VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ANALÝZA POSTOJŮ K POČÍTAČOVÝM HRÁM

SENTIMENT ANALYSIS FOR THE FIELD OF COMPUTER GAMES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **PAVEL BALAJKA**
AUTHOR

VEDOUCÍ PRÁCE — **Doc. RNDr. PAVEL SMRŽ, Ph.D.**
SUPERVISOR

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Balajka Pavel**
Obor: Informační technologie
Téma: **Analýza postojů k počítačovým hrám**
Sentiment Analysis for the Field of Computer Games
Kategorie: Algoritmy a datové struktury

Pokyny:

1. Seznamte se s přístupy a metodami počítačové analýzy postojů a se způsoby získávání dat ze sociálních sítí a dalšího, uživateli generovaného obsahu.
2. Shromážděte a průběžně aktualizujte datovou sadu, která bude sloužit k průběžnému hodnocení výsledků i jejich závěrečné prezentaci.
3. Navrhněte a implementujte systém, který dokáže indexovat a analyzovat stahovaná data.
4. Vytvořte systém pro automatickou klasifikaci shromažďovaných dat, analýzu trendů a vizualizaci výsledků.
5. Demonstrujte vytvořený systém na vhodně zvolených příkladech ze zpracované datové sady.
6. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- Manning, C. D., Schütze, H., Foundations of Statistical Natural Language Processing, MIT Press, 1999, ISBN 0-262-13360-1.

Pro udělení zápočtu za první semestr je požadováno:

- funkční prototyp řešení

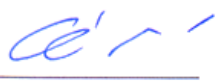
Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Smrž Pavel, doc. RNDr., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 17. května 2017

L.S.



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Práce se zabývá analýzou postojů vyjadřovaných v příspěvcích uživatelů na sociálních sítích. Popisuje obecný systém, který byl pro uvedené účely vytvořen a specializován na oblast strategických počítačových her. Hlavní důraz je kladen na problémy získávání a analýzy dat ze sociálních sítí a zobrazení získaných výsledků uživateli. Jsou zmíněny jednotlivé etapy zpracování textu jako např. tokenizace a filtrace nepotřebných slov, za účelem efektivnější analýzy názorů a rozebírány metody strojového učení jako např. Decision Trees a Naive Bayes, a jejich použití. Dále je popsán návrh uvedeného systému a jeho následná implementace s vybranými částmi a metodami. Nakonec je provedeno srovnání výsledků testů analyzátoru provedených za různých podmínek.

Abstract

The thesis deals with sentiment analysis extracted from opinions of users on social networks. It describes a general system that was created for presented purpose and specialised on the field of strategic computer games. In particular we unravel the problems of acquiring data from social networks, sentiment analysis and results presentation to the user. We mention particular ways of text processing e.g. tokenization and unnecessary word filtration, for purpose of more effective sentiment analysis and we mention machine learning methods e.g. Decision Trees and Naive Bayes, and their usage. Next we describe design of desired system and its implementation with chosen parts and methods. In the end we compare results of tests of sentiment analyzator done under various circumstances.

Klíčová slova

analýza sentimentu, počítačové hry, strojové učení, zpracování přirozeného jazyka

Keywords

sentiment analysis, computer games, machine learning, natural language processing

Citace

BALAJKA, Pavel. *Analýza postojů k počítačovým hrám.* Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Smrž Pavel.

Analýza postojů k počítačovým hrám

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. RNDr. Pavla Smrže, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Balajka
3. května 2017

Poděkování

Rád bych poděkoval svému vedoucímu této práce panu docentu Smržovi za ochotu a skvělou pomoc při tvorbě této práce.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V dnešní době lidé vyjadřují své postoje k různým tématům zejména prostřednictvím sociálních sítí, blogů, diskusních fór apod. Podíváme-li se na větší množství těchto názorů různých lidí vztahujících se k určitému tématu, můžeme se dozvědět, jak lidé obecně na dané téma reagují a jaký k němu mají vztah. Tyto informace pak mohou posloužit jako relevantní zdroj pro průzkumy veřejného mínění, jako zpětná vazba pro výrobce určitých produktů apod. Pro získání určitého objektivního přehledu je však potřeba prohlédnout větší množství příspěvků, blogů, recenzí apod., což je pro člověka časově náročná záležitost, zejména má-li prohlížet názory k více tématům. Proto je lepší nechat takovou práci na stroji, který nejen že by ji provedl mnohem rychleji, ale také, jelikož lidé na internet přidávají své názory neustále, by byl schopen prohlížet názory opakovaně.

Nejjednodušším způsobem klasifikace postojů lidí je určení, zda se jedná o pozitivní, negativní nebo neutrální postoj. Pro člověka je mnohdy jednoduché rozlišit, do jaké kategorie názor spadá. Má-li ale názor či postoj klasifikovat stroj, nejedná se o jednoduchý úkol. Takový stroj nazýváme *Analyzátor sentimentu*. Předložíme-li analyzátoru určitý názor v textové podobě, pak pro provedení správné klasifikace si musí vstupní text nejprve zpracovat do takové podoby, aby na něm byl schopný provést analýzu, a tím klasifikovat daný názor co možná nejpřesněji. Analyzátor se musí vypořádat hned s několika problémy. Jedním z nich je zpracování vstupního textu tak, aby mu stroj porozuměl, což spadá do oblasti zpracování přirozeného jazyka (používá se zkratka *NLP*, z angl. *Natural Language Processing*). Jedná se o oblast umělé inteligence zameřenou na komunikaci mezi počítači a lidmi, kterou dnes využívá mnoho aplikací a nástrojů (např. překladače, recenzní systémy apod.) Tuto technologii lze tedy využít k extrakci určitých informací z textu. Takovými informacemi mohou být i názory či postoje člověka k určitému tématu, problému či výrobku. Vývoj zpracování přirozeného jazyka, a tedy i vývoj umělé inteligence, jde stále kupředu a ani v této době na světě ještě nevznikl systém nebo robot, který by dokázal dokonale zpracovat přirozený jazyk a plně mu porozumnět.

Ve své práci budu využívat technologie zpracování přirozeného jazyka k analýze příspěvků ze sociálních sítí. Z důvodu instanciace systému na počítačové hry se bude jednat o příspěvky, které se vztahují k určitým herním titulům z okruhu strategických počítačových her. Systém lze využít i pro zpracování jiných témat, ovšem v této práci se budu věnovat výhradně herním titulům, a to zejména z oblasti strategických her. Mým cílem je vytvořit funkční systém pro analýzu názorů. Jednotlivými podcíly práce bude tedy stahování potřebných dat ze sociálních sítí, analýza názorů nad staženými daty a nakonec zobrazení statistik názorů a postojů k zadaným herním titulům, včetně samotných klasifikovaných dat uživateli. Tvorbu uvedeného systému je nutné si nejdříve rozdělit na menší podproblémy.

Bude tedy potřeba nejprve navrhnout způsob stahování a indexování požadovaných dat z různých sociálních sítí a poté jejich zpracování, analýzu a extrakci relevantních informací. Takových způsobů může existovat mnoho a bylo by dobré najít ten nejvhodnější pro tuto práci.

V následující kapitole bude pojednáváno o teoretickém rozboru problému, a to konkrétně, jaké má analýza názorů spojení s českým slovem *sentiment*, jak jej budeme chápat ve slovním spojení *Analyzátor sentimentu* a co všechno je potřeba k efektivní realizaci potřebného analyzátoru, čili z jakých částí se bude skládat. Kapitoly 3 a 4 budou pojednávat o návrhu a realizaci celého systému na základě teoretických znalostí zmíněných v předešlých kapitolách. Bude v nich také vysvětleno, jak bylo postupováno při řešení různých problémů při návrhu a implementaci systému. Před závěrem se nachází poslední kapitola 5, kde bude popsáno experimentování se systémem a následné srovnání dosažených výsledků. Experimentování je zamýšleno tak, že při testování analýzy názorů budou použity různé způsoby klasifikace za odlišných podmínek a jejich výsledky spolu budou porovnány.

Kapitola 2

Analýza názorů

Nejspíše každý obchodník či firma poskytující služby nebo prodávající nějaké produkty, chce vědět, co si lidé o jejich produktech či službách myslí a zda-li by se měli v některých ohledech zlepšit či zda-li se produkty a služby lidem líbí tak jak jsou právě poskytovány. Při pořízení produktu obvykle jen málo lidí napíše prodávajícímu recenzi. Navíc existuje mnoho různých recenzních systémů. Velmi často se však potkáme s případy, kdy lidé sdělují své názory na sociálních sítích (např. Twitter, Facebook apod.) Nejen v tomto odvětví, ale také v dalších lze využít právě analýzu názorů k získání obecnějšího povědomí o tom, co si lidé o produktu myslí. Analyzátor názorů dokáže získat informace ze sociálních sítí i z různých recenzí či jiných zdrojů, které obsahují textové vyjádření něčího názoru.

Jedná se o metodu zpracování přirozeného jazyka, kterou lze označit i jako textovou analýzu nebo dolování názorů (angl. *opinion mining*). Pokud chceme zjistit postoj či názor subjektu na určité téma, analýza názorů je jeden ze způsobů jak toho docílit [1][17]. Můžeme ji provádět na různcých úrovních textu [11]:

- *slovo* – určení postoje samotného slova
- *fráze* – často používáné slovní spojení 2 či více slov
- *věta* – ohodnocení celé věty či úseku textu (nadpis, položka listové struktury apod.)
- *dokument/množina* – kolekce či množina textů, celý dokument (recenze, písemný dokument apod.)

Analýzu názorů můžeme rozdělit na dva základní typy [14]. Jeden pojednává o určování, zda-li je vybraný text ovlivněn emocemi autora nebo zda-li autor textu pouze konstatuje fakta. Druhý (pro nás důležitější) typ se věnuje rozlišení zjištěného postoje na *pozitivní* nebo *negativní*. Obvykle přidáváme ještě *neutrální* kategorii. Může však existovat i typ, kdy nerozdělujeme postoj jen do kategorií, ale přímo na hodnoty stupnice, která určí, do jaké míry je analyzovaný text pozitivní nebo negativní. Ještě složitější typ analýzy názorů by byl kdybychom nechtěli zjišťovat pouze postoj samotný, ale také k čemu je postoj navázán (např. host hotelu hodnotí kladně pokojovou službu, ale vůbec se mu nelíbí stravování).

Stručně lze říci, že analyzátor názorů se skládá z operací *zpracování textu* – 2.2 a *klasifikace postoje* – 2.3. Proces analýzy názorů se však skládá z několika hlavních částí [1]:

1. *extrakce příznaků* – získání samotných slov a výrazů z textu a jejich zpracování do tzv. vektoru příznaků

2. *extrakce nosičů názorů* – získání nosiče, ke kterému se bude vztahovat zjištěný postoj (v recenzi hotelu např. ubytování, stravování apod.), provádí se pouze pokud nám nestačí obecná analýza vloženého textu, ale chceme zjistit i co si člověk myslí detailněji

3. *klasifikace postoje* – přiřazení vektoru příznaků výsledný postoj obvykle za použití metod strojového učení

2.1 Sentiment

O sentimentu mluvíme jako o náladě mysli, vzniklé z přehnaného poddávání se cítům kdy je objektivní názor na věc zcela zanedbán [10]. Sentiment můžeme také chápat jako myšlenku, názor či postoj člověka k určitému tématu [7], což je pro nás při analýze názorů důležitá informace. Ve slovním spojení *analyzátor sentimentu* je význam slova *sentiment* chápán právě jako postoj nebo názor člověka. Takový názor může být označován jako pozitivní či negativní a v některých případech i jako neutrální.

2.2 Zpracování textu

Můžeme očekávat, že data, která budeme chtít analyzovat, budou ve formátu prostého textu (angl. *plain-text*) nebo i zřídka ve formátu nějakého značkovacího jazyka (angl. *markup language*) [6]. Abychom mohli provést klasifikaci postoje, budeme si text chtít přetvořit do *vektoru příznaků* (angl. *feature vector*), což je reprezentace objektu, který obsahuje jednotlivé extrahované příznaky (angl. *features*). Vektor příznaků potřebujeme zejména proto, že je lépe zpracovatelný klasifikačními algoritmy a zároveň dobře uchovává informace o vstupním textu [11][14].

Kvalitní vektor příznaků můžeme získat provedením následujících operací. Ze všeho nejdříve je potřeba text rozdělit na jednotlivá slova. Pokud chceme převádět text na pouze velká nebo pouze malá písmena, je efektivní provést to právě v předchozím kroku, nikoliv později. Rozdělením textu na jednotlivá slova se zabývá hlavně *tokenizace* – 2.2.1 a částečně následná *filtrace nepotřebných slov* – 2.2.2. Poté pro zvýšení přesnosti analýzy názorů můžeme ještě provést operace *lemmatizace* – 2.2.3, *POS tagging* – 2.2.4 a *negace* – 2.2.5, které jsou vysvětleny níže [6][11][12].

Po provdení úprav textu uvedených výše získáme vektor příznaků, který když použijeme při klasifikačních algoritmech, tak bychom měli dosáhnout dostatečně přesných výsledků. Přesnost analýzy se ovšem odvíjí i od jiných faktorů jako např. gramatická kvalita vstupního textu, použití vícesmyslových slov apod. Existují i další metody, které mohou zlepšít přesnost analýzy názorů, ale těmi se zde zabývat nebudeme [6].

2.2.1 Tokenizace

Jedná se o bežně používanou operaci v oblasti zpracování přirozeného jazyka, která rozdělí text na tzv. *tokeny*, což mohou být např. slova, slovní spojení, interpunkce apod. Neexistuje pouze jedna správná cesta, jak rozdělovat text na tokeny. Záleží na tom, jaký problém řešíme a tím pádem jaký tokenizátor potřebujeme [6][12].

Najít správný způsob tokenizace nebývá vždy jednoduché. Slova jsou totiž oddělena nejen bílými znaky, ale i různými interpunkčními znaménky. Měli bychom si uvědomit, zda-li jsou pro nás důležité následující entity a pokud ano, jakým způsobem je zpracovávat [6]:

- tzv. emotikony – určité kombinace interpunkčních znaků a písmen jako např. :-D >:-(
- slovní spojení *co-operation*, *e-mail* apod.
- speciální značkování jako např. <bold>, &lt apod.
- e-mailové adresy
- telefonní čísla
- URL
- další znakově kombinované entity jako např. A4, $5.99, 0x0001 apod.

Pro nás jsou z výše uvedených při analýze názorů důležité zejména emotikony v textu, jelikož lidé na sociálních sítích vyjádřují své emoce, cítění a názory také pomocí emotikon.

2.2.2 Filtrace nepotřebných slov

Známá také pod označením *stopwords*. Jedná se o slova či tokeny, která mají být při zpracování přirozeného jazyka vyfiltrována, čili jsou to slova, která dále v analýze nechceme či nepotřebujeme [13]. Seznamů stopwords existuje mnoho druhů a obsahují slova v závislosti na tom, který systém nebo aplikace je používá a hlavně v jakém jazyce. Technika slouží zejména k zefektivnění analýzy názorů. Stopwords obvykle nejsou nositeli informace o postoji což je další důvod pro jejich vyfiltrování z textu.

Mezi stopwords patří hlavně často používaná slova což nejčastěji bývají předložky, spojky a zájmena (např. *a, k, on, tak, byl, který,* apod.) V anglickém jazyce jsou to také členy. Dále se do stopwords někdy (např. v našem případě) zahrnují samotná interpunkční znaménka (samotná protože nechceme odfiltrovat interpunkci, která je součástí libovolného tokenu, např. emotikona) [14].

2.2.3 Lematizace

Jedná se o převod slova do jeho základního tvaru, který nazýváme *lemma* [6][5][14]. Operace přispěje ke kvalitě analýzy názorů tak, že sjednotí různé tvary určitého slova do jeho základního tvaru čímž podstatně sníží počet slov, vyskytujících se ve výsledném vektoru přiznaků a také zvýrazní použitá slova v jejich základních tvarech.

K lematizaci je potřeba znát základní tvary slov a také je třeba u každého slova rozeznat, ke kterému základnímu tvaru jej přiřadit. Je tedy potřeba využít dalších zdrojů, které analýzatoru poskytnou informace o tvarech slov. Tento problém lze částečně obejít použitím metody *stematizace*, která se od *lematizace* liší tím, že nepřevádí slova do jejich základních tvarů, ale pouze usekne předpony a přípony, takže ze slova zbyde jen jeho kmen [5].

2.2.4 POS tagging

Další metoda, která může navýšit kvalitu analýzy názorů je tzv. *part-of-speech tagging* nebo-li *POS tagging*. Zjišťuje slovní druhy a tvary zpracovávaných slov a následně tato slova náležitě označí (obvykle slovním druhem a případně i informací o tvaru, ve kterém se slovo nachazí) [6][14]. Metoda má za následek, že jednotlivé příznaky výsledného vektoru příznaků již nebudou obsahovat pouze slova, nýbrž dvojce *slovo-tag* nebo-li *slovo-značka*.

Tento krok může vést ke zvýšení kvaliti analýzy názorů tak, že analyzátor bude moci slova lépe klasifikovat na základě slovních druhů. Další způsob zlepšení kvality je použití

POS taggingu v kombinaci s lematizací nebo stematizací jelikož získáme sice ořezaná slova (tedy pouze slova v základních tvarech nebo jejich kmeny), ale díky POS taggingu budeme mít k dispozici informace o tvaru, ve kterém se slovo nacházelo. Dalo by se tedy říci, že POS tagging vytváří náhradu za informace, které ztratíme při lematizaci nebo stematizaci.

2.2.5 Negace

Výrazné zlepšení kvality výsledku může zajistit skutečnost, že budeme brát v potaz negaci a správně s ní naložíme. Např. věty *„I like this game“* a *„I don't like this game“* jsou z hlediska použitých slov velmi podobné, mají ovšem zcela opačný význam [11][12]. Věty, v nichž je použita negace, se mohou na sociálních sítích i jinde vyskytovat často a cheme-li tento problém řešit, musíme vybavit analyzátor sentimentu tak, aby negaci byl schopen rozpoznat a správně zpracovat.

Jeden z přístupů jak vyřešit problém negace může být upravení vektoru příznaků tak, že nalezneme-li určitý přiznak značící negaci (např. *don't*, *never* apod.), pak označíme nebo upravíme jeden či více následujících přiznaků, aby bylo jasné, že jsou negované. Obvykle se snažíme označit příznaky od slova značící negaci až po zbytek věty samotné nebo pokud se jedná o souvětí tak po zbytek oné věty v souvětí. Označit příznak můžeme buď přidáním značky a tím vytvořit dvojci *příznak-značka* nebo lze použít efektivnější metodu, kdy upravíme přímo slovo v příznaku tak, že např. na začátek přidáme řetězec *„NOT_“* (např. z věty *„I don't like this game“* získáme příznaky *I, NOT_like, NOT_this, NOT_game*) [11].

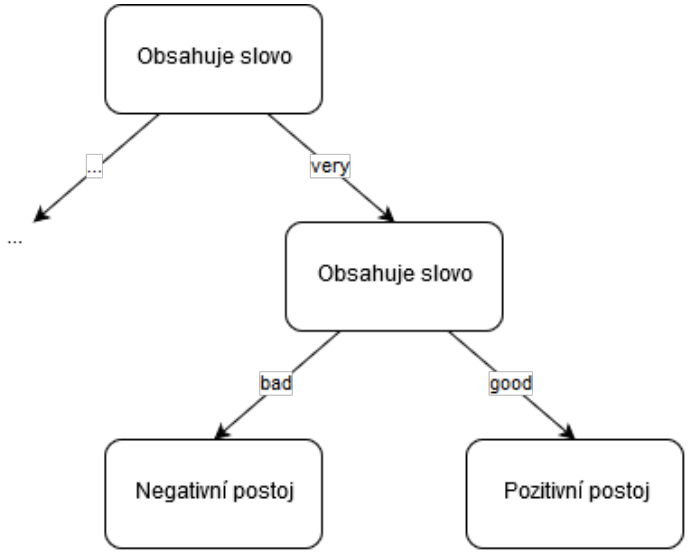
2.3 Klasifikace

Analyzátor sentimentu musí být schopen rozdělit analyzovaná data do určitých kategorií (v našem případě *pozitivní*, *negativní* a *neutrální*). Tento proces se nazývá *klasifikace* a k jeho provedení se používají zejména algoritmy strojového učení. Existuje mnoho způsobů jak lze klasifikaci provést a my si popíšeme několik používaných, které jsou vhodné pro provádění analýzy názorů. Algoritmy využívané při klasifikaci se dělí do dvou kategorií [8][11]:

- *učení s učitelem* (angl. *supervised learning*) – Algoritmům jsou předkládána kromě textů ke zpracování také tzv. *anotovaná data*. Jde o způsob, kterým dáváme algoritmu na vědomí, jaká data jsou očekávána na jeho výstupu [8]. Mezi algoritmy učení s učitelem patří např. *Rozhodovací stromy*, *Naivní Bayesův klasifikátor*, *Maximální entropie*, *Nearest Neighbour*, *Support Vector Machines*, aj.

- *učení bez učitele* (angl. *unsupervised learning*) – Algoritmům jsou předkládána pouze data ke zpracování. Algoritmus tedy neví, jaký výstup je očekáván a musí jej určit sám např. na základě podobnosti dat. Metody učení bez učitele jsou lépe aplikovatelné, a to zejména protože narozdíl od metod učení s učitelem nepotřebují člověka, aby manuálně připravoval anotovaná data [8]. Patří zde např. *k-means*, *Neurální sítě*, aj.

Nejčastěji používané metody pro klasifikaci textu čili i pro analýzu názorů jsou metody učení s učitelem [14]. Algoritmům bude tedy vždy kromě dat ke zpracování předkládána také tzv. *trénovací sada*, která poslouží jako anotovaná data.



Obrázek 2.1: Příklad jednoduchého rozhodovacího stromu při klasifikaci textu *„The Game was very good“*

2.3.1 Rozhodovací stromy

Anglicky známé jako *Decision Trees* jsou grafům podobné nástroje, které slouží pro podporu rozhodování. Při zpracování přirozeného jazyka se rozhodovací stromy používají k vybrání nejvhodnějších lingvistických znalostí pro řešení zadaného problému [4][6].

Každý uzel reprezentuje test určité události, vlastnosti nebo stavu (např. jaké číslo padne na hrací kostce) a každá větev reprezentuje výsledek onoho testu (např. na hrací kostce padlo číslo 4). Listy stromu pak symbolizují rozdělení do kategorií nebo-li rozhodnutí po propočtu všech potřebných vlastností.

Velikou výhodou rozhodovacích stromů je jejich jednoduchá interptretace a nenáročnost na pochopení. Příklad jednoduchého rozhodovacího stromu lze vidět na obrázku 2.1.

2.3.2 Naivní Bayesův klasifikátor

Anglicky označován jako *Naive Bayes*. Jedná se o metodu pravděpodobnostního učení, která je založena na *Bayesově větě* (angl. *Bayes' theorem*), která lze matematicky vyjádřit jako

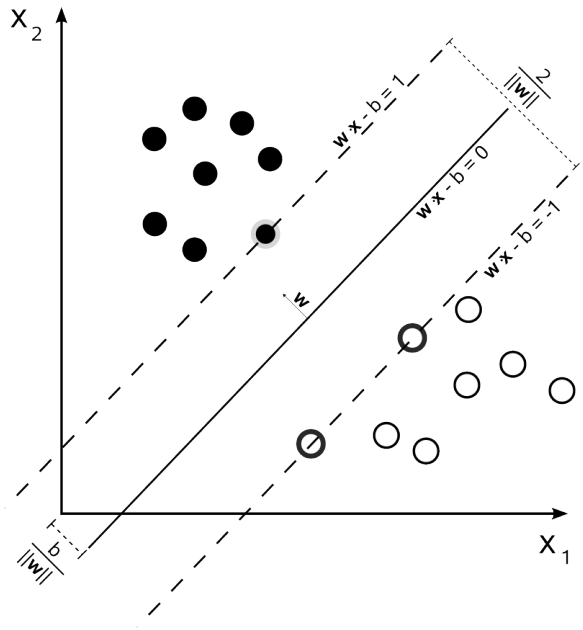
$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{2.1}$$

kde A a B jsou události, *P(A)* je pravděpodobnost, že nastane událost A (obdobně pro *P(B)*) a *P(A|B)* je podmíněná pravděpodobnost, že nastane událost A pokud nastala událost B (obdobně pro *P(B|A)*).

Pravděpodobnost, že dokument d náleží do kategorie (třídy) c lze vypočítat jako

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \tag{2.2}$$

kde $P(t_k|c)$ je podmíněná pravděpodobnost výskytu termu t_k v dokumentu kategorie c. Jinak řečeno lze $P(t_k|c)$ interpretovat jako míru toho, jak moc $P(t_k|c)$ nasvědčuje tomu, že třída c je vybrána správně. $P(c)$ je apriorní pravděpodobnost, že dokument spadá do kategorie c. $\langle t_1, t_2, \ldots, t_{n_d} \rangle$ jsou příznaky z dokumentu d a n_d je počet těchto příznaků [5][12].



Obrázek 2.2: Klasifikace metodou SVM s označením nadroviny, podpůrných vektorů a hraničního pásma [16]

2.3.3 Maximální entropie

V metodě anglicky známe jako *Maximum Entropy* se opět zjišťuje podmíněná pravděpodobnost a na základě této pravděpodobnosti se dokumentu přiřazuje kategorie. Narozdíl od Naivního Bayesova klasifikátoru se nepředpokládá, že jsou na sobě příznaky zavislé [6][12][14]. Základní odhad pravděpodobnosti lze vypočítat jako

$$P(x_i) = \frac{1}{Z(\lambda_1, \ldots, \lambda_m)} exp[\lambda_1, f_1(x_i) + \cdots + \lambda_m, f_m(x_i)] \tag{2.3}$$

kde f je funkce, která nabývá hodnot 0 nebo 1 podle toho, zda-li se v daném textu vyskytuje i-tý příznak. λ je pak váha tohoto příznaku pro danou kategorii a hodnota tohoto parametru se nastavuje tak, aby bylo dosaženo co nejvyšší entropie. Normalizační konstantu $Z(\lambda_1, \ldots, \lambda_m)$ lze vypočítat jako

$$Z(\lambda_1, \ldots, \lambda_m) = \sum_{i=1}^{n} exp[\lambda_1, f_1(x_i) + \cdots + \lambda_m, f_m(x_i)] \tag{2.4}$$

2.3.4 Support Vector Machines

Známá také pod zkratkou *SVM* je metoda učení s učitelem, která klasifikuje zadané příklady do jedné ze dvou kategorií. Nejprve je nutné vložit anotovaná data (trénovací sadu), která jsou poté promítnuta do prostoru. V tomto prostoru se pak hledá hranice nazývaná *nadrovina* (angl. *hyperplane*), která rozdělí prostor na dvě části a tím pádem i vložená data do dvou kategorií [5][14][16]. Né vždy lze data přesně rozdělit do dvou kategorií a v takovém případě se metoda snaží o rozdělení co nejoptimálnější. Data nacházející se nejblíže nadrovině se nazývají *podpůrné vektory* (angl. *support vectors*), z čehož pochází název metody.

Nadrovinu můžeme vyjádřit rovnicí

$$\vec{w} \cdot \vec{x} - b = 0 \tag{2.5}$$

kde $\vec{w}$ je normálový vektor nadroviny, $\vec{x}$ je vstupní vektor trénovacích dat a parametr b udává polohu vzhledem k počátku.

Na obrázku 2.2 můžeme vidět v blízkosti nadroviny oblast, kde se nevyskytují žádná data. Takovou oblast nazýváme *hraničící pásmo* (angl. *margin*) a ohraničují ji právě podpůrné vektory.

Kapitola 3

Návrh systému

Kapitola se věnuje návrhu celého systému pro stahování a analýzu dat. Nejprve si rozebereme možnosti, jak navrhnout systém tak, aby co nejlépe odpovídal stanoveným cílům této práce. Vybereme si nejvhodnější způsob návrhu a implementace a v sekci 3.1 si uvedeme a zvolíme vhodné prostředí a nástroje k vývoji systému.

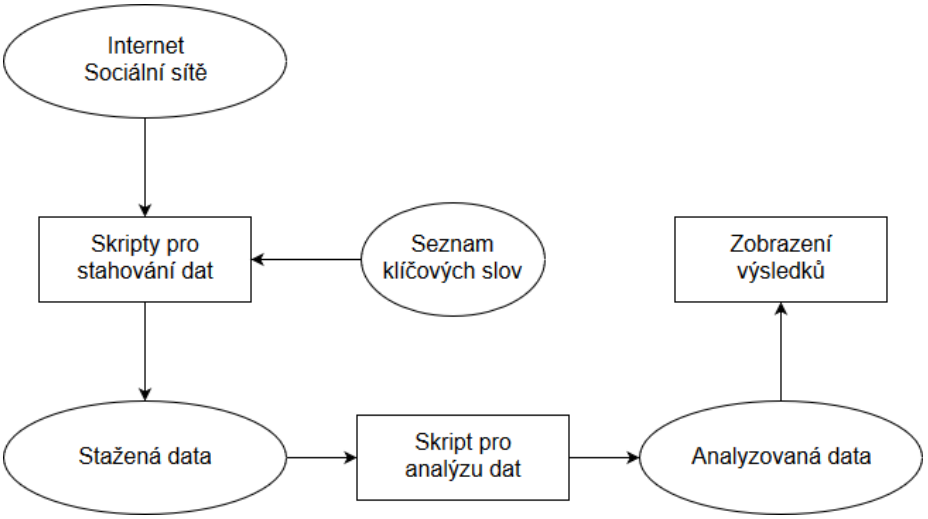
Jak je popsáno v úvodní kapitole 1, mým cílem a tedy i cílem práce je vytvořit systém, který bude stahovat, analyzovat a vhodně interpretovat data, zde konkrétně postoje k počítačovým hrám. Jaká data se budou zpracovávat bude určeno připraveným *seznamem klíčových slov*. V našem případě bude seznam obsahovat právě jednotlivé názvy herních titulů a sérií. Systém bychom mohli navrhnout a implementovat jako jediný celek (jediný modul, jediný skript), který by provedl hned všechny úkoly. Takové řešení by však bylo velmi neefektivní z hlediska časové náročnosti a ovladatelnosti. Je vhodné si systém rozdělit na jednotlivé části, reprezentující jednotlivé úlohy. Stručné schéma systému lze vidět na obrázku 3.1.

Rozdělíme tedy systém na 3 hlavní části:

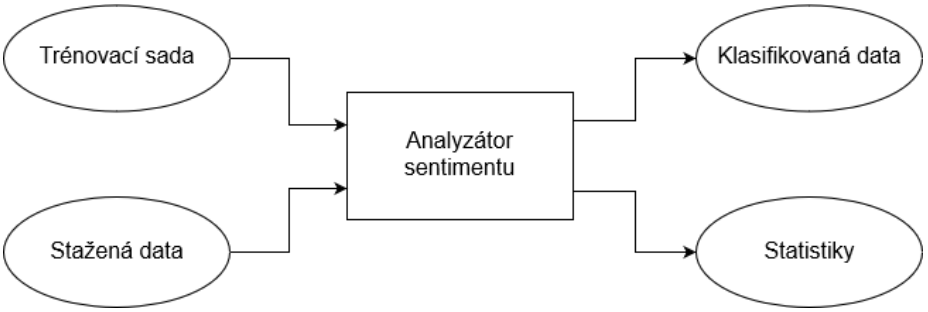
1. **stahování dat** – Nejprve bude potřeba obstarat data, která bychom mohli analyzovat. Jaká data to budou čili co máme obstarat nám určí *seznam klíčových slov*. Data budou stahována ze sociálních sítí, a to zejména Twitter[1] a Facebook[2]. Je třeba si také zvolit způsob ukládání a indexování dat, který musí dodržovat určitou strukturu, jelikož u staženého obsahu nás bude zajímat zejména text příspěvku, datum a autor příspěvku. Můžeme si např. vybrat zda-li příspěvky budeme ukládat do každéhou souboru zvlášť nebo zda-li budeme mít pro každý titul jeden soubor a do něj budeme ukládat všechny příspěvky vztahující se k danému titulu. Je třeba také určit formát ukládání dat tak, abychom mohli ukládat různé druhy informací. Např. lze použít JSON[3] (zkratka z angl. *JavaScript Object Notation*) či CSV[4] (zkratka z angl. *Column Separated Values*).

2. **analýza dat** – Stažená data je třeba klasifikovat a uložit jako analyzovaná data, aby s nimi mohla pracovat část systému pro zobrazení výsledků. Hlavní úkol analýzy spočívá v klasifikaci postoje ke staženým datům. Pak je zde ještě vedlejší úkol, a to vytváření statistik už během analýzy, aby se část systému pro zobrazování výsledků touto procedurou nezdržovala, ale využívala už vytvořené soubory se statistikami.

[1] https://twitter.com/
[2] https://www.facebook.com/
[3] http://www.json.org/
[4] https://tools.ietf.org/html/rfc4180



Obrázek 3.1: Stručné schéma navrhovaného systému



Obrázek 3.2: Stručné schéma části systému pro analýzu dat

Analýza názorů bude probíhat pomocí metod strojového učení, které byly zmíněny v sekci 2.3, takže bude zapotřebí anotovaných dat, čili předvytvoření testovací sady. Lze si vytvořit vlastní testovací sadu specializovanou přímo na danou problematiku, která bude obsahovat příklady příspěvků s již ohodnoceným postojem, a nebo lze použít některý z již existujících korpusů. Takové sady však bývají příliš obecné a jejich použití by mohlo vést k neuspokojivým výsledkům analýzy názorů. Stručné schéma této části systému zle vidět na obrázku 3.2.

3. **zobrazení výsledků** – Výsledky musejí být uživateli zobrazeny správně, rychle a přehledně. Je tedy vhodné zejména v této části systému využít grafických nástrojů. K rychlosti zobrazení již přispívá předvytvoření souborů se statistikami z předchozí části. Uživatel by měl mít možnost vidět kromě samotných příspěvků k vybranému titulu také obecnější statistiky jako např. žebříček titulů s nejvíce příspěvky a nebo mapu intenzity počtu příspěvků podle zeměpisné lokace.

První dvě části (stahování a analýza dat) mohou zůstat téměř nezměněny v případě, že se rozhodneme použít systém pro jiná témata než jsou počítačové hry. Bude pouze třeba náležitě upravit *seznam klíčových slov* a případně i anotovaná data pro metody klasifikace. Poslední část však zobrazuje o právě vybrané počítačové hře informace získané z jiného

zdroje (herní žánr, vývojáři hry, krátký popis hry apod.), což by při použití jiných témat mohlo působit rušivě.

3.1 Vývojové prostředí a použité nástroje

Bude-li systém rozdělen do 3 částí takovým způsobem, že jedna část systému bude pouze vytvářet vstupní data pro další část systému, pak lze uvažovat použití rozdílných vývojových nástrojů a prostředí pro každou část systému. U stahování a analýzy dat není důležitá grafická stránka, a mohly by se tedy realizovat jako konzolové aplikace nebo skripty. Naopak pro zobrazení výsledků bychom museli použít knihovny či přídavné moduly pro realizaci GUI[5], a nebo můžeme využít nástrojů, které jsou přímo specializované a propojené s GUI.

Stahování dat klade důraz zejména na správnou a efektivní extrakci požadovaných dat z internetových zdrojů (zde konkrétně se jedná o sociální sítě). Je také potřeba schopnost správně pracovat s formátem zvoleným pro ukládání dat, což v našem případě bude JSON, a to hlavně díky jeho rozšířenosti. Všechny tyto předpoklady splňuje skriptovací jazyk *Python*, který byl vybrán pro implementaci této části také proto, že se jedná o jazyk nenáročný, velmi rozšířený, dobře dokumentovaný a s bohatým repertoárem přídavných knihoven [2]. Mezi knihovny, které lze v této práci využít pro zpracování přirozeného jazyka a strojového učení, patří např. *NLTK*[6], *TextBlob*[7] nebo *SpaCy*[8].

Jelikož je očekáván rozsáhlý seznam titulů (v řádu stovek), pak aby souborů nebylo vytvářeno příliš málo a ani příliš mnoho, byl zvolen následující způsob ukládání: Každý titul bude reprezentován jedním souborem, do kterého se budou ukládat všechny příspěvky, které se k danému titulu vztahují. V případě, že existuje více zdrojů, ze kterých bylo k danému titulu čerpáno, pak titul bude reprezentován složkou a v ní soubory pojmenovanými podle nalezených zdrojů (stránek). Jednotlivé příspěvky pak budou v těchto příslušných souborech.

Analýza dat musí být schopna určit postoj textu s co největší přesností. Musí si také poradit s vybraným formátem pro vstupní i výstupní data. Stejně jako u předchozí části je i zde Python dobrá volba, a to hlavně proto, abychom nepoužívali zbytečně mnoho ruzných nástrojů. Stejné knihovny pro zpracování přirozeného jazyka lze obvykle využít i ke klasifikaci postoje. Je tedy potřebné, aby vybrané knihovny byly schopny provádět analýzu pomocí metod strojového učení. Testovací sada je očekávána ve formátu CSV kde první sloupec obsahuje text a druhý sloupec obsahuje postoj přidělený textu v prvním sloupci.

Analýza dat bude vytvářet i speciální soubory, které budou obsahovat statistiky a to zejména takové, které lze získat pouze postupným procházením obrovského počtu příspěvků. Jedná se např. o počty příspěvků z jednotlivých zeměpisných lokací. Jelikož pracujeme se strukturovanými daty, budou soubory také ve formátu JSON.

Zobrazení výsledků klade velký důraz na správnost, rychlost a přehlednost. Uživateli se kromě obecných statistik musejí ukázat také jednotlivé příspěvky tak, jak byly staženy

[5] Uživatelské rozhraní (angl. *Graphical User Interface*)
[6] http://www.nltk.org/
[7] https://textblob.readthedocs.io/en/dev/
[8] https://spacy.io/

z internetových zdrojů. Ideální volbou je webový portál, což přináší výhodu, zobrazit výsledky z jakéhokoliv zařízení, které má internetový prohlížeč. Serverové počítače bývají z hlediska výpočetního výkonu rychlejší než jiná bežně používaná zařízení.

Pro zobrazení výsledků budou tedy použity webové technologie HTML, CSS, PHP a JavaScript. Server využije poskytnuté soubory a zpracované informace pošle do prohlížeče uživatele, kde bude použita JavaScriptová knihovna *amCharts*[9], která se specializuje na tvorbu grafů a map.

Obecně budeme chtít zobrazit, které tituly jsou nejdiskutovanější (bude se jednat o graf se sloupci seřazenými podle počtu příspěvků k danému titulu) a pak také mapu, která bude ukazovat, kolik příspěvků a z jakých různých zemí bylo analyzováno. Při výběru konkrétního titulu bude portál schopen zobrazit časovou osu příspěvků (počet příspěvků za určitý časový interval) a samotné příspěvky rozdělené podle kategorií postojů.

[9] https://www.amcharts.com/

Kapitola 4

Implementace

Kapitola pojednává o implementaci výsledného systému pro analýzu názorů. Upřesníme si zde, jaké vývojové nástroje byly použity, které knihovny či přídavné moduly byly vybrány a jak s nimi bylo nakládáno. Kapitola je rozdělena na 3 sekce – každá pokrývající jednu část systému podle rozdělení z předchozí kapitoly 3. V sekci 4.1 rozebereme jakými způsoby jsou získávána data ze sociálních sítí a jejich zpracování vzhledem k úkládání. Další sekce 4.2 pojednává o způsobu přípravy a zpracování textu, o použitých metodách strojového učení a o tvorbě souborů se statistikami. V poslední sekci 4.3 zmíníme popstup implementace webového portálu pro zobrazení výsledků, způsob a podmínky zobrazování jednotlivých prvků webu a které zdroje jsou portálu poskytnuty a jak jsou využívány.

Části systému *stahování dat* a *analýza názorů* jsou realizovány jako samostatné skripty v jazyce Python 3.5.2 a jsou spouštěny jako konzolové aplikace. Při běhu tisknou na standartní výstup informace o svém průběhu (např. který titul je právě zpracováván a kolik nových příspěvků k němu bylo staženo). Využívají společný soubor `usedVars.py`, ve kterém jsou uchovány globální proměnné společné pro oba skripty (např. název a cesta adresáře pro ukládání dat). Jazyk Python nabízí možnosti objektového programování, která byla při konstrukci těchto skriptů využita a kombinována s funkcionálním programováním.

4.1 Skripty pro stahování dat

Jedná se o 2 skripty, `downloadF.py` pro získávání dat ze sociální sítě Facebook a `downloadT.py` pro stahování dat z Twitteru. Oba pracují velmi podobně a liší se pouze ve způsobu získávání dat ze serverů a jejich ukládání do souborů a adresářů.

Kromě souboru s globálními proměnnými `usedVars.py` se zde využívá ještě soubor `keys.py`, který obsahuje bezpečnostní a identifikační klíče pro přístup k *REST API*[1] a *Graph API*[2], což jsou rozhraní, skrze která mohou aplikace přistupovat k datům na uvedených sociálních sítích. Dále je ještě využíván *seznam klíčových slov*, což je textový soubor s jedním klíčovým slovem (herním titulem) na každém řádku. Implicitní název souboru je `seznam.txt`. Je nutné uvědomit si, že skripty pro stahování dat pracují se seznamem klíčových slov tak, že vezmou jeden celý řádek souboru (tedy celé jedno klíčové slovo) a použijí jej jako dotaz pro vyhledávání pomocí odpovídající API. Je tedy vhodné použít jako klíčová slova přímo názvy jednotlivých herních titulů.

[1] https://dev.twitter.com/rest/public/
[2] https://developers.facebook.com/docs/graph-api

Skript pro stahování dat z Twitteru využívá externí knihovnu *TwitterSearch*[3], která zjednodušuje práci s *REST API*, kdežto skript pro stahování dat z Facebooku využívá knihovnu *requests*[4], jelikož v době tvorby této části práce nebyly k dispozici knihovny, které by vhodně ulehčovaly práci s *Graph API*.

Při zpracovávání dat z Facebooku nelze příspěvky vyhledávat přímo podle klíčového slova, a proto jsou vyhledávány podle klíčových slov pouze stránky a jsou stahovány jejich příspěvky. Narozdíl od zpracovávání dat z Twitteru zde nevytváříme jeden soubor na herní titul, ale vytváříme jednu složku na herní titul. Ve složce jsou pak soubory se jmény a ID odpovídajících Facebookových stránek. Tyto soubory obsahují stahované příspěvky z uvedených stránek.

Po spuštění si skripty nejprve zkontrolují cílové adresáře (pokud neexistují pak je skripty vytvoří) a načtou do paměti jednotlivé tituly ze seznamu klíčových slov. Jakmile úspěšně ověříme identifikační a bezpečnostní klíče v odpovídající API, začneme cyklovat nad polem klíčových slov. V každém cyklu nejprve zkontrolujeme, zda-li už neexistuje soubor odpovídající danému titulu. Pokud existuje, načteme si do paměti datum posledního příspěvku, které využojeme později. Pokud soubor neexistuje, je pouze vytvořen nový prázdný soubor. Poté posíláme přes API žádosti o příspěvky související s daným titulem tak dlouho, dokud nenarazíme na stejný příspěvek, jehož datum máme uložené v paměti, nebo tak dlouho dokud nám API dovolí stahovat data (Twitter při přístupu přes API dovoluje stahovat příspěvky staré maximálně jeden týden). Může se stát, že během posílání požadavků narazíme na limit (*REST API* dovoluje maximálně 180 požadavků během 15 minut a *Graph API* dovoluje maximálně 200 požadavků za hodinu). V takovém případě musí skripty vyčkat daný časový interval než mohou opět zasílat požadavky na servery. Po získání požadovaných příspěvků se tyto uloží do souboru spojeného s právě zpracovávaným herním titulem a cyklus přechází na další klíčové slovo.

Pokud nastane chyba při práci se soubory nebo při práci s API, je odchycena a na standartní výstup je vytisknuto odpovídající chybové hlášení. Může se také stát, že při odeslání požadavku na server z neznámého důvodu nedostaneme odpověď. Pro takový případ je využit algoritmust *Exponential Backoff*, jehož princip spočívá v tom, že při chybě vyčkáme určitou dobu, než se pokusíme operaci znovu provést a pokud opět nastane chyba, čekací doba bude dvojnásobně větši než při předchozím čekání. Je určena mez, kolikrát se chyba může opakovat, než to algoritmmus vzdá, vypíše chybové hlášení a ukončí se [15].

4.2 Skript pro analýzu dat

Jedná se o jeden soubor `senti.py` spouštěný z příkazové řádky s jedním parametrem, který určuje, zda-li se budou analyzovat data stažená ze sociální sítě Facebook (pouze pokud je skript spouštěn s parametrem `F`) nebo zda-li analýza proběhne na datech z Twitteru (pokud je zadán jakýkoliv jiný parametr nebo pokud není zadán vůbec).

Skript ke svému fungování potřebuje soubor s globálními proměnnými `usedVars.py` a dále k práci požaduje soubory `stopwords.txt`, který využívá filtr nepotřebných slov 2.2.2, a `training_set.csv` jako trénovací sadu pro klasifikaci 2.3. Samozřejmně nesmějí chybět data, která chceme analyzovat.

Jsou využity externí knihovny *spaCy* a *NLTK* (sekce 3.1). Knihovna *spaCy* je využita kvůli jednoduchému a efektivnímu zpracování textu, a to zejména tokenizaci. Z *NLTK* jsou

[3] https://twittersearch.readthedocs.io/en/v0.78.2/

[4] http://docs.python-requests.org/en/master/

používány metody strojového učení[5] *Rozhodovací stromy* 2.3.1, *Naivní Bayesův klasifikátor* 2.3.2 a *Maximální entropie* 2.3.3 pro následnou klasifikaci sentimentu.

Základním kamenem skriptu je třída `sentiment_analyzer`, kterou je třeba inicializovat a natrénovat na testovací sadě a poté lze provádět klasifikaci prostojů vstupních dat. Jsou ještě přítomny třídy `global_statistics` a `frequent_statistics`, obě se věnující zaznačování a ukládání statistik během analýzy.

Spustíme-li skript, nejprve se musejí načíst moduly a externí knihovny. Poté zkontrolujeme, zda-li byl zadán parametr a náležitě nastavíme potřebné proměnné, podle nichž pak bude skript probíhat. Zkontrolují se a případně i vytvoří adresáře, kam budeme ukládat analyzovaná data. Inicializujeme instanci třídy `sentiment_analyzer` čímž také proběhne natrénování příslušné metody strojového učení (implicitně *Naivní Bayesův klasifikátor*) na trénovací sadě. Vytvoříme si seznam souborů z adresáře se staženými daty a vstoupíme do cyklu, který bude postupně zpracovávat všechny soubory z vytvořeného seznamu (další možností by bylo procházet soubory pomocí seznamu klíčových slov, což by ale mohlo být neefektivní až nebezpečné, protože bychom mohli narazit na neexistující soubory – tituly, ke kterým nemáme stažena žádná data). Načteme soubor do paměti a jednotlivé příspěvky v něm začneme klasifikovat pomocí třídy `sentiment_analyzer`, která k tomuto účelu využívá metody knihovny *NLTK*. Po zpracování data uložíme do odpovídajícího souboru, doplníme třídy se statistikami a zároveň statistiky vypisujeme na standartní výstup. Přístup k datům se mírně liší podle toho, zda-li jsou zpracovávána data z Facebooku nebo Twitteru. Statistiky jsou jednou za určitý časový interval uloženy do souborů. Uvedený cyklus se při zpracovávání dat z Facebooku upakuje pro každý titul zvlášť z důvodu způsobu rozdělení dat do složek (viz předchozí sekce 4.1).

4.3 Webový portálu pro zobrazení výsledků

Základ portálu je postaven na technologii PHP a nástroji s ní spojenými (HTML, CSS, JavaScript). K funkčnosti značně přispívá také technologie AJAX[6], která nám umožňuje měnit pouze konkrétní obsah stránky bez potřeby načítat do prohlížeče celou stránku znova. Ke zobrazení výsledků jsou taktéž využity grafy sestrojené pomocí externí JavaScriptové knihovny *amCharts*, která ke svému správnému fungování potřebuje ještě knihovnu *jQuery*. Jelikož drtivá většina klíčových slov, použitých v této práci, byla čerpána z webového portálu *GiantBomb*[7], je při zobrazení výsledků o konkrétních titulech využita *GiantBomb API*.

Portál je umístěn na fakultním serveru *athena1* na url adrese `http://athena1.fit.vutbr.cz/xbalaj03/www/` a skládá se pouze z jediné webové stránky `index.php`, která se pomocí technologie AJAX dotazuje na soubor `functions.php` a z něj získává zpracované informace, které uživatel zrovna požaduje. Kromě standartních souborů pro webové stránky jako např. `style.css` či `scripts.js` a kromě souborů externích knihoven jsou také využívána analyzovaná data a soubory se statistikami `X_global_stats.json` a `X_frequent_stats.json` kde `X` je identifikační znak určující, na základě jakých dat a z které sociální sítě jsou statistiky vytvořeny (`T` pro Twitter a `F` pro Facebook).

Po vstupu na webovou stránku jsou uživateli nejprve zobrazeny obecné statistiky a grafy (v této časti jsou využity pouze soubory se statistikami, nikoliv samotná analyzovaná data,

[5] Knihovna *spaCy* sice taktéž obsahuje metody strojového učení, ale bohužel ze zdrojů dostupných na internetu jsem nedokázal zjistit, jak správně tyto metody použít, kdežto u knihovny *NLTK* jsem byl schopen metody úspěšně použít.

[6] Asynchronous JavaScript and XML - `https://www.w3schools.com/xml/ajax_intro.asp`

[7] `http://giantbomb.com/`

což je výhodné hlavně protože se uživateli při návštěvě úvodní stránky nenačítá až příliš mnoho dat). Konkrétně se jedná o sloupcový graf deseti titulů s nejvíce příspěvky, kde každý sloupec je ještě rozdělen na 3 části podle postoje. Každá část zabírá zlomek sloupce, který odpovídá počtu příspěvků v dané kategorii postojů ku počtu celkových příspěvků k danému titulu. Uživatel má možnost vybrat, zda-li chce vidět u zmíněného grafu pouze příspěvky z Facebooku nebo pouze z Twitteru. Jelikož značná část příspěvků byla klasifikována jako irelevantní (neutrální), má uživatel také možnost tuto část skrýt a vidět tak pouze pozitivní a negativní části grafu. Případně knihovna *amCharts* zahrnuje méně intuitivní schopnost zakrytí určité části grafu kliknutím přímo na položku legendy.

Další statistikou, která se zobrazí po vstupu na úvodní stránku portálu je žebříček lokací, ze kterých byly zasílány příspěvky z Twitteru, s odpovídající mapu světa. Jelikož autoři tweetů[8] si mohou do lokace napsat cokoliv, jsou do statistiky zahrnuty pouze lokace, které se ve stažených datech vyskytly nejméně 10 krát. Tato skutečnost v některých případech velmi zesložiťuje úlohu přiřazení lokace správnému místu na mapě.

Vespod stránky se nachází sekce informací o konkrétním titulu, která po rozkliknutí nabízí seznam titulů. Uživatel mezi tituly může přepínat a po každém takovém přednutí portál zobrazí několik informací o hře či herní sérii a graf závislosti počtu příspěvků na čase (tzn. kolik příspěvků bylo zaznamenáno za určité časové období). K zobrazení informací o herním titulu je využita *GiantBomb API*, která pomáhá získávat informace z webového portálu *GiantBomb* (viz úvod této sekce), a to konkrétně obrázek, herní žánr anebo žánry, herní vývojáře, datum vydání a krátký popis hry, to vše související s vybraným titulem. K API je přistupováno pomocí PHP funkce `file_get_contents()` s url jako prvním parametrem, který reprezentuje dotaz na *GiantBomb API*, obsahuje detaily, které chceme ze serveru získat a obsahuje také klíč[9], který je potřeba při každém dotazu poskytnout. Už jsou zde využita i samotná analyzovaná data neboť pod grafem se uživateli zobrazí tabulka konkrétních příspěvků rozdělená na 3 sloupce podle kategorie postojů. Všechna data v tabulce jsou reálné analyzované příspěvky. Dále jsou zobrazeny počty příspěvků v každé kategorii postojů a jeich procentuelní poměr k celkovému počtu všech příspěvků k vybranému titulu.

Data na serveru jsou uložena v souborech typu JSON, z nichž některé dosahují velikostí i přes 20 MB. Pro jejich načtení do paměti je využívána standartní PHP funkce `json_decode()`, která zapříčiňuje několikanásobně větší zatížení paměti, než je velikost čteného souboru. Stává se tedy, že portál při zobrazování výsledků uživateli na krátký časový úsek nadměrně vytěžuje paměť serveru. Více viz sekce 5.4.3.

[8] Příspěvěk na sociální síti Twitter

[9] Uživatelský klíč získaný po vytvoření účtu na http://giantbomb.com/

Kapitola 5

Experimenty a srovnávání

V předchozích kapitolách bylo zmíněno několik metod, kterými se klasifikace postojů může efektivně provádět. V této kapitole se v sekci 5.1 blíže podíváme na úspěšnost a efektivnost zmíněných metod a otestujeme, do jaké míry záleží na zvolení vhodné trénovací sady a jakých výsledků se při použití různých sad dosáhlo. Poté si v následující sekci 5.2 řekneme, které přístupy jsou efektivní, jaké metody se prokázaly jako nejúspěšnější a jak tedy dosáhnout co nejlepších výsledků. V sekci 5.3 se následně podíváme na srovnání dat získaných analýzou názorů s reálnými daty z jiných zdrojů abychom tak ověřili váhu a relevanci získaných dat. Při provádění analýzy názorů bylo zjištěno několik problémů, které si blíže rozvedeme v sekci 5.4.

Bylo provedeno celkem 12 různých testů rozdělených do 4 podkategorií. V každé podkategorii se testovaly 3 metody, a to *Rozhodovací stromy* 2.3.1, *Naivní Bayesův klasifikátor* 2.3.2 a *Maximální entropie* 2.3.3. Zaměřili jsme se zejména na zkoumání přesnosti klasifikace postojů a rychlosti provedení analýzy.

Pro experimenty byl použit stažený *Twitter Sentiment Analysis Dataset* [9], který obsahuje přibližně 1 500 000 ohodnocených tweetů. Každý tweet je ohodnocen buď 1 - pozitivní nebo 0 - negativní. Stažený dataset však obsahuje mnoho tweetů, u kterých je jejich ohodnocení diskutabilní (např. *„omg its already 7:30 :O“*, *„u guys knw whyy“*, atp.) což mírně dataset znevažuje pro měření přesnosti našeho analyzátoru. Tento problém lze částečně vyřešit tím, že tweety, které náš analyzátor označí jako neutrální, nebudeme do výsledného hodnocení přesnosti zahrnovat. Existuje totiž velká šance, že právě tweety, které náš analyzátor ohodnotí jako neutrální, budou tweety jejichž postoj je diskutabilní.

Jelikož zpracování celého staženého datasetu by zabralo velké množství času i zdrojů, byly z něj vytvořeny 2 další datasety. *První testovací dataset* obsahuje přibližně 16 000 ohodnocených tweetů ze začátku staženého datasetu a *druhý testovací dataset* obsahuje 50 000 ohodnocených tweetů (ze staženého datasetu jsou to tweety následující za prvními přibližně 17 000 tweety). Tyto datasety byly v experimentech použity pro ohodnocení naším analyzátorem sentimentu a následným porovnáním přestosti.

Při experimentech byl pro trénování metod strojového učení použit můj vlastní dataset, který byl vytvořen přímo pro účely této práce, a tak i přes menší obsah (cca 160 ohodnocených tweetů) lze jeho použitím dosáhnout velmi dobrých výsledků. Dále byl použit dataset vytvořený ze 190 náhodně vybraných tweetů staženého datasetu. Bohužel metody z knihovny NLTK přestávaly fungovat při použití větších trénovacích datasetů – viz sekce 5.4.

5.1 Testování různých přístupů

Následující testy jsou rozděleny do 4 podkategorí. V každé je popsána odlišnot vůči ostatním podkategoriím a v každé byly testovány 3 vybrané metody strojového učení uvedené na začátku této kapitoly. U každého testu je uvedeno:

- identifikační číslo testu
- název metody strojového učení
- celkový počet zpracovaných příspěvků
- počet příspěvků klasifikovaných jako neutrální
- počet správně klasifikovaných příspěvků
- úspěšnost analyzátoru v procentech[1]
- doba, za kterou byla analýza provedena

Jednotlivé podkategorie jsou nazvány podle podmínek vytvořencých pro testování. Konkrétně se jedná o použité trénovací a testovací datasety. Jako trénovací datasety jsou použity: *můj trénovací dataset*, který jsem sestavil sám pro potřeby této práce a *vytvořený trénovací dataset*, který vznikl ze staženého datasetu. Oba testovací datasety byly vytvořeny taktéž ze staženého datasetu. Podrobnější informace o nich lze nalézt v úvodu této kapitoly.

5.1.1 Můj trénovací dataset a první testovací dataset

Testy č. 1 až 3. Kombinace použití mnou vytvořeného datasetu pro natrénování metody strojového učení a prvního testovacího datasetu pro analýzu a ásledné porovnání úspěšnosti.

Test č. 1 – Maximální entropie

Celkem příspěvků:	16685
Neutrálních:	10684
Správných:	4240
Úspěšnost:	70,64 %
Čas:	21,23 s

Test č. 2 – Naivní Bayesův klasifikátor

Celkem příspěvků:	16685
Neutrálních:	10684
Správných:	4249
Úspěšnost:	70,79 %
Čas:	16,22 s

[1] Je nutno dodat, že z důvodů uvedených na začátku této kapitoly, do měření úspěšnosti nezapočítáváme příspěvky klasifikované jako neutrální.

Test č. 3 – Rozhodovací stromy

Celkem příspěvků:	16685
Neutrálních:	10872
Správných:	3858
Úspěšnost:	66,36 %
Čas:	22,21 s

5.1.2 Můj trénovací dataset a druhý testovací dataset

Testy č. 4 až 6. Kombinace použití mnou vytvořeného datasetu pro natrénování metody strojového učení a druhého testovacího datasetu pro analýzu a následné porovnání úspěšnosti.

Test č. 4 – Maximální entropie

Celkem příspěvků:	50000
Neutrálních:	29257
Správných:	14810
Úspěšnost:	71,40 %
Čas:	48,88 s

Test č. 5 – Naivní Bayesův klasifikátor

Celkem příspěvků:	50000
Neutrálních:	29257
Správných:	14865
Úspěšnost:	71,66 %
Čas:	54,06 s

Test č. 6 – Rozhodovací stromy

Celkem příspěvků:	50000
Neutrálních:	29756
Správných:	11650
Úspěšnost:	57,55 %
Čas:	38,60 s

5.1.3 Můj trénovací dataset a druhý testovací dataset – bez negace

Testy č. 7 až 9. Kombinace použití mnou vytvořeného datasetu pro natrénování metody strojového učení a druhého testovacího datasetu pro analýzu a následné porovnání úspěšnosti. Nasledující testy byly provedeny bez zpracování negace 2.2.5 zejména proto, abychom zjistili, jak velkou roli při klasifikaci postojů hraje správné zpracování negace.

Test č. 7 – Maximální entropie

Celkem příspěvků:	50000
Neutrálních:	23573
Správných:	17757
Úspěšnost:	67,19 %
Čas:	48,16 s

Test č. 8 – Naivní Bayesův klasifikátor

Celkem příspěvků:	50000
Neutrálních:	23574
Správných:	16835
Úspěšnost:	63,71 %
Čas:	58,81 s

Test č. 9 – Rozhodovací stromy

Celkem příspěvků:	50000
Neutrálních:	24072
Správných:	17244
Úspěšnost:	66,51 %
Čas:	35,43 s

5.1.4 Vytvořený trénovací dataset a druhý testovací dataset

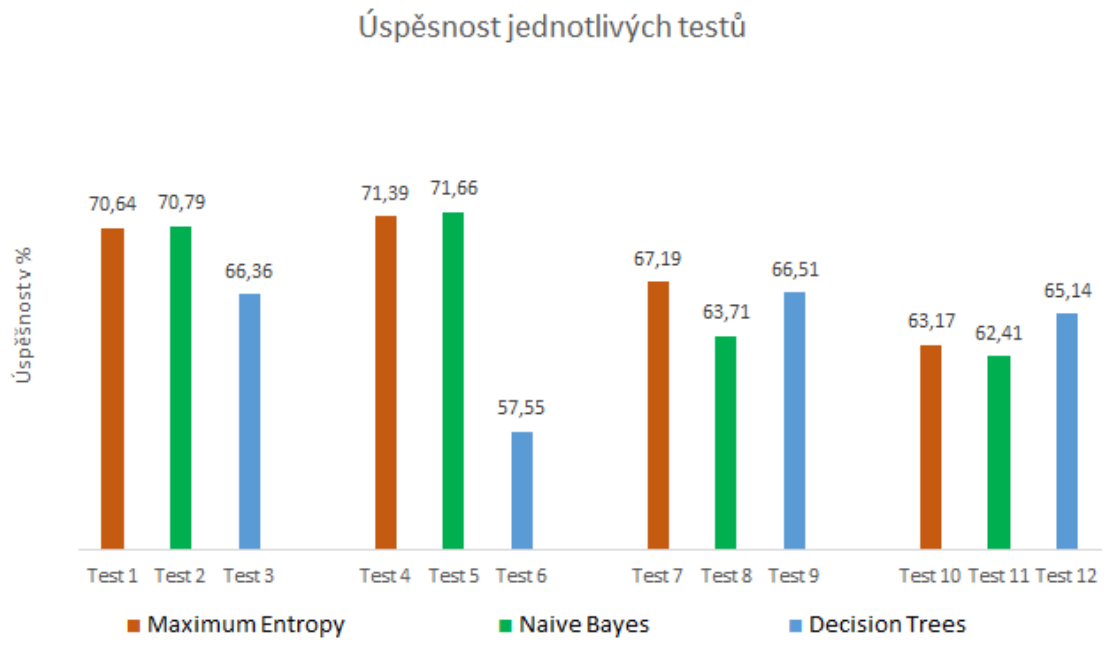
Testy č. 10 až 12. Kombinace použití trénovacího datasetu vytvořeného ze staženého datasetu a druhého testovacího datasetu pro analýzu a následné porovnání úspěšnosti.

Test č. 10 – Maximální entropie

Celkem příspěvků:	50000
Neutrálních:	15751
Správných:	21639
Úspěšnost:	63,17 %
Čas:	2 min 5,98 s

Test č. 11 – Naivní Bayesův klasifikátor

Celkem příspěvků:	50000
Neutrálních:	12228
Správných:	23572
Úspěšnost:	62,41 %
Čas:	2 min 37,07 s



Obrázek 5.1: Graf úspěšnosti jednotlivých testů

Test č. 12 – Rozhodovací stromy

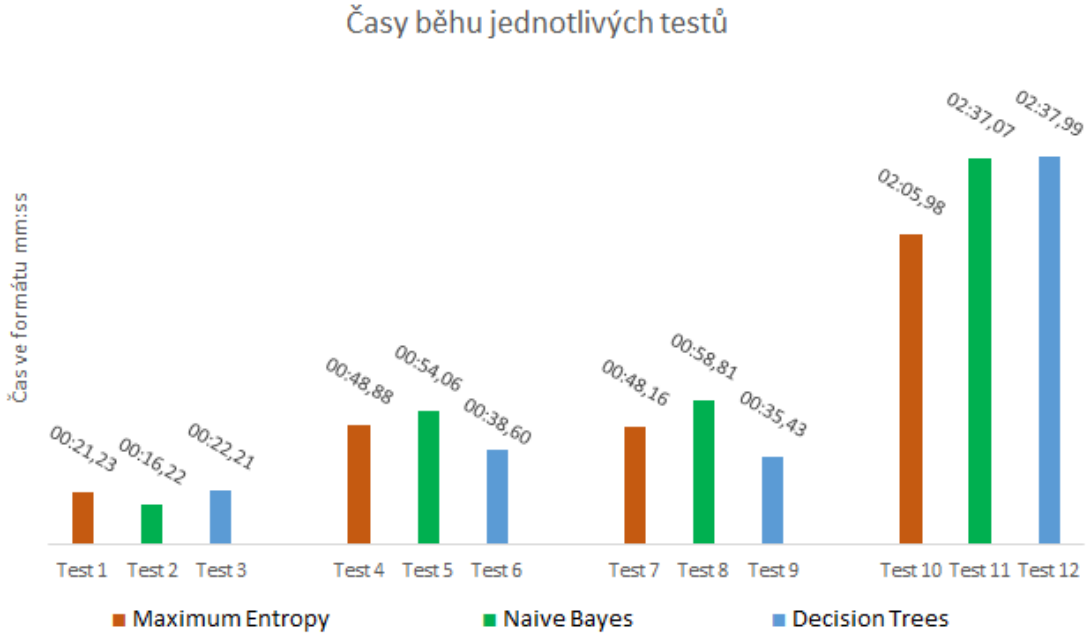
Celkem příspěvků:	50000
Neutrálních:	17790
Správných:	20981
Úspěšnost:	65,14 %
Čas:	2 min 37,99 s

5.2 Zhodnocení testů

Experimentováním a prováděním testů bylo získáno hned několik užitečných poznatků, které budou v této sekci zmíněny. Nejprve se podíváme na výsledky testů z širší perspektivy a poté si je rozebereme podrobněji. K dispozici jsou taktéž grafy 5.1, který zobrazuje a porovnává úspěšnost všech provedených testů, a 5.2 zobrazující časy jednotlivých provedených testů.

Úspešnost testovaných případů se pohybuje v rozmezí od 55 % do 75 % (minimum: Test č. 6 s 57,55 %; maximum: Test č. 5 s 71,66 %). Analyzátor sentimentu dokázal ve všech testech správně vyhodnotit více než polovinu příspěvků, což je úspěch byť né příliš veliký neboť úspěšnost se příliš neblíží očekávané hranici 80 %.

Z grafu 5.2 lze vypozorovat, že klasifikace pomocí *Naivního Bayesova klasifikátoru* trvá znatelně déle při zpracování většího množství textů. V tomto ohledu je naopak metoda *Maximální entropie* rychlejší než ostatní. Lze si také povšimnout značného skoku u poslední podkategorie testů. Použitý dataset sice obsahuje pouze o 30 anotovancých příspěvků více, ale je nutno podotknout, že texty v tomto datasetu obsahují mnohem více slov, tudíž jejich zpracování a následná klasifikace trvají znatelně delší dobu.



Obrázek 5.2: Graf časů běhu jednotlivých testů

Dále si lze povyšimnout, že značná část příspěvků (přes 20 % a při použití mého trénovacího datasetu okolo 50 %) byla klasifikována jako neutrální, což je způsobeno zejména diskutabilností ohodnocených textů ve staženém datasetu (viz začátek této kapitoly).

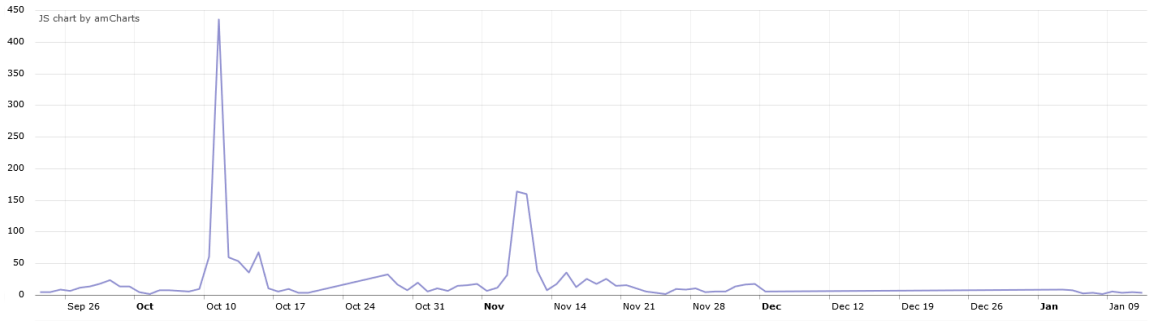
Testy prováděné v podkategorii, kde nebyla nijak zpracována negace, trvají přibližně stejnou dobu jako testy, kde negace zpracována byla. Rozdíl lze však vidět v úspěšnosti těchto testů, která je podle očekávání nižší (tedy s vyjímkou metody Rozhodovací stromy, která naopak projevyla výrazné zlepšení přesnosti). Zpracování negace v našem analyzátoru sentimentu tedy přispívá ke kvalitnějším výsledkům. Bylo by však možné zpracování negace ještě zdokonalit a získat tak o několik procent kvalitnější výsledky, ovšem za cenu podstatně větší námahy a tedy i času. Usoudil jsem, že by se takový obchod přiliš nevyplatil a spokojil jsem se s řešením negace tak, jak je použito v této praci.

Zprůměrujeme-li výsledné úspěšnosti dílčích metod získáme následující výsledky:

- *Maximální entropie* – 68,10 %
- *Naivní Bayesův klasifikátor* – 67,14 %
- *Rozhodovací stromy* – 63,89 %

Na uvedených datech lze vidět, že analyzátor sentimentu produkuje nejkvalitnější výsledky pomocí metody *Maximální entropie* a nebo *Naivní Bayesův klasifikátor*, přičemž zpracování metodou *Maximální entropie* prokazuje nepatrně lepší úspěšnost a při klasifikaci rozsáhlých textů také zabere kratší čas.

Z provedených testů tedy vyplývá, že pro co možná nejlepší výsledky implementovaného analyzátoru je potřeba zvolit jako trénovací sadu *Můj trénovací dataset* a klasifikaci provádět pomocí metody *Maximální entropie*.



Obrázek 5.3: Časový průběh příspěvků k titulu *Ashes of the Singularity* analyzovaných naším systémem

5.3 Ověřování výsledných dat

Sekce se zabývá porovnáváním získaných dat s již existujícími daty a skutečnostmi z jiných zdrojů jimiž jsou např. články, recenze apod. Snažíme se zde prokázat na konkrétních příkladech, že naše získaná data opravdu odpovídají skutečnosti. Bylo vybráno několik konkrétních titulů, na nichž lze snadno vidět a také prokázat spojitost s informacemi dostupnými z jiných zdrojů, které jsou také uvedeny v literatuře na konci této práce.

5.3.1 Ashes of the Singularity

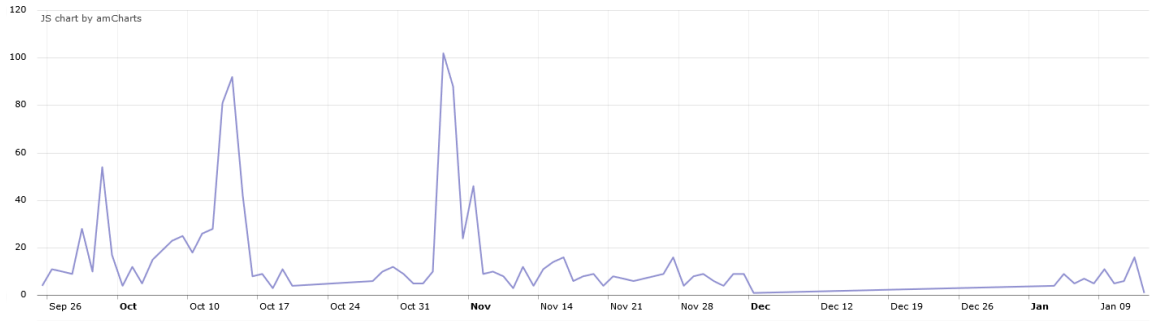
Při nahlédnutí do grafu 5.3 můžeme spatřit 2 výrazné vrcholy. První výraznější vrchol připadá na den 11. října 2016, kdy byla vydána ukázka (angl. *trailer*) na rozšíření uvedené hry. To však vyvolalo pouze vlnu sdílení vydané ukázky, ale příspěvků vyjadřujících relevantní postoje bylo pramálo. Oproti tomu druhý vrchol grafu ze dnů 10. a 11. listopadu 2016 reprezentuje z drtivé většiny kladné příspěvky natěšených uživatelů, které obsahovaly první dojmy z hraní nebo vypovídaly o tom, že už se nemohou dočkat, až si zahrají nové rozšíření *Ashes of the Singularity: Escalation*, které bylo vydáno v tento den.

5.3.2 Brütal Legend

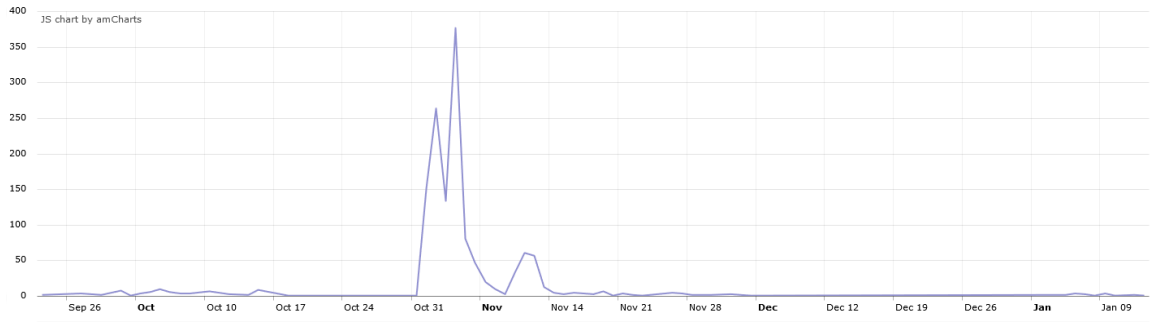
U uvedeného titulu můžeme v grafu 5.4 opět vidět 2 významné vrcholy. První vrchol je pro nás celkem nezajímavý. Jedná se o období, kdy byla hra ve slevě. Mnohem zajímavější je však druhý výrazný vrchol grafu připadající na 4. listopadu 2016. V tento den totiž v čase 9:55 uživatel *@brutallegend*, oficiální reprezentant uvedené hry *Brütal Legend* na sociální síti *Twitter*, po více než 2 letech neaktivity zveřejnil příspěvek, ve kterém stálo „Ormagöden Lives.“. Tento příspěvek zejména na sociální síti *Twitter* rozproudil diskuse. Mnoho fanoušků bylo nadšených, zda-li se chystá pokračování, kdežto menší část uživatelů pojednávala o tom, zda-li jde opět o nějaký podlý marketingový tah, kterých vydavatel této hry (dle získaných příspěvků) provedl v minulosti několik.

5.3.3 House of the Dying Sun

Kromě viditelných vrcholů v grafu 5.5 ke hře přibyly přibližně 4 příspěvky denně. Hra byla oficiálně vydána 2. listopadu 2016 a ve dnech 1. až 5. listopadu 2016 bylo zaznamenáno až 1000 příspěvků. Uživatelé projevily o hru velký zájem a zalíbení. Po těchto dnech ovšem počty přibývajících příspěvků značně klesly, a můžeme se tedy domnívat, že ačkoliv se



Obrázek 5.4: Časový průběh příspěvků k titulu *Brütal Legend* analyzovaných naším systémem



Obrázek 5.5: Časový průběh příspěvků k titulu *House of the Dying Sun* analyzovaných naším systémem

v době vydání hry její fanoušci značně projevili, řadí se *House of the Dying Sun* k nepříliš populárním hrám.

5.3.4 Pikmin

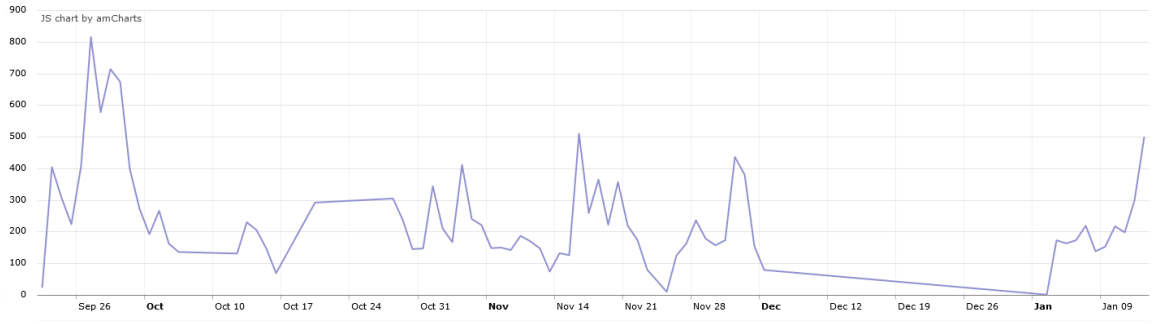
Tato hra se poprvé objevila v roce 2001 na konzoli *GameCube* a měla 2 další zdařilé díly. Dne 26. září 2016 vývojář této hry *Nintendo* uveřejnil, že plánuje její vydání na platformu *Wii U*, a to hned tři dny poté, tedy 29. září 2016 [3]. Na grafu 5.6 můžeme vidět nadšení fanoušků (s vrcholem dne 27. září 2016 s 816 příspěvky) ve formě výrazného počtu příspěvků ještě několik dní po uveřejnění výše uvedené zprávy. Jednalo se nejen o příspěvky plné nadšení pro tuto hru, ale také i pro její následná pokračování.

5.4 Zjištěné problémy a nedostatky

Během tvorby systému, provádění analýzy a následném experimentovaní bylo zjištěno několik problémů a nechtěných situací, které by bylo vhodné v této práci zmínit. Jedná se o těžko ovlivnitelné problémy, ale taktéž o oblasti, kde lze práci ještě vylepšit.

5.4.1 Víceznačná klíčová slova

Na sociálních sítích vyhledáváme data pouze pomocí klíčových slov, která jsou v našem případě reprezentována názvy konkrétních her nebo herních sérií. Vyhledávání požadovaných



Obrázek 5.6: Časový průběh příspěvků k titulu *Pikmin* analyzovaných naším systémem

dat na sociálních sítích pomocí našeho systému funguje tak, že jsou vyhledány všechny příspěvky, které někde v sobě obsahují klíčové slovo, které k vyhledávání zrovna používáme. Může se stát, že některé názvy herních titulů nebudou jednoznačné, bude se jednat o příliš obecné výrazy či slova a tím pádem získáme i nechtěná data vztahující se k těmto pro nás nezajímavým tématům. Z hlediska počítačových her se tento problém vztahuje například k titulům *Ares*, *Allegiance*, *Majesty* apod. Jedná se sice o názvy her, ale tato slova jsou používána na sociálních sitích i ve zcela jiném kontextu. *Ares* je také řecký bůh války, *Allegiance* je také anglický výraz pro „oddanost“ a *Majesty* je anglický výraz pro „výsost“ nebo „veličenstvo“. Např. při použití klíčového slova *Majesty* nás z hlediska analýzy postojů k počítačovým hrám zajímají příspěvky hovořící o stejnojmenné strategické hře a nikoliv příspěvky související s anglickou královnou či jinou šlechtou.

Při zhodnocování výsledků je tedy nutno brát na vědomí, že získaná data u určitých titulů mohou být infikována nesouvisejícími postoji, které budou mít zavádějící vliv na výsledné informace o daném titulu. Tento problém se však nevyskytuje pouze u domény počítačových her, ale může se objevit i při zpracování jiných témat, a to např. u značek automobilů (automobil Škoda vs české slovo „škoda“).

5.4.2 Omezená velikost trénovací sady

Problém se vztahuje ke skriptu pro analýzu stažených dat. Při klasifikaci postojů jsou využívány funkce a metody knihovny *NLTK*. Experimentováním bylo zjištěno, že při pokusu použít obsáhlejší trénovací sady, tím je myšleno trénovací sady s více než 300 řádky, nejen že analýza trvala značně delší dobu, ale hlavně metody *Maximální entropie* a *Rozhodovací stromy* při klasifikaci postojů nebyly příliš úspěšné (testy dostahovaly úspěšnosti okolo 50 %) a v některých případech dokonce ohodnotily všechny vstupní texty jako neutrální. Tento jev je také doprovázen konzolovým varováním knihovny *numpy*:
`RuntimeWarning: overflow encountered in power` a případně dalšími podobnými upozorněními. Jedná se o situaci, kdy je z důvodu zpracovávání příliš velkého počtu slov při trénování metody velmi obtížné vypočítat určité potřebné části a mezivýpočty načež se potýkáme s přetečením (angl. *overflow*) hodnot daného typu.

5.4.3 Paměťové nároky webového portálu

Jak je již uvedeno v sekci 4.3, použití PHP funkce `json_decode()` pro dekódování obsahu souborů způsobuje několikanásobně větší zatížení paměti serveru než je velikost čteného souboru na serveru. Tato skutečnost způsobila, že při lokálním testováním portálu na vlast-

ním stroji bylo potřeba navýšit poskytovanou paměť z 64 MB na 128 MB, a to pouze pro přečtení souboru se statistikami o velikosti 7,3 MB. Zanedlouho poté bylo zjištěno, že tato velikost paměti je také málo pro přečtení souboru o velikosti 20 MB. Problém ustal až po přidělení paměti 256 MB webovému portálu.

Je tedy potřeba si uvědomit, že při prohlížení webového portálu pro zobrazení výsledků uživateli, je na krátký časový okamžik vytížena část paměti serveru (nepřesahující 256 MB). Tento problém nastává pouze při použití výše uvedené PHP funkce a bylo by možné jej vyřešit použitím různých jiných JSON parserů, které jsou ovšem náročnější na implementaci nebo i na práci se vzniklými objekty.

Kapitola 6

Závěr

Cílem této práce bylo vytvořit systém pro stahování dat, analýzu názorů a následné zobrazení zpracovaných dat uživateli. Práce byla zaměřena konkrétně na postoje vůči počítačovým hrám. Systém byl vytvořen a je funkční, včetně všech jeho částí, které zahrnují skripty na přípravu textu a analýzu názorů a webový portál pro zobrazení výsledků uživateli. Lze tvrdit, že určené podcíle práce byly splněny, ovšem je zde stále prostor pro zlepšování, jelikož skripty pro získávání dat byly vytvořeny pouze pro sociální sítě *Twitter* a *Facebook* a webový portál by mohl být rozšířen například o určité vyhledávací rozhraní jako např. vyhledávání titulů podle vývojáře. Pomocí testů z kapitoly 5 bylo zjištěno, že implementovaný analyzátor sentimentu při zvolení vhodné trénovací sady a klasifikační metody dosahuje přesnosti přibližně 71 %, což je uspokojivý výsledek. V dnešní době mají však jiné analyzátory sentimentu nasazenou laťku úspěšnosti výše (např. [14]), ovšem je třeba přihlédnout ke skutečnosti, že v této práci byla prováděna analýza dat ze sociálních sítí, čili analýza příspěvků z Twitteru a Facebooku, což mnohdy nelze přímo srovnávat např. s analýzou recenzí. Systém je samozřejmně možné použít i k analýze jiných témat než jsou počítačové hry, ovšem je nutné si uvědomit, že použitá trénovací sada byla vytvořena speciálně pro analýzu témat týkajících se počítačových her. Při analýze jiných témat by tedy bylo dobré kromě seznamu klíčových slov také vhodně upravit i trénovací sadu.

Systém je zajímavý zejména z hlediska flexibility. Tím jsou myšleny možnosti jeho použití pro jakákoliv témata, upravitelnosti trénovací sady dle vlastních potřeb, možnosti výběru, kterou klasifikační metodu (ze tří implementovaných) chceme použít a která slova při zpracování textu chceme odfiltrovat.

Z rozebraných metod klasifikace textu byly vybrány a implementovány *Tokenizace* 2.2.1, *Filtrace nepotřebných slov* 2.2.2, *Negace* 2.2.5 a z metod strojového učení *Rozhodovací stromy* 2.3.1, *Naivní Bayesův klasifikátor* 2.3.2 a *Maximální entropie* 2.3.3. Výsledný systém tedy ke své správné funkčnosti potřebuje kromě svých zrojových souborů také *seznam klíčových slov* obsahující herní tituly na jednotlivých řádcích, *seznam nepotřebných slov* nebo-li *seznam stopwords* využívaný při filtraci nepotřebných slov a *trénovací sadu*, kterou používají metody strojového učení při klasifikaci postojů.

Způsob zpracování textu *Lemmatizace* 2.2.3 nebyl implementován z důvodu potřeby korpusu, slovníku nebo seznamu, který by obsahoval jednotlivá slova, jejich různé tvary a jim odpovidající lemmy. V rámci této práce by se jednalo o nadměrně velké zatížení analyzátoru sentimentu za příliš malou a neúměrnou cenu možného zlepšení. Dalším důvodem je skutečnost, že by takto zpracovaná slova mohla zcela pozbýt svého významu což by mohlo vést k chybné klasifikaci postojů. *POS tagging* 2.2.4 byl zcela vynechán jelikož přínos tohoto kroku k analýze názorů, pokud vůbec nějaký, by byl nejspíše pramalý a stejně jako

u lemmatizace by se jednalo o zbytečně větší zatížení analyzátoru a souborů určených pro zpracování analyzátorem.

Systém lze vylepšit, a to například implementací výše zmíněných částí, které byly vynechány z uvedených důvodů, implementací dalších tradičních či netradičních způsobů zpracování textu a klasifikace a nebo rozšířením trénovací sady o další vhodná data. Vývoj v oblasti rozpoznávání přirozeného jazyka postupuje neustále dál, takže i přes uvedené příklady je zde prostor k dalšímu zlepšování.

Literatura

[1] Bing, L.: *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, Květen 2012.

[2] Foundation, P. S.: *Welcome to Python.orgl*. 2017, [Online; navštíveno 06.03.2017]. URL https://www.python.org/

[3] Ganos, J.: *New Play Control! Pikmin coming to the Wii U eShop on September 29th | Nintendo Wire*. 2016, [Online; navštíveno 10.04.2017]. URL http://nintendowire.com/news/2016/09/26/new-play-control-pikmin-coming-wii-u-eshop-september-29th/

[4] Lazăr, M.; Militaru, D.: *The Role of Decision Trees in Natural Langugage Processing*. SISOM & ACOUSTICS, 2015.

[5] Manning, C. D.; Raghavan, P.; Schütze, H.: *An Introduction to Information Retrieval*. Cambridge University Press, 2009.

[6] Manning, C. D.; Schütze, H.: *Foundation of Statistical Natural Language Processing*. The MIT Press, Cambridge, 1999, ISBN 0-262-13360-1.

[7] Merriam-Webster: *Sentiment | Definiton of Sentiment by Merriam-Webster*. [Online; navštíveno 19.02.2017]. URL https://www.merriam-webster.com/dictionary/sentiment

[8] Murphy, K. P.: *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, 2012, 1–25 s.

[9] Naji, I.: *Twitter Sentiment Analysis Training Corpus (Dataset) | Thinknook*. 2012, [Online; navštíveno 17.03.2017]. URL http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

[10] Otto, J.: *Ottův slovník naučný*. J. Otto, 1904, svazek 22. Rozkošný-Schloppe.

[11] Pang, B.; Lee, L.: *Opinion mining and sentiment analysis*, ročník 2. Foundations and Trends in Information Retrieval, 2008, 1–135 s.

[12] Potts, C.: *Sentiment Symposium Tutorial*. 2011, [Online; navštíveno 21.02.2017]. URL http://sentiment.christopherpotts.net/

[13] Rajaraman, A.; Ullman, J. D.: *Mining of Massive Dataasets*. Cambridge University Press, 2011, ISBN 9781139058452, 1–17 s.

[14] Sychra, M.: *Analýza sentimentu s využitím dolování dat*. Diplomová práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2016, vedoucí práce Bartík Vladimír.

[15] Wikipedia: Exponential backoff — Wikipedia, The Free Encyclopedia. 2016, [Online; navštíveno 08.03.2017].
URL https://en.wikipedia.org/w/index.php?title=Exponential_backoff&oldid=747705118

[16] Wikipedia: *Support vector machine — Wikipedia, The Free Encyclopedia*. 2016, [Online; navštíveno 28.02.2017].
URL https://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=753894433

[17] Wikipedia: *Sentiment analysis — Wikipedia, The Free Encyclopedia*. 2017, [Online; navštíveno 20.02.2017].
URL https://en.wikipedia.org/w/index.php?title=Sentiment_analysis&oldid=765738069