by odejít, žádá Benda Echo24.cz via @echo24cz , Ištvan a Šlachta nás stáli dva miliony! Měli by odejít, žádá Benda , RT @echo24cz: Ištvan se Šlachtou nás stáli dva miliony. Měli by odejít, žádá Benda Echo24.cz via @echo24cz , Ištvan a Šlachta nás stáli miliony, ať skončí! žádá Benda }

4 Jednoho muže zastřelil, druhého zranil. Soud strážníka pustil z vazby: Okresní soud v Hradci Králové propusti z vazby. . . #ZprávyDomov { Jednoho muže zastřelil, druhého zranil. Soud strážníka pustil z vazby: Okresní soud v Hradci Králové propustil z vazby. . . #ZprávyDomov, Jednoho muže zastřelil, druhého zranil. Soud strážníka pustil z vazby , Jednoho muže zastřelil, druhého zranil. Soud strážníka pustil z vazby , Soud pustil z vazby strážníka obviněného ze střelby ve Vysokém Mýtě: Okresní soud v Hradci Králové v úterý propustil z vazby strážníka,. . . }

4 #IIHFCZE Češi do toho ! ! ! { #IIHFCZE Jedeme hoši jedeme, #IIHFCZE Kdo nefandi není Čech. HoP HoP HoP, #IIHFCZE Češi do

toho ! ! !, #IIHFCZE Na tento zápas se moc těším a držím palce. Češi do toho !}

4 Růžička má pořádný problém. Čelí trestnímu oznámení kvůli korupci. { Růžička má pořádný problém. Čelí trestnímu oznámení kvůli korupci. , Další rána pro Růžičku. Čelí trestnímu oznámení kvůli korupci , Hokejový trenér Růžička čelí trestnímu oznámení, převzal prý úplatek. , Hokejový trenér Růžička čelí trestnímu oznámení, převzal prý úplatek: Trenér české hokejové reprezentace Vladimír Růžička... #ZprávyDomácí}

5 Krnáčová zůstává primátorkou, nikoliv nečekaně. #praha {Krnáčová zůstává primátorkou, nikoliv nečekaně. #praha, RT @Radiozurnal1: Adriana Krnáčová zůstává primátorkou Prahy, zastupitelé nebudou o jejím odvolání hlasovat. #praha #politika, #Praha: Krnáčová primátorkou zůstane, o odvolání se hlasovat nebude., Krnáčová zůstává primátorkou #Praha, o odvolání se ani nehlasovalo. , RT @Aktualnecz: Krnáčová zůstává primátorkou #Praha, o odvolání se ani nehlasovalo.}

4 Děti, zvládne máma řídit autobus? Firma rozjíždí v Teplicích neobvyklou kampaň. { Děti, zvládne máma řídit autobus? Firma rozjíždí v Teplicích neobvyklou kampaň. , RT @michal_pavec: Děti, zvládne máma řídit autobus? Firma rozjíždí v Teplicích neobvyklou kampaň. , RT @michal_pavec: Děti, zvládne máma řídit autobus? Firma rozjíždí v Teplicích neobvyklou kampaň. , RT @lidovky: Děti, zvládne máma řídit autobus? Firmě chybí řidiči, za volant láká ženy }

8 Pražské TOP 09 a ODS se nepodařilo odvolat primátorku Krnáčovou: Pražská primátorka Adriana Krnáčová (za ANO) zůstává ve funkci. Opozičním... { TOP 09 a ODS se nepodařilo odvolat primátorku Prahy Adrianu Krnáčovou z ANO. Nedokázaly to kvůli nedostatku hlasů dostat na program jednání., RT @PepaKopecky: TOP 09 a ODS se nepodařilo odvolat primátorku Prahy Adrianu Krnáčovou z ANO. Nedokázaly to kvůli nedostatku hlasů dostat n..., Krnáčová zůstává. Zastupitelům TOP 09 a ODS se primátorku nepodařilo odvolat , Krnáčová zůstává. Zastupitelům TOP 09 a ODS se primátorku nepodařilo odvolat , Zastupitelům TOP 09 a ODS se nepodařilo odvolat primátorku Prahy Krnáčovou @adkrn, bod nedostali na program jednání: , RT @zpravyrozhlascz: Zastupitelům TOP 09 a ODS se nepodařilo odvolat primátorku Prahy Krnáčovou @adkrn, bod nedostali na program jednání: h..., Pražské TOP 09 a ODS se nepodařilo odvolat primátorku Krnáčovou: Pražská primátorka Adriana Krnáčová (za ANO) zůstává

ve funkci. Opozičním. . . , RT @zpravyrozhlascz: Zastupitelům TOP 09 a ODS se nepodařilo odvolat primátorku Prahy Krnáčovou @adkrn, bod nedostali na program jednání: h. . . }

4 Do La Manche vpluly ruské válečné lodě, míří na manévry v Atlantiku: Válečné lodě patřící k ruské Severní flotile v úterý. . . #ZprávyNATO { Do La Manche vpluly ruské válečné lodě, míří na manévry v Atlantiku. , Do La Manche vpluly ruské válečné lodě, míří na manévry v Atlantiku: Válečné lodě patřící k ruské Severní flotile v úterý. . . #ZprávyNATO, RT @iDNEScz: Do La Manche vpluly ruské válečné lodě, míří na manévry v Atlantiku. , Do La Manche vpluly ruské válečné lodě, míří na manévry v Atlantiku: Válečné lodě patřící k ruské Severní flotile v úterý. . . #ZprávyNATO}

4 Seznam rozšiřuje @mapy_cz na celý svět, cizina bude i k offline stažení #novemapy { Seznam rozšiřuje @mapy_cz na celý svět, cizina bude i k offline stažení #novemapy , RT @Lupacz: Seznam rozšiřuje @mapy_cz na celý svět, cizina bude i k offline stažení #novemapy , RT @Lupacz: Seznam rozšiřuje @mapy_cz na celý svět, cizina bude i k offline stažení #novemapy , RT @Lupacz: Seznam rozšiřuje @mapy_cz na celý svět, cizina bude i k offline stažení #novemapy }

4 Policisté zasahují v kancelářích ROP Střední Morava: V kancelářích Regionálního operačního programu Střední Morava v Olomouci a Zlíně od. . . { V Olomouckém a Zlínském kraji zasahují policisté na pracovištích Regionálního operačního programu Střední Morava: , Policie zasahuje v sídlech Regionálního operačního programu Střední Morava v Olomouci a ve Zlíně , Policisté zasahují v kancelářích ROP Střední Morava: V kancelářích Regionálního operačního programu Střední Morava v Olomouci a Zlíně od. . . , V kancelářích ROP Střední Morava od rána zasahují policisté: V kancelářích Regionálního operačního programu Střední Morava v. . . #ZprávyDomov}

4 Jestli to trestní oznámení na Růžičku dostane k vyřízení JUDr. Ištvan, tipuju pět plus do konce zápasu. { Jestli to trestní oznámení na Růžičku dostane k vyřízení JUDr. Ištvan, tipuju pět plus do konce zápasu. , RT @jindrichsidlo: Jestli to trestní oznámení na Růžičku dostane k vyřízení JUDr. Ištvan, tipuju pět plus do konce zápasu. , RT @jindrichsidlo: Jestli to trestní oznámení na Růžičku dostane k vyřízení JUDr. Ištvan, tipuju pět plus do konce zápasu. , RT @jindrichsidlo: Jestli to trestní oznámení na Růžičku dostane k vyřízení JUDr. Ištvan, tipuju pět plus do konce zápasu. }

8 RT @JaromirBosak: Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rud... { Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rudá pěst zasahuje..., RT @JaromirBosak: Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rud..., RT @JaromirBosak: Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rud..., RT @JaromirBosak: Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rud..., RT @JaromirBosak: Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rud..., RT @JaromirBosak: Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rud..., RT @JaromirBosak: Četl jsem výzvu R,Berbra k DK ohledně potrestání M.Baroše-musel jsem se kouknout do kalendáře, že se nepíše rok 1978. Rud... }

4 DOPLNĚNÍ: Policie zasahuje v sídlech ROP Střední Morava { DOPLNĚNÍ: Policie zasahuje v sídlech ROP Střední Morava , RT @CT24zive: DOPLNĚNÍ: Policie zasahuje v sídlech ROP Střední Morava , RT @CT24zive: DOPLNĚNÍ: Policie zasahuje v sídlech ROP Střední Morava , RT @CT24zive: ČT: Policisté zasahují v sídlech ROP Střední Morava v Olomouci a ve Zlíně.}

4 RT @zdenek_john: Zatim nejlepší hláška: Růžička by měl zveřejnit ceník, at' v tom není hokej. { Vyhlašujeme soutěž o nejlepší hlášku k MS v hokeji. Zatím vede: "Růžička by měl na férovku zveřejnit ceník, at' v tom není hokej.", RT @mfdnes: Vyhlašujeme soutěž o nejlepší hlášku k MS v hokeji. Zatím vede: "Růžička by měl na férovku zveřejnit ceník, at' v tom není hokej..., RT @zdenek_john: Zatim nejlepší hláška: Růžička by měl zveřejnit ceník, at' v tom není hokej., RT @mfdnes: Vyhlašujeme soutěž o nejlepší hlášku k MS v hokeji. Zatím vede: "Růžička by měl na férovku zveřejnit ceník, at' v tom není hokej... }