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

Diplomová práce

Automatické rozpoznávání (analýza) sentimentu

Plzeň, 2012 Bc. Michal KOKTAN

Prohlášení

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

V Plzni dne 10. 7. 2012 .....

Bc. Michal Koktan

Abstract

This work deals with automatic sentiment analysis. It has been developed for needs of Czech News Agency (CTK) and the data are provided from the ČTK database. The main goal of this work is to automatically classify articles to a class of sentiment (positive, negative or neutral). The sentiment analysis is viewed as a special case of the document classification task. We take into account the possibilities of the parallel processing. Three main classifiers are (Naive Bayes, Maximum Entropy and Support Vector Machine). The obtained accuracies are evaluated and compared. The best recognition score is 72% and are given by the SVM classifier. This result is very promising for the future real case of the ČTK.

Seznam obrázků

Seznam tabulek

Obsah

1 Úvod

Tato diplomová práce byla zadána Českou tiskovou kanceláří. Pojmem „sentiment“ je uvažován názor (hodnocení) uživatele na daný objekt (případně na určitou osobu). Tento objekt může být ohodnocen kladně, záporně nebo neutrálně. Dokumenty obsahující sentiment se vyskytují zejména v internetových diskuzích, kde zákazníci hodnotí nějaké zboží, nebo ve fórech o filmech, kde komentují kvalitu filmu.

Na základě studie a dostupné literatury je úloha automatické analýzy sentimentu pojata jako specifický případ úlohy automatické klasifikace dokumentů do tříd úrovně sentimentu. V rámci této práce jsou také řešeny možnosti paralelního zpracování dokumentů

Struktura práce

V první části práce se čtenář seznámí se sentimentální analýzou a klasifikací dokumentů. Tato kapitola obsahuje s analýzou a klasifikací související metody klasifikace. Další část se zabývá paralelním zpracováním textu a strukturou dat poskytnutých. Šestá kapitola se zabývá dostupnými klasifikačními nástroji, které lze použít pro analýzu sentimentu. V dalších kapitolách se práce zabývá vlastním řešením diplomové práce a to vytvořením korpusu a klasifikací dokumentu. V části práce o klasifikaci dokumentů se vyskytují informace o potřebném předzpracování vstupních dat pro klasifikační nástroj a dále samotná úprava nástroje pro účely diplomové práce. V této kapitole se čtenář dozví o paralelní klasifikaci a hledání optimálních nastavení parametrů. Předposlední kapitola se zabývá dosaženými výsledky experimentů. Výsledky jsou znázorněny přehledně v podobě tabulek a grafů. Tato kapitola obsahuje srovnání úspěšnosti klasifikace při použití klasifikátorů samostatně nebo při použití paralelního zpracování. V poslední kapitole jsou shrnuty dosažené výsledky a navrženy případná další rozšíření.

2 Klasifikace a analýza sentimentu

V této části jsem se zaměřil na termín sentimentu, jak probíhá jeho analýza a na dostupné metody, které se mohou použít při analýze. V další části jsem se zaměřil na dostupná data, která jsou potřebná a jejich zpracování pro analýzu.

Sentiment vyjadřuje lidský pocit nebo názor na nějaký podnět [1]. Sentiment vždy vyjadřuje nějakou náklonnost nebo odpor, který není závislý na samostatné úvaze. Vždy vyvolá hodnocení do dané kategorie pozitivnosti či negativnosti, ale nepodléhá vědomé kontrole. Dle sentimentu můžeme dokumenty klasifikovat na pozitivní, negativní a neutrální.

2.1 Analýza sentimentu

2.1.1 Historie sentimentu

Oblasti výzkumu sentimentální analýzy nebo také získávání názorového vnímání se rozvíjí do roku 2001. Do tohoto roku se výzkum zaměřil na výklad metafory, vyprávění a názory v obsahu textu a v souvisejících oblastech.

Rok 2001 se označuje jako začátek uvědomování si výzkumných problémů a příležitostí, které sentimentální analýza a názorové vnímání přináší. Následně byly publikovány stovky novinových článků s předmětem sentimentu.

Dále budou uvedeny důvody, které zapříčinily tento vývoj:

- vzestup metod strojového učení v přirozeném jazyce,
- dostupnost datových množin algoritmy strojového učení, které mohou být učeny z rozvoje webu,
- realizace myšlenkových výzev a komerčních aplikací umělé inteligence.

2.1.2 Charakteristika a použití

Analýza sentimentu spočívá v automatickém (počítačovém) zpracování přirozeného jazyka v zadaných dokumentech s cílem určení postoje komentátora nebo zapisovatele s ohledem na zadané téma.

Sentiment nalezený uvnitř komentářů, v odezvách nebo v nějaké kritice poskytuje užitečné indikátory pro mnoho účelů. Dokumenty mohou být z hlediska sentimentu

klasifikovány buď do dvou kategorií (pozitivní nebo negativní) nebo také do n-kategorií (např. pro n=5: velmi dobrá, dobrá, uspokojivá, špatná, velmi špatná). V tomto ohledu může být analýza sentimentu popisována jako úkol třídění, tedy zařadit daný dokument do předem zadaných kategorií sentimentu.

Zdrojem sentimentu mohou být především blogy nebo fóra vyjadřující něčí názor. Dalším dobrým zdrojem sentimentu jsou recenze filmů a komentáře u jednotlivých filmů ve filmových databázích.

Analýza sentimentu může poskytovat společnostem z hodnocení zákazníků odhady rozsahu přijetí trhem daného výrobku, aby se mohla například zjistit kvalita produktu. Analýza sentimentu také poskytuje taktiku výrobcům nebo politikům k analyzování veřejného mínění s ohledem na politiku, veřejné služby či politické problémy.

V současné době se zabývá výzkumem analýzy sentimentu 20 až 30 společností v USA [3].

2.1.3 Automatická klasifikace dokumentů

Automatická klasifikace dokumentů je proces, při kterém třídíme dokumenty do jednotlivých tříd dle daného kritéria. Třídy můžeme určovat například binární klasifikací, která rozlišuje pouze dvě třídy a to pozitivní a negativní. Tato úloha se nazývá sentimentální klasifikace polarity. Pod pojmem pozitivní si můžeme představit věty, kde se vyskytují fráze jako je „líbí se“ a pod negativním „nelíbí se“, ale je spousta dalších problémů, kdy se nedá jednoduše určit, kam zadaný text zařadit [1].

Klasifikovat novinový článek přímo do dobrých nebo špatných zpráv bylo dříve považováno za klasifikaci sentimentu v literatuře. Části zpráv ale mohou být dobré nebo špatné i bez subjektivního názoru autora. Jako příklad si můžeme uvést: „Ceny akcií stoupají“ – tato informace je obecně považována za dobrou zprávu. V jiných souvislostech se ale může jednat o zprávu špatnou. Dalším úkolem je určování, zda část informace je subjektivní nebo objektivní informací, protože rozdíl mezi těmito informacemi je velice choulostivý. Jako dva protipříklady můžeme posoudit „baterie vydrží dvě hodiny“ a „baterie vydrží jen dvě hodiny“.

2.2 Metody klasifikace

Ke klasifikaci dokumentů dle sentimentu budou na základě doporučení v nastudované literatuře [1] a [2] použity následující tři metody.

K implementaci těchto algoritmů strojového učení se používají následující vlastnosti:

- Nechť $f_1, ...f_m$ je předdefinovaná sada m vlastností, které se mohou objevit v dokumentu.
- Nechť $n_i(d)$ je číslo kolikrát se f_i vyskytuje v dokumentu d.

Potom každý dokument d je reprezentován jako vektor:

$$\vec{d} := (n_1(d), n_2(d), ..., n_m(d)) \tag{1}$$

2.2.1 Naivní Bayesův klasifikátor

Metoda byla prokázána jako velmi účinná pro klasifikaci textů vzhledem k vlastnostem (inkrementální učení, díky kterému je možno trénovat větším množstvím dat, jednoduchost a rychlost klasifikace)[2].

Úloha klasifikace textových dokumentů je úloho, kde se přiřadí danému dokumentu d třída c:

$$\hat{c} = \arg\max_c P(c|d) \tag{2}$$

kde c je množina všech možných tříd (v našem případě se jedná o třídy sentimentu).

Pro podmíněnou pravděpodobnost $P(c|d)$ je použita Bayesova věta:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{3}$$

kde $P(c)$ je apriorní pravděpodobnost výskytu dokumentu v třídě c, která se určí z trénovacího korpusu. Hodnota pravděpodobnosti $P(d)$ nehraje roli s maximalizací a bude proto dále vynechána.

Pro odhad pravděpodobnosti $P(d|c)$, je nahrazen následujícím rozkladem, který je možný za předpokladu vzájemné nezávislosti příznaků dokumentu.

Rozpoznávací třída je pak definována jako:

$$\hat{c} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(f_i|c). \tag{4}$$

kde n je počet slov v dokumentu a f_i jsou hodnoty příznaků, které popisují daný dokument.

2.2.2 Maximální entropie

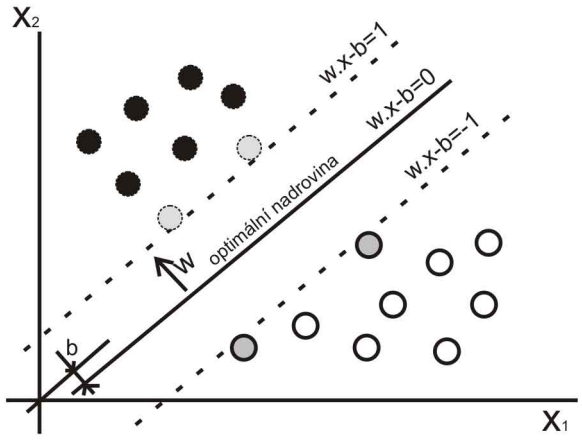
Klasifikátor maximální entropie (MaxEnt) je alternativní algoritmus, který dokázal být účinný v mnoha aplikacích zabývajících se automatickým jazykovým zpracováním. Ukázal se být lepší ve standartní textové klasifikaci než NB klasifikátor, ale ne vždy. Metoda se také používá ke strojovým překladům nebo ke značkování textu [2].

Pokud máme vypočtena všechna omezení, model s maximální entropií máme jednoznačně zaručený. Odhad pravděpodobnosti $P(c|d)$ má exponenciální rozdělení, které ukazuje následující vzoreček:

$$P_{ME}(c|d) := \frac{1}{Z(d)} \exp \sum_i \lambda_i f_i(d, c) \tag{5}$$

kde $Z(d)$ je normalizační funkce pro správné zjištění pravděpodobnosti. $f_i(d, c)$ je příznakový model a λ_i odhadovaný parametr. Normalizační funkce se vypočítá:

$$Z(d) := \sum_c \exp\left(\sum_i \lambda_i f_i(d, c)\right) \tag{6}$$



Obrázek 1: Nalezená optimální nadrovina v prostoru příznaků.

Na rozdíl od NB, MaxEnt nevytváří žádné předpoklady o vztazích mezi vlastnostmi a tím by mohl MaxEnt klasifikovat lépe, pokud se předpoklad nezávislosti nijak nepotkal s předem danými vlastnostmi.

Základní myšlenkou je, že by se měl vybrat model vytvářející nejmenší počet předpokladů o datech a zároveň zůstává v souladu s modelem, který dává intuitivně smysl [2].

2.2.3 Support Vector Machine

Klasifikátor Supprot Vector Machine (SVM) je dle dostupné literatury také vysoce efektivní v úloze kategorizace textů. Oproti NB a MaxEnt má širší možnosti při nalezení hranic mezi jednotlivými třídami [2].

Každý příklad reprezentuje příznaky a tyto příznaky jsou potom promítány do prostoru. V binárním klasifikování je základním principem metody nalezení nadroviny reprezentované vektorem $\vec{w}$, který odděluje od sebe množiny dat. Cílem je najít minimální vzdálenost bodů od nadroviny, aby vzniklé množiny byly maximální. Nalezení optimální nadroviny je znázorněno na obrázku 1.

Šedivě obarvené příznaky jsou podpůrné vektory, které určují optimální nadrovinu.

Optimální nadrovina má potom v prostoru rovnici:

$$w.x–b = 0 \tag{7}$$

Příznaky, které spadají do jedné třídy (na obrázku 1 černá kolečka) pak splňují podmínku pro jeden okraj nadroviny $w.x - b = 1$ a tím všechny příznaky musí splňovat podmínku:

$$w.x–b \geq 1 \tag{8}$$

Pro druhou třídu (na obrázku 1 bílá kolečka) má okraj nadroviny rovnici $w.x–b = -1$ a tím všechny příznaky splňují podmínku:

$$w.x–b \leq -1 \tag{9}$$

3 Paralelní zpracování textu

Během studie literatury nebyly nalezeny žádně paralelní metody pro automatickou analýzu sentimentu a celkově se příliš nepoužívají metody pro paralelní zpracování textu. Z prostudovaných metod stojí za zmínku zejména následující metoda, kde se používá algoritmus Expectation-Maximization. Tento algoritmus bude podrobně popsán dále v obou variantách bez paralelizmu i s paralelismem.

Textová klasifikace je proces klasifikování dokumentů do předdefinovaných kategorií založených na jejich obsahu. Paralelní zpracování textových dokumentů vede k rychlejšímu dosažení požadovaných výsledků.

Bylo vymyšleno mnoho algoritmů pro automatickou textovou klasifikaci. Tyto algoritmy obvykle zpracovávají velké množství označkovaných dokumentů pro trénování. Není ale žádný algoritmus, který by zpracovával označkované a neoznačkované dokumenty společně. Expectation-Maximization (EM) algoritmus zpracovává zároveň označkované i neoznačkované dokumenty pro učení [9]. Tento algoritmus, je bohužel velice pomalý, pokud se mu předloží veliké množství dokumentů.

3.1 Expectation-Maximization Algoritmus

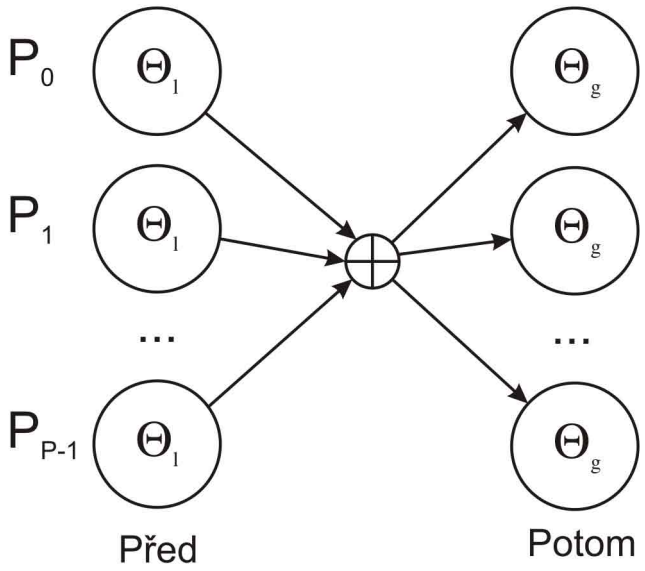
EM algoritmus je obecná technika pro zjišťování maximální pravděpodobnosti nebo pro pozdější zjišťování maximálního odhadu v neúplných datech. Princip a více informací o algoritmu lze nalézt v [9].

3.2 Paralelní implementace EM Algoritmu

Pro paralelizaci se využívá modelu SPMD (Single Program Multiple Data). Zdrojový program je napsán a poté rozšířen na ostatní procesory, na kterých běží kopie programu. Předpokladem je, že je k dispozici P procesorů, kde každému procesoru se zadá unikátní číslo mezi 0 a $P - 1$ [1] a přidělí paměť. Procesory mezi sebou komunikují používáním MPI (Message Passing Interface) knihovny.

Algoritmus začíná tím, že se použije Naivní Bayesův klasifikátor k inicializaci parametrů. Kroky probíhají v iteracích, dokud změna MAP je menší než některá před-

[1] Unikátní číslo je zadáno z důvodu, aby se vědělo, jaký procesor právě provádí operace a přistupuje do paměti.



Obrázek 2: Nastavování parametrů Theta v iteracích EM algoritmu. Zdroj [9]

definovaná hodnota prahu. Jednotlivé kroky algoritmu jsou popsány v následujících bodech:

1. Procesor P_0 vytvoří základní parametry Θ_g z označkovaných dokumentů D_l a rozšíří je do ostatních procesorů.

2. Každý procesor P_r přečte trénovací dokumenty založené na vlastní zodpovědnosti z disku.

3. Jednotlivé iterace:

   (a) E-kroky: Každý procesor P_r odhadne třídu dokumentu používající aktuální globální parametr Θ_g.

   (b) M-kroky: Každý procesor P_r znovu odhaduje třídu, ale pomocí vlastních parametrů Θ_l z daných odhadů třídy každého dokumentu.

   (c) Shrnou se lokální parametry Θ_l k získání nových globálních parametrů Θ_g a vrátí je zpět procesorům.

Naznačení průběhu jednotlivých iterací je naznačeno na obrázku 2 [9].

3.3 Možnosti použití na klasifikaci sentimentu

Použití paralelního zpracování textu v návaznosti na úlohu klasifikace sentimentu je možno využít v předzpracování vstupních dat, případně při tvorbě korpusu. Využití paralelizace je u předzpracování dat lépe využitelný, než na tvorbu korpusu, protože předzpracování probíhá na pozadí ze vstupních neparametrizovaných dat, přičemž tvorba korpusu probíhá v postupných krocích určování kategorií dokumentu a tím není využita hlavní výhoda paralelnosti ve zrychlení zpracování.

4 Korpus

Korpus je soubor počítačově uložených textů, který slouží k jazykovému výzkumu [4]. Tyto texty jsou opatřeny metajazykovými značkami, které vypovídají o samotném textu a také o zařazení jednotlivých slov do různých kategorií (např. kategorie frekvence slov v korpusu, kategorie slovních druhů). K práci s korpusy slouží také i speciální vyhledávací programy. S jejich pomocí je možné vyhledávat slova a slovní spojení v kontextu a zjistit jejich frekvenci v korpusu i původní textový zdroj. Pro formátování textů a vkládání značek se obvykle používá standardizovaný jazyk XML.

4.1 Analýza sentimentu a korpus

Pro analýzu sentimentu bylo potřeba nalézt korpus označený třídami sentimentu, podle kterého by se dalo klasifikovat. Protože analýza sentimentu v České republice není ještě dostatečně prozkoumaná, nebylo možné žádný korpus použitelný pro účely analýzy sentimentu nalézt. Vzhledem k tomu, že si budeme muset korpus vytvořit sami, bylo nutné prostudovat literaturu zabývající se problematikou tvorby korpusů v oblasti analýzy sentimentu.

Byl nalezen dokument [4], kde byla popsána tvorba korpusu z reakcí na články na sociální síti Twitter. V dokumentu se popisuje tvorba anglického korpusu, nicméně použité metody se dají aplikovat i na další jazyky včetně češtiny.

Hlavní důvody pro vybrání Twitteru, jako zdroj zpráv pro tvorbu korpusu byly následující:

- obsahuje velké množství krátkých zpráv vytvořených jednotlivými uživateli této sociální sítě,
- obsah těchto zpráv většinou vyjadřuje osobní názor k nějaké zprávě,
- odpovědi píší různé osoby, není to tedy jednotvárný názor, ale vždy se objeví reakce od různých lidí,
- na Twitteru se může diskutovat o různých tématech (produkty, společnosti, celebrity, politici a další témata) a tím se mohou vyjádřit různé osoby a nezmenšuje se tím okruh diskutujících,

- k daným tématům se mohou vyjádřit i lidé z dalších cizích zemí.

Zprávy v tomto korpusu jsou děleny do následujících kategorií:

- texty, které obsahují kladné emoce/vyjádření (např. štěstí, zábava, radost, kladné zkušenosti),
- texty, které obsahují negativní emoce/vyjádření (smutek, hněv, zklamání, špatné zkušenosti),
- objektivní (neutrální) texty, které obsahují fakt, nebo nevyjadřují žádné emoce.

5 Data ČTK

K této diplomové práci byla poskytnuta data z České tiskové kanceláře (ČTK). Tato data byla vygenerována z databáze InfoBanka ČTK, která obsahuje články psané v českých médiích. Články jsou děleny do různých kategorií (zpravodajství, sport, počasí, ekonomika, celebrity a mnoho dalších). Databáze obsahuje vlastní tvorbu ČTK, ale také články z ostatních českých novinových medií (např. MF Dnes, Právo, Hospodářské noviny, Lidové noviny a další).

Data byla poskytnuta ve formě novinových článků psaných o konkrétní osobě. Tato data byla automaticky vygenerována z databáze ČTK a poté byly vybrány články s danou relevancí o dané osobě. Bylo poskytnuto řádově 200-300 článků o jedné osobě a dohromady bylo k dispozici 10 osob, např. Dagmar Havlová, Michael Kocáb, Martin Roman, Andrej Babiš, Iveta Bartošová a další).

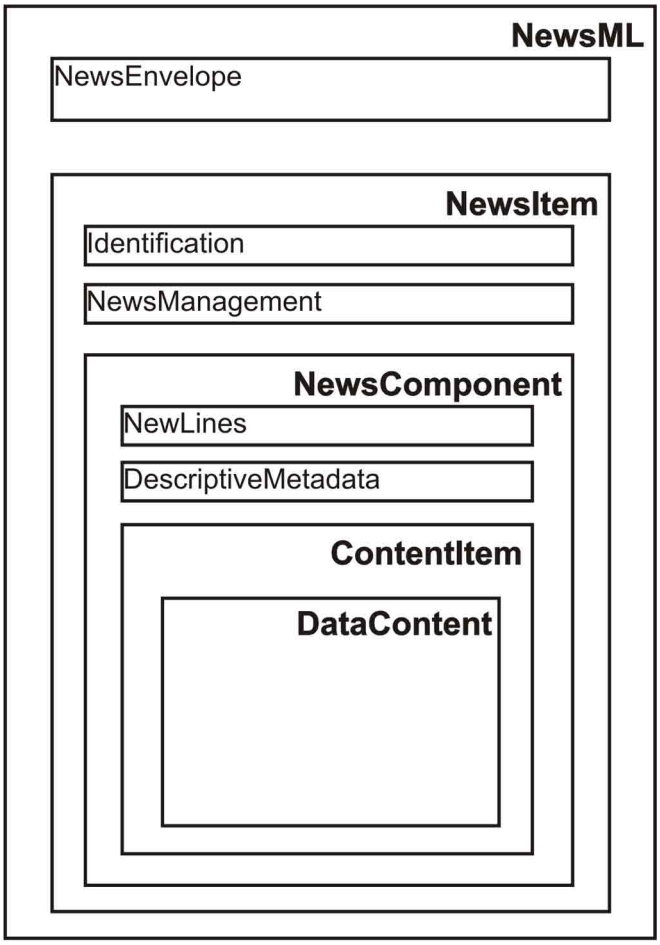
Články vygenerované z databáze o jedné osobě jsou uloženy do jednoho souboru ve formátu NewsML ČTK. Od května 2007 ČTK nabízí své články v tomto novém formátu, který má hlavní přednosti v ulehčení zprostředkovat odběratelům agenturních servisů všechny možnosti, které přineslo zavedení nového redakčního systému.

Tento formát je založen na standardu XML a hlavní výhodou je přímé využití na webových stránkách. Dokumenty v tomto formátu obsahují širokou sadu metadat (údaje popisující vlastní dokument) a tím může uživatel dostat rychlejší přehled o zaměření daného dokumentu. Všechna metadata uvedená v dokumentu jsou vybírána ze slovníku, kde jsou uvedeny množiny jednotlivých hodnot.

5.1 Základní struktura formátu

Pro textovou zprávu dokumentu je vždy definován jeden element **NewsItem**, ve kterém je jeden element **NewsComponent**. Tento element obsahuje další elementy **NewsLines**, **DescriptiveMetadata** a **ContentItem**. Element ContentItem pak v elementu **DataContent** obsahuje vlastní textovou zprávu. Schéma je ukázáno na obrázku 3.

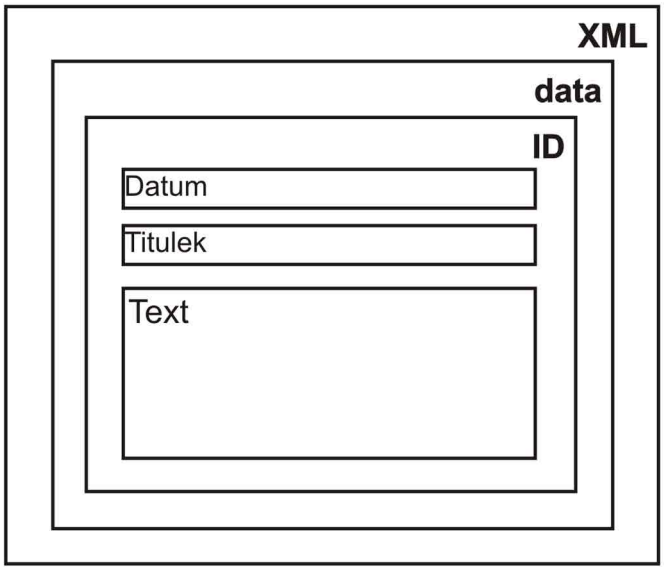
Textová zpráva může obsahovat XHTML značky pro zjednodušení značkování textů. ČTK využívá značky a atributy uvedené v tabulce 1. Všechny značky jsou používány jako párové.



Obrázek 3: Základní struktura textové zprávy formátu NewsML ČTK.

Atribut	**Význam atributu**
<p>	označení odstavce
<a href=””>	hypertextový odkaz s atributem href
<table>	označení tabulky
<tr>	řádek tabulky
<td>	buňka tabulky
<tbody>	alternativní označení dat v tabulce

Tabulka 1: Přehled XHTML značek a atributů a jejich významu.



Obrázek 4: Formát článku ČTK pro analýzu sentimentu.

5.2 Data pro analýzu sentimentu

Původně jsme plánovali na analýzu sentimentu použít korpus z Karlovy Univerzity [2]. Tento korpus nám bohužel nebyl poskytnut, proto bylo potřeba vytvořit korpus vlastní. K tvorbě budou použity data z ČTK.

Články pro analýzu sentimentu byly poskytnuty ve formě jednoho souboru pro danou osobu ve formátu XML ve struktuře podle obrázku 4. Každý článek o konkrétní osobě měl své jednoznačné ID, které se skládalo z média, které článek vydalo (např. T=ČTK, M=MF Dnes, PR=Právo a další) a z data vydání. Dalšími elementy jsou konkrétní datum vydání ve formátu RR-MM-DD, titulek článku a samostatný text článku, který může obsahovat XHTML značky pro značkování textu.

Anotace bude popsána dále.

[2] http://ufal.mff.cuni.cz/

6 Klasifikační nástroje

Po provedení analýzy bylo potřeba nalézt nástroje pro automatickou klasifikaci dokumentů dle sentimentu. Na základě použité studie literatury byly vybrány 3 metody - NB, MaxEnt a SVM. Proto bylo vhodné nalézt takové nástroje, které implementují všechny tři metody zároveň.

6.1 Požadavky na nástroje

Na výběr použitého nástroje byly definovány následující požadavky:

- licence použitelná pro komerční účely ČTK, tzn. nejlépe volně šiřitelné nástroje,
- nástroje programované v jazyce JAVA, pro případné upravování zdrojového kódu,
- nástroj musí podporovat metody klasifikace z analýzy (NB, MaxEnt, SVM),
- dobrá dokumentace, která pomůže k dobrému a rychlejšímu seznámení s nástrojem,
- nástroj bude mít živé projekty, které půjdou ihned zprovoznit.

6.2 Nástroje pro klasifikaci

Při hledání vhodného nástroje byly nalezeny následující systémy, kde je implementována pouze jedna klasifikační metoda. Tyto systémy zřejmě nebudou v této práci použity, nicméně je považují za vhodné je zde uvést pro případ potřeby pouze jednoho klasifikačního algoritmu.

Naivní Bayesův klasifikátor:

- **jBNC**
  - openSource
  - JAVA
  - JavaDoc
  - ''`http://jbnc.sourceforge.net/`''

Maximální Entropie:

- **JTextPro**
  - openSource
  - JAVA
  - ''http://jtextpro.sourceforge.net/''
- **OpenNLP**
  - openSource
  - JAVA
  - vlastní dokumentace
  - ''http://opennlp.apache.org/''

Support Vector Machine:

- **LIBSVM**
  - openSource
  - JAVA, GUI
  - vlastní dokumentace
  - metoda křížové validace
  - ''http://www.csie.ntu.edu.tw/~cjlin/libsvm/''

Nástroje uvedené dále nabízí více klasifikačních algoritmů.

6.3 Mallet

- Naivní Bayesův klasifikátor, Maximální Entropie
- Open source SW, Common Public License 1.0
- JAVA
- Vlastní tutoriál a dokumentace

- Homepage: ''http://mallet.cs.umass.edu/''
- Multiplatformní SW

6.4 OpenSV

- Naivní Bayesův klasifikátor, Support Vector Machine
- Open source BSD licence
- Multiplatformní SW
- JAVA, C, C++
- Vlastní tutoriál a dokumentace
- Homepage: ''http://opencv.willowgarage.com/wiki/''

6.5 Minorthird

- Naivní Bayesův klasifikátor, Support Vector Machine, Maximální Entropie
- Open source BSD licence
- Multiplatformní SW
- JAVA
- Vlastní tutoriál a dokumentace
- Homepage: ''http://sourceforge.net/projects/minorthird/''

Pro vlastní klasifikaci dokumentů byl zvolen nástroj **Minorthird**, který splňoval všechny požadavky, které byly na nástroj požadovány. Nástroj Minorthird navíc obsahuje vlastní grafické uživatelské rozhraní (GUI) a tím bylo seznámení s nástrojem a potom i klasifikace a prohlížení výsledků klasifikace pohodlnější.

6.6 Výsledek klasifikace

Výsledkem klasifikace jsou udávány pomocí dvou veliči: úspěšnost a chybovost klasifikace.

Úspěšnost (ACC – accuracy) [2] klasifikace je určena poměrem počtu úspěšně klasifikovaných dokumentů k celkovému počtu všech klasifikovaných dokumentů. Vztah je ukázán na následujícím vzorci:

$$ACC = \frac{pocet\_uspesne\_klasifikovanych\_dokumentu}{pocet\_vsech\_dokumentu}[\%] \tag{10}$$

Chybovost (ERR – error rate) [2] klasifikace je určena poměrem počtu neúspěšně klasifikovaných dokumentů k celkovému počtu všech klasifikovaných dokumentů. Vztah je ukázán na následujícím vzorci:

$$ERR = \frac{pocet\_chybne\_klasifikovanych\_dokumentu}{pocet\_vsech\_dokumentu}[\%] \tag{11}$$

6.7 Metoda křížové validace

Metoda křížové validace [3] je statistická metoda, kterou dokážeme vyhodnotit úspěšnost klasifikace. U této metody se data rozdělí v zadaném poměru a tím se rozdělí data na množinu dat pro trénování klasifikátoru a na druhou množinu pro testování klasifikátoru. Celý tento postup se opakuje, dokud se nevystřídají všechny kombinace množin zadaného poměru.

Jako příklad uvedu použití křížové validace v poměru 1:5. V tomto poměru se rozdělí soubor dat na pět stejně velkých částí a do množiny trénování se přidělí 4/5 dat a do množiny testování se přidělí 1/5 dat. Tento postup se provede celkem 5x a tím se vystřídají všechny kombinace množin v poměru 1:5. Na závěr celé metody se potom vypočítají výsledky.

Použití metody křížové validace umožňuje i zvolený klasifikační nástroj Minorthird.

[3] Anglicky Cross Validation

7 Tvorba korpusu

Poznatky uvedené v kapitole 4.1, byly využity na tvorbu korpusu z databáze článků ČTK. Byly zvoleny také tři kategorie (pozitivní, negativní a neutrální). Odstavce z jednotlivých článků byly klasifikovány do každé kategorie. Články od ČTK byly poskytnuty, jako souhrn článků o jedné dané osobě. Více v kapitole 7.1. U každé kategorie bylo potřeba získat dostatečné množství dokumentů, aby bylo možné vytvořit statistické modely pro klasifikaci sentimentu.

7.1 Vstupní data pro analýzu a předzpracování dat

Pro automatickou klasifikaci dokumentů je potřeba předzpracovat dokumenty do správné podoby vstupních dat.

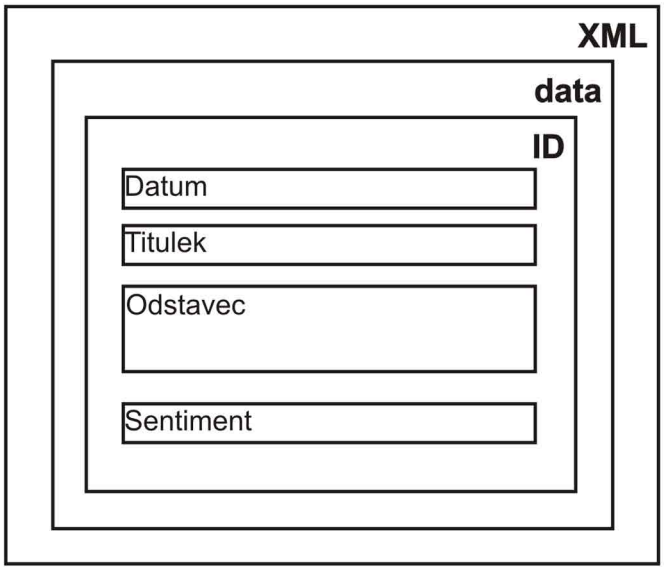
Dokumenty z ČTK jsou v podobě jednoho souboru článků o jedné dané osobě v XML souboru (více v kapitole 5.2). Tento soubor článků bylo potřeba roztřídit do dílčích menších souborů. Na základě prostudované literatury z oblasti analýzy sentimentu jsme rozhodli, že se bude klasifikovat podle odstavců. Každý článek bylo potřeba rozdělit na odstavce a potom uložit do jednoho souboru. V následujícím kroku potom určit, do jaké kategorie vybraný odstavec patří.

Pro tento případ byl vytvořen program, ve kterém se vybere vstupní soubor článků ve formátu XML (viz obrázek 4). Program předzpracuje soubor článků a rozdělí každý článek na odstavce. Jednotlivé odstavce se potom zobrazují uživateli ve vymezeném poli grafického rozhraní programu a uživatel může poté rozhodnout, do jaké kategorie zobrazený odstavec zapadá.

Program využívá již známého formátu XML souboru. Tento formát je zachováván a texty článků se postupně roztřiďují na odstavce. V elementech XML jsou také základní metadata o článku. Program tyto metadata uchovává i v databázi samostatných odstavců ve formátu uvedeném na obrázku 5.

Každý odstavec si zachovává metadata:

- ID článku, kterého byl součástí,
- datum článku, kdy byl vytvořen,
- titulek článku,



Obrázek 5: Formát uložení odstavce v databázi.

- odstavec,
- sentiment (element, který byl do XML souboru přidán: určení třídy sentimentu daného odstavce).

Program využívá struktury dat v článcích ČTK a odděluje jednotlivé odstavce ze samostatných článků pomocí XHTML značek v textu článku (kapitola 5.2). V textech článků se objevuje značka <***p***>, která označuje počáteční značku odstavce a konec odstavce označuje konečná značka <***/p***>. Tyto značky se v tomto formátu nezobrazují, protože by redakční systém ČTK mohl značky chápat jako součást jazyka HTML a značku odstavce nerozpoznat. Z tohoto důvodu se používají pro výpis těchto znaků znakové entity. V textech článků se používají k oddělení odstavců značky:

- **<p>** - jako počáteční značka odstavce
- **</p>** - jako konečná značka odstavce

Program na předzpracování dat má dva typy výstupních souborů, kde je obsažen odstavec. Formáty výstupních souborů jsou:

- XML soubor s metadaty článku, který je dělen na odstavce
  „jméno-osoby“_„třída-sentimentu“_„číslo-odstavce“.xml
- textový soubor pouze se samostatným textem odstavce
  „jméno-osoby“_„třída-sentimentu“_„číslo-odstavce“.txt

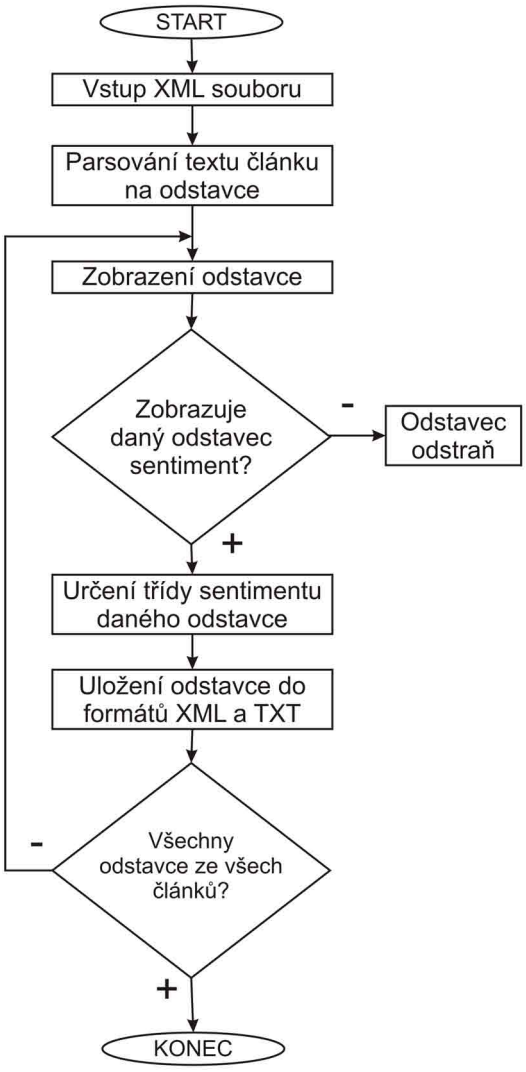
Program na předzpracování dat ze souborů z databáze ČTK pracuje podle blokového schématu, které je na obrázku 6.

7.2 Program na předzpracování dat

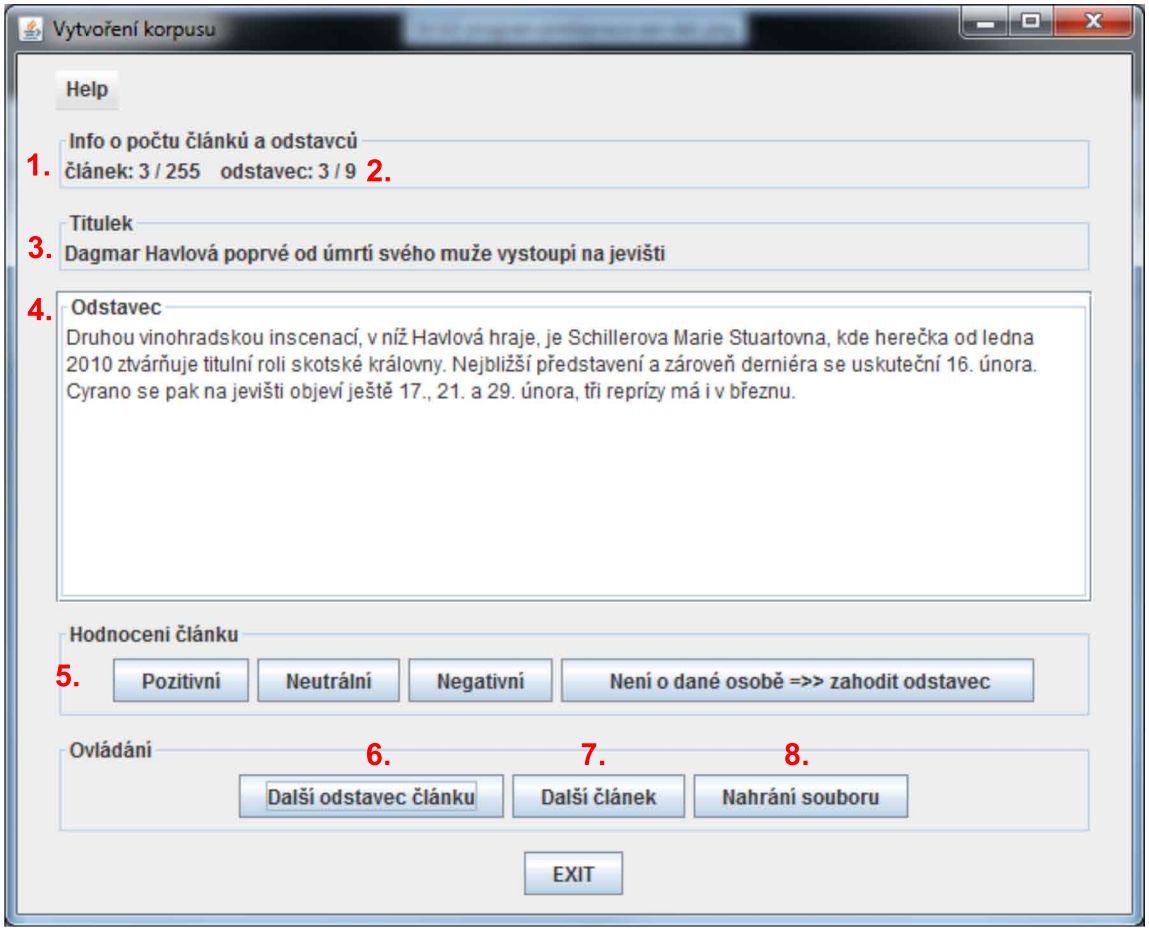
Na obrázku 7 vidíme obrazovku grafického uživatelského rozhraní programu pro předzpracování článků pro analýzu sentimentu s nahraným odstavcem o Dagmar Havlové a s podrobnějším vysvětlením níže.

Význam očíslovaných komponent:

- 1. Zobrazení pořadového čísla článku z nahraného souboru článků o dané osobě.
- 2. Zobrazení pořadového čísla zobrazeného odstavce v článku.
- 3. Zobrazení titulku vybraného článku.
- 4. Zobrazení odstavec, který se anotuje.
- 5. Tlačítka pro určení sentimentu daného odstavce. Více o určování sentimentu v kapitole 7.
- 6. Tlačítko pro zobrazení dalšího odstavce v článku.
- 7. Tlačítko pro zobrazení začátku dalšího článku.
- 8. Tlačítko pro vybrání souboru s články, které se budou anotovat, zobrazení dialogového okna pro vybrání souboru, v tomto okně je zapotřebí vybrat soubor formátu XML.



Obrázek 6: Blokové schéma předzpracovnání dat z ČTK.



Obrázek 7: Obrazovka programu pro předzpracování dat.

7.3 Problémy související s předzpracováním

Bylo zjištěno, že ne každý článek je tvořen značkami, které obsahují v textu značky pro začátek a konec odstavce. Tyto značky obsahují pouze články, které vytvořila ČTK. Články od ostatních médií v databázi ČTK tyto značky neobsahují, a proto nebylo možné automaticky oddělit odstavce v jednotlivých článcích. Jiné články než články ČTK proto nebyly použity pro tvorbu korpusu.

7.4 Parametrizace korpusu

Každý anotovaný článek z vytvořeného korpusu je potřeba parametrizovat pro klasifikaci. Pro parametrizaci korpusu pro rozpoznávání sentimentu byla použita základní metoda a to četnost výskytu slov v daném dokumentu.

Formát výstupní parametrizace je:

k název-souboru třída parametrizovaná-slova

- parametr **k** je použit pro kompatibilitu se staršími verzemi Minorthirdu,
- **název-souboru** je název vstupního souboru v korpusu, pro určení anotovaného dokumentu,
- **třída** je třída určeného sentimentu, do jaké byl daný dokument zařazen,
- **parametrizovaná-slova** je pole všech slov v dokumentu, ve formátu slovo=četnost, oddělené mezerou.

Všechny parametry parametrizace pro jeden daný dokument musí být v jedné řádce a vždy oddělené mezerou.

Dále je uveden příklad dokumentu, který byl zařazen do negativní třídy a jeho název byl **michael_kocab_neg_0064**:

k michael_kocab_neg_0064 neg Začátkem=1 letošního=1 března=1 rozvířil=1 deník=1 Blesk=1 Kocábovy=1 problémy=1 když=1 uvedl=1 že=1 ministr=1 má=1 poměr=1 se=1 svou=2 mluvčí=1 Lejlou=1 Abbásovou=1 a=1 údajně=1 začal=1 řešit=1 svůj=1 rozvod=1 Kocáb=1 kvůli=1 zveřejnění=1 informací=1 ze=1 soukromého=1 života=1 nabídl=1 premiéru=1 Fischerovi=1 rezignaci=1 kterou=1 premiér=1 nepřijal=1

Byl vytvořen skript, který každý dokument z korpusu parametrizuje zmíněnou metodou a jako výstup ze skriptu je potom soubor ve formátu, který umožní ke klasifikaci jako vstup do klasifikátoru Minorthird (viz. Kapitola 8.1). Skript lze spustit z příkazové řádky a má dva vstupní parametry oddělené mezerou.

Prvním parametrem je adresář, ve kterém jsou soubory s anotovaným odstavcem, nebo lze také zadat pouze jeden soubor, ve kterém je anotovaný odstavec. Druhým parametrem je kategorie sentimentu, do které byl, nebo byly anotovány odstavce.

Výstupem programu je jeden soubor ve formátu $< vstupni\_soubor > - < kategorie >$ *.txt*, ve kterém jsou všechny soubory z adresáře (nebo jeden zadaný soubor) parametrizovány a tento soubor poté se může dále zpracovávat jako vstup pro klasifikaci nástroje Minorthird.

8 Klasifikace dokumentů

Tato kapitola se zabývá hlavním cílem diplomové práce a to klasifikací sentimentu jednotlivých dokumentů. Cílem klasifikace je vždy určit do jaké třídy sentimentu patří testovaný dokument. V první části je popsána klasifikace pomocí nástroje Minorthird a v další části se zabýváme paralelní kombinací metod.

V analýze byly zvoleny 3 třídy (pozitivní, negativní, neutrální), do kterých by měl vždy testovaný dokument být zařazen. Výsledkem klasifikace jsou tři pravděpodobnosti, které procentuálně určují zařazení daného dokumentu do jednotlivých tříd.

8.1 Klasifikace s Minorthird

Ke klasifikaci dokumentů byl použit nástroj Minorthird, kde jsou integrovány všechny 3 metody: NB, SVM a MaxEnt. Metody jsou implementované jako samostatné třídy.

Minorthird je možné spouštět dvojím způsobem a to z příkazové řádky nebo pomocí vlastního GUI. Na začátek na seznámení s nástrojem a vyzkoušení jeho možností je vhodné využívat GUI. Pro pozdější samostatné trénování/testování klasifikátoru je lepší využívat spouštění nástroje z příkazové řádky s dalšími parametry spuštění. V tabulce 2 jsou ukázány parametry pro spuštění nástroje Minorthird.

Vstupní parametr	**význam parametru při spuštění Minorthird**
-saveAs $< vystupni\_soubor >$	uložení výsledku klasifikace
-data $< vstupni\_soubor >$	vstupní data pro trénování klasifikátoru
-learner $< LEARNER >$	určení dané metody klasifikace. (Více v tabulce 4)
-splitter $< SPLITTER >$	spuštění experimentu pro trénování a testování klasifikátoru zároveň ze vstupních dat. (Více v tabulce 3)
-help	vypsání nápovědy
-gui	zobrazí se GUI Minorthirdu
-showResults	zobrazí výsledek experimentu v GUI
-showLabels	zobrazení trénovacích dat
-showTestDetails	zobrazení skutečného průběhu klasifikace

Tabulka 2: Spouštěcí parametry Minorthird z příkazové řádky.

SPLITTER:	
k5	pěti-násobná metoda křížové validace
k10	deseti-násobná metoda křížové validace
s10	vrstvená desetinásobná křížová validace

Tabulka 3: Přehled parametrů SPLITTER Minorthird, která je použita pro nastavení křížové validace.

LEARNER:	
"new OneVsAllLearner(\"new NaiveBayes()\")"	spuštění NB klasifikátoru
"new OneVsAllLearner(\"new MaxEntLearner()\")"	spuštění MaxEnt klasifikátoru
"new OneVsAllLearner(\ "new SVMLearner()\")"	spuštění SVM klasifikátoru
Použití *OneVsAllLearner* je z důvodu, aby se používal více-třídový klasifikátor	

Tabulka 4: Přehled parametrů LEARNER Minorthird, které jsou použity pro volbu klasifikátoru.

8.2 Návrh systému pro klasifikaci sentimentu

Celý systém je navržen tak, že na začátku se musí parametrizovat vstupní data. Po parametrizaci se nahrají data do nástroje Minorthird. Tento nástroj obsluhuje jedna třída, která vytvoří 3 vlákna, které představují klasifikátory (NB, MaxEnt, SVM). Paralelní zpracování probíhá stejně u každého klasifikátoru a to tak, že nejdříve probíhá trénování a poté testování klasifikátoru. Výsledkem klasifikace jsou soubory, ve kterých jsou vyhodnoceny výsledky klasifikace pro každý klasifikátor. Z těchto důvodů bylo potřeba upravit Minorthird. Blokové schéma systému je uvedeno na obrázku 8. Druhou součástí systému je hledání optimálních parametrů pro paralelní klasifikaci. Blokové schéma je uvedeno na obrázku 9. Více informací v následujících kapitolách.

8.3 Úprava nástroje Minorthird pro účely DP

Klasifikační nástroj bylo třeba upravit pro vlastní klasifikaci dokumentů. Z předchozích kapitol byla zvolena možnost spuštění z příkazové řádky. Celý proces klasifikace se skládá ze dvou hlavních kroků. První část programu se zabývá paralelní klasifikací dokumentů s nastavenými hodnotami vah. Druhou částí programu je hledání optimálních parametrů pro kombinaci klasifikátorů. Paralelní klasifikace je realizována pomocí tří klasifikátorů: NB, SVM a MaxEnt. Dílčí výsledky klasifikace jsou pak složeny do

jednoho hlavního výsledku.

První úprava spočívá ve změně formátu výstupních souborů. V původním formátu má Minorthird výstupní soubory komprimovány metodou ***GZIPOutputStream*** v třídě ***Evaluation*** balíku ***classify/experiments***. V této třídě byly provedeny i následující úpravy Minorthirdu.

Dále bylo zapotřebí upravit výstupní soubor statistik klasifikace – přidání konfuzní matice[4] v procentech a ne jen v počtech daných dokumentů zařazených do dané třídy a přidání času klasifikace všech dokumentů daným klasifikátorem. Soubor výstupních statistik je vždy ve formátu:

$$Results- < klasifikator > - < soubor >$$

kde $< klasifikator >$ je název klasifikátoru (NB, SVM, MaxEnt) a $< soubor >$ je název souboru, kam se mají uložit výsledky (zadaný v příkazové řádce při spuštění).

Další úprava se týká výstupního souboru, kde jsou uloženy výsledky klasifikace. Byla provedena z důvodu určení jednotlivých dokumentů po klasifikaci a jím vypočítané pravděpodobnosti přiřazení do dané třídy. Soubor obsahuje v prvním řádku třídy, do kterých se klasifikovalo, v dalším řádku záhlaví, ve kterém jsou uvedeny zkrácené názvy sloupců (viz tabulka 6. V dalších řádkách jsou uvedeny hodnoty klasifikace pro každý klasifikovaný dokument. Soubor výsledků klasifikace je vždy ve formátu:

$$< klasifikator > - < soubor >$$

Ukázka části (prvních pět řádků) obsahu výstupního souboru s výsledky klasifikace je uvedena v tabulce 5.

8.3.1 Vstupní parametry

Dále byla upravena část se vstupními parametry nástroje. Program je vytvořený tak, že se celý ovládá z příkazové řádky s parametry v tabulce 7.

Pro účely diplomové práce byla odebrána možnost volby jiného klasifikátoru než NB, SVM a MaxEnt. Parametr ***-data*** pro zvolení vstupních dat a parametr ***-splitter*** pro

[4]**Konfuzní matice** je matice úspěšnosti zařazení klasifikovaných dokumentů do jednotlivých tříd. Úspěšnost klasifikace je vyjádřena na diagonále matice, na ostatních místech matice je chybovost klasifikace dané třídy. Rozměr matice je dán počtem tříd, do kterých se dokumenty zařazují – v našem případě tedy 3x3).

[neg,neut,poz]
name class negLabel negWeight posLabel posWeight neutLabel neutWeight bestLabel bestWeight
michael_kocab_poz_1012 poz neg 0.41537340477643836 poz 0.4776882945414192 neut 0.10693830068214244 poz 0.4776882945414192
zdenek_bakala_neut_0175 neut neg 0.12252904654081348 poz 0.11048876329214939 neut 0.7669821901670372 neut 0.7669821901670372
ladislav_batora_neg_0215 neg neg 0.4989284538768949 poz 0.17772968691526558 neut 0.32334185920783975 neg 0.4989284538768949

Tabulka 5: Příklad obsahu výstupního souboru s výsledky klasifikace.

Název sloupce	**Význam hodnoty**
name	Název klasifikovaného dokumentu.
class	Předpokládaná třída, kam by měl být dokument zařazen.
negLabel	Název negativní třídy.
negWeight	Pravděpodobnost zařazení do negativní třídy.
posLabel	Název pozitivní třídy.
posWeight	Pravděpodobnost zařazení do pozitivní třídy.
neutLabel	Název neutrální třídy.
neutWeight	Pravděpodobnost zařazení do neutrální třídy.
bestLabel	Název nejlépe klasifikované třídy, kam klasifikátor zařadil dokument.
bestWeight	Pravděpodobnost, se kterou klasifikátor zařadil dokument do nejlépe klasifikované třídy.

Tabulka 6: Přehled a význam sloupců ve výstupním souboru výsledku klasifikace.

Vstupní parametr	**Význam parametru**
-help	Výpis nápovědy programu.
-saveAs $< soubor >$	Uložení výsledků jednotlivých klasifikátorů do $< souboru >$.
-paralelClasf	Spuštění stanovení parametrů pro paralelní klasifikaci dokumentů.
-testParamVahy	Spuštění testování optimálních parametrů vah.
-vahaNB $< vaha >$	Zadání váhy pro NB klasifikátor.
-vahaSVM $< vaha >$	Zadání váhy pro SVM klasifikátor.
-vahaMaxEnt $< vaha >$	Zadání váhy pro MaxEnt klasifikátor.

Tabulka 7: Přehled a význam vstupních parametrů klasifikační aplikace.

spuštění metody křížové validace a další původní parametry Minorthirdu byly použity pro výběr vstupních dat a pro použití křížové validace.

8.3.2 Paralelní klasifikace

Jedním z úkolů diplomové práce bylo zjistit, jestli paralelní zpracování metod zjištěných v analýze sníží dobu klasifikace dokumentů a také jestli paralelní klasifikací se dosáhne lepších výsledků, než kdyby se používaly metody samostatně. Na tento úkol byl upraven i nástroj Minorthird, ve kterém byla vytvořena tři vlákna klasifikace, které obsluhuje jedna třída (ClassifyDP). Každému vláknu byla přiřazena jedna metoda (NB, SVM a MaxEnt) a byly předány globální parametry, které se zadávají při spuštění celé aplikace. V každém klasifikátoru probíhá trénování klasifikátoru ze vstupních dat. Při použití metody křížové validace probíhá i testování (viz. Kapitola 6.7). Blokové schéma aplikace je zobrazeno na obrázku 8. Tento navržený paralelismus je vlastní.

Po ukončení činnosti paralelního klasifikátoru (vlákna) je vypočten výsledek celé klasifikace.

Lineární kombinace klasifikátorů

Výsledná pravděpodobnost je dána podle vzorce:

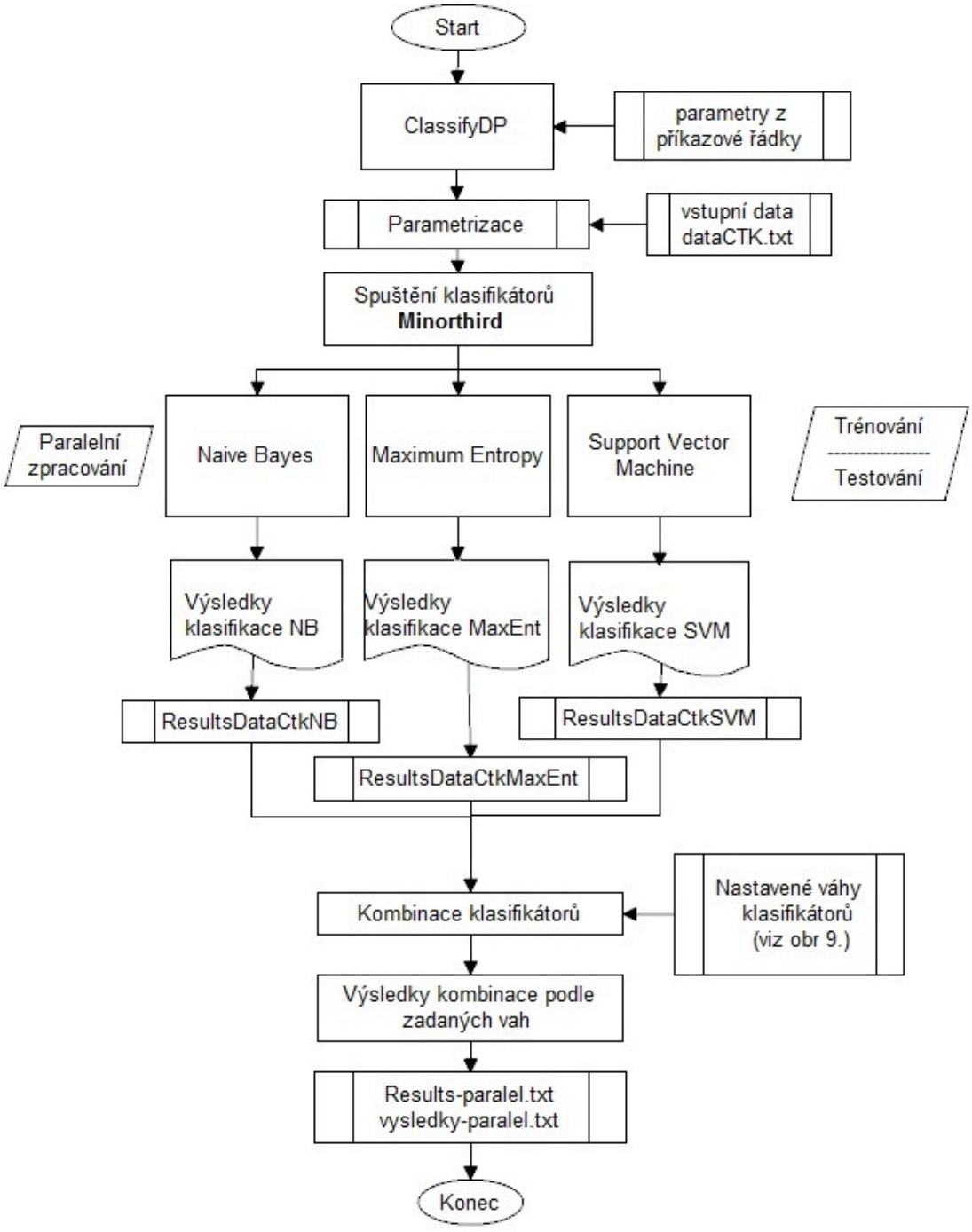
$$P(C|F) = w_{NB} * P_{NB} + w_{SVM} * P_{SVM} + w_{MaxEnt} * P_{MaxEnt} \tag{12}$$

kde w_{NB} je váha klasifikátoru NB, P_{NB} je výsledná pravděpodobnost klasifikátoru NB, w_{SVM} je váha klasifikátoru SVM, P_{SVM} je výsledná pravděpodobnost klasifikátoru SVM, w_{MaxEnt} je váha klasifikátoru MaxEnt a P_{MaxEnt} je výsledná pravděpodobnost klasifikátoru MaxEnt. Dále platí, že $w_{NB} + w_{SVM} + w_{MaxEnt} = 1$. C značí klasifikované třídy a F je množina příznaků, které charakterizují dokument.

Výsledná třída je potom dána jako

$$\hat{C} = \arg\max_{C} P(c|d) \tag{13}$$

Výsledek klasifikace pro každý dokument můžeme nalézt v souboru ***vysledky-paralel.txt*** pod názvem sloupce ***vysledekParalelVahy***.



Obrázek 8: Blokové schéma aplikace klasifikace dokumentů.

Většinové hlasování

V této metodě se porovnávají dosažené výsledky klasifikace u jednotlivých klasifikátorů tak, že výsledná třída je stanovena třídou, kterou určila většina klasifikátorů. V našem případě (3 klasifikátory) mohou nastat případy:

- Všechny 3 klasifikátory zařadí dokument do stejné třídy ⇒ výsledek tato třída.
- Pouze 2 klasifikátory zařadí dokument do stejné třídy ⇒ výsledek třída, která byla určena u 2 klasifikátorů.
- V případě, že byla u každého klasifikátoru určena rozdílná třída, je zvolena třída s nejvyšší výstupní a posteriorní pravděpodobností.

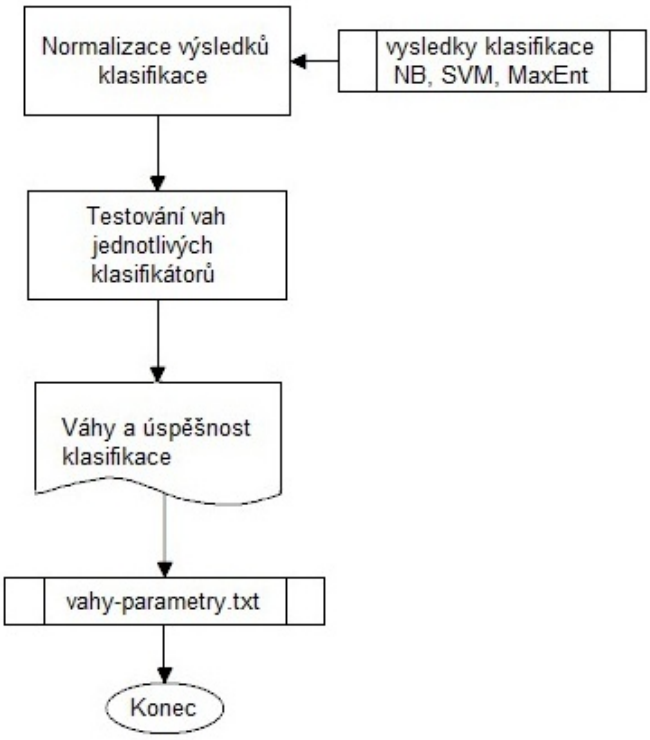
Výsledek klasifikace pro každý dokument můžeme nalézt v souboru ***vysledky-paralel.txt*** pod názvem sloupce ***vysledekPorovnani***.

8.3.3 Hledání optimálních parametrů vah klasifikátorů

Pro paralelní kombinaci klasifikátorů je nutné vhodně nastavit parametry vah klasifikátorů w_{NB}, w_{SVM}, w_{MaxEnt}. To se provede nastavením parametru ***–testParamVahy*** z příkazového řádku při spuštění aplikace. Získáme tak přehled všech možných kombinací vah klasifikátorů a ke každé kombinaci získáme úspěšnost klasifikace.

Optimálním nastavením vah jsou zvoleny hodnoty w_{NB}, w_{SVM}, w_{MaxEnt} tak, že celková přesnost klasifikace na zvolené části korpusu (tzv. vývojová část) je maximální. Postupně jsou prohledávány všechny možné kombinace hodnot
w_{NB}, w_{SVM}, $w_{MaxEnt} \in (0; 1)$ s krokem 0,1

Postup při hledání optimálních nastavení vah je znázorněn blokovým schématem na obrázku 9. Důležitým předpokladem pro hledání vah je, že musí být provedena klasifikace dokumentů a tím známy jednotlivé výsledky klasifikace z každého klasifikátoru, protože vstupními soubory jsou právě výsledky klasifikace. Všechny výsledky klasifikací se ze souborů nahrají a vytvoří se databáze výsledků klasifikací pro každý dokument, tzn., že pro každý dokument máme pohromadě předpoklad, do které třídy měl být dokument zařazen a všechny pravděpodobnosti s výsledky klasifikace jednotlivých klasifikátorů. Tato databáze je tvořena ArrayListem třídy ***DataClassificationAll***.



Obrázek 9: Blokové schéma hledání optimálních parametrů.

Z blokového schématu je vidět, že je provedena normalizace výsledků. Normalizace spočívá v tom, že výstupní pravděpodobnosti u jednotlivých klasifikátorů jsou v různém rozmezí a normalizací dosáhneme toho, že všechny pravděpodobnosti budou v rozmezí 0 až 1. V dalším kroku probíhá testování všech možných kombinací vah klasifikátorů a ukládá se úspěšnost klasifikace.

Výsledkem testování je soubor ***vahy-parametry.txt***, ve kterém je tabulka se všemi kombinacemi vah a vypočtenou úspěšností klasifikace. V prvním řádku a prvním sloupci jsou uvedeny hodnoty vah, pro které byla vypočítávána úspěšnost, třetí hodnota váhy se dopočítává.

9 Dosažené výsledky

Z přečtené literatury a z dostupných experimentů byly zvoleny statistické metody klasifikace dokumentů, Naive Bayes (Naivní Bayesova), Support vector machine (metoda podpůrných vektorů) a Maximum Entropy (Maximální Entropie). Tyto klasifikátory byly vybrány na základě dosahované úspěšnosti klasifikace a také proto, že jsou nejpoužívanější v oblasti analýzy sentimentu. Nejlepších výsledků by měl dosahovat SVM, poté MaxEnt a relativně nejhorších výsledků by měl dosahovat Naive Bayesův klasifikátor.

Z dostupné literatury bylo zvoleno, že se bude klasifikovat do tří tříd: Pozitivní, Negativní a Neutrální. V literatuře se také experimentovalo pouze se dvěma třídami (poz a neg), ale také s pěti třídami. Klasifikovat se budou pouze celé odstavce, nikoli celé články.

9.1 Korpus

K ověření metod byl použit korpus z dat ČTK, jehož tvorba byla popsána v kapitole 7. Struktura korpusu je zobrazena v tabulce 8. Tvorba korpusu zabrala více času, než se předpokládalo a to bylo hlavním důvodem pozdního odevzdání této diplomové práce.

Počet osob, o kterých byly články, vybrané z databáze ČTK	5
Počet zobrazených článků	Zdeněk Bakala – 40 Ladislav Bátora - 51 Dagmar Havlová - 19 Michael Kocáb - 51 Martin Roman - 51
Počet celkem zobrazených článků	212
Počet hodnocených odstavců v článcích	4012
Počet odstavců zařazených do pozitivní kategorie	258
Počet odstavců zařazených do negativní kategorie	173
Počet odstavců zařazených do neutrální kategorie	319
Počet nezařazených odstavců do korpusu	3262
Počet odstavců v korpusu	750

Tabulka 8: Přehled článků a odstavců anotovaných v korpusu.

Pokud není uvedeno jinak, byl pro experimenty použit celý korpus.

9.2 Klasifikace dokumentů

Korpus a vstupní dokumenty bylo potřeba parametrizovat. Na parametrizaci byla využita základní metoda četnosti výskytu slov v daném dokumentu (viz kapitola 7.4.

Pro klasifikaci dokumentů byla použita metoda křížové validace v poměru 1:5. Celá množina dokumentů se tedy rozdělila na 4/5 dokumentů pro trénování klasifikátoru a 1/5 pro testování a tento postup se 5x opakoval. Pro vyhodnocování výsledků klasifikace byly použity veličiny úspěšnost (ACC) a chybovost (ERR), uvedeny v kapitole 6.6.

V následujících kapitolách jsou uvedeny výsledky klasifikace u jednotlivých metod. Ke každému klasifikátoru jsou uvedeny tabulky s výsledky klasifikace, kde je uvedena konfuzní matice, celková úspěšnost a chybovost klasifikace. Dále je zobrazena na obrázku 10 závislost úspěšnosti klasifikace klasifikátoru na počtu použitých dokumentů. Následující tři experimenty s klasifikátory budou provedeny s tímto scénářem.

9.3 Naivní Bayesův klasifikátor

Při experimentech se úspěšnost pohybovala mezi 70 - 72% při klasifikaci 350 i 750 dokumentů.

NB klasf.	**DataCTKLight (14 dok.)**			**DataCTKShort (150 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,75	0	0,25	0,62	0,1	0,28
	0,4	0,6	0	0,3846	0,5	0,1154
	0,6	0,2	0,2	0,375	0,1042	0,5208
ACC	0,5			0,547		
ERR	0,5			0,453		

Tabulka 9: Naive Bayesův klasifikátor - úspěšnost klasifikace pro 14 a 150 dokumentů.

9.4 Maximální Entropie

Při experimentech se úspěšnost pohybovala maximálně do 70% při klasifikaci 350 i 750 dokumentů.

NB klasf.	**DataCTKMedium (350 dok.)**			**DataCTK (750 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,7143	0,1837	0,102	0,6647	0,2197	0,1156
	0,1587	0,7989	0,0423	0,1285	0,7053	0,1661
	0,1928	0,241	0,5663	0,1124	0,1977	0,6899
ACC	0,725			0,691		
ERR	0,275			0,309		

Tabulka 10: Naive Bayesův klasifikátor - úspěšnost klasifikace pro 350 a 750 dokumentů.

MaxEnt klasf.	**DataCTKLight (14 dok.)**			**DataCTKShort (150 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,5	0	0,5	0,38	0,32	0,3
	0	0,6	0,4	0,2115	0,6538	0,1346
	0	0,2	0,8	0,1667	0,2708	0,5625
ACC	0,643			0,534		
ERR	0,357			0,466		

Tabulka 11: Maximální Entropie - úspěšnost klasifikace pro 14 a 150 dokumentů.

MaxEnt klasf.	**DataCTKMedium (350 dok.)**			**DataCTK (750 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,5918	0,3163	0,0918	0,5954	0,2312	0,1734
	0,0847	0,8201	0,0952	0,1097	0,7429	0,1473
	0,1205	0,3253	0,5542	0,1085	0,2519	0,6395
ACC	0,7			0,674		
ERR	0,3			0,326		

Tabulka 12: Maximální Entropie - úspěšnost klasifikace pro 350 a 750 dokumentů.

9.5 SVM

Při experimentech se úspěšnost vždy pohybovala v rozmezí 70-72% při klasifikaci 350 i 750 dokumentů. Lehce klesající přesnost rozpoznávání (rozdíl -2%) při použití 750 dokumentů je statisticky nevýznamný (95% interval spolehlivosti ±2%).

SVM klasf.	**DataCTKLight (14 dok.)**			**DataCTKShort (150 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,25	0	0,75	0,32	0,38	0,3
	0	0,4	0,6	0,1923	0,6731	0,1346
	0	0,8	0,2	0,2292	0,25	0,5208
ACC	0,286			0,507		
ERR	0,714			0,493		

Tabulka 13: SVM - úspěšnost klasifikace pro 14 a 150 dokumentů.

SVM klasf.	**DataCTKMedium (350 dok.)**			**DataCTK (750 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,5918	0,3061	0,102	0,5491	0,2659	0,185
	0,0741	0,8836	0,0423	0,0752	0,7962	0,1285
	0,1205	0,3614	0,5181	0,093	0,2481	0,6589
ACC	0,724			0,692		
ERR	0,276			0,308		

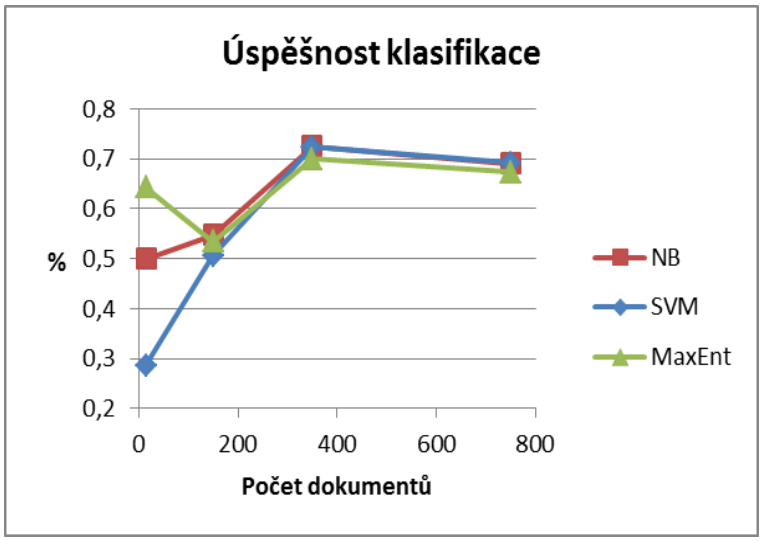
Tabulka 14: SVM - úspěšnost klasifikace pro 350 a 750 dokumentů.

9.6 Paralelní kombinace

Paralelní klasifikace a její vyhodnocování probíhalo dvěma metodami (lineární kombinací klasifikátorů a většinové hlasování), uvedené v kapitole 8.3.2.

9.6.1 Metoda lineární kombinace klasifikátorů

Výsledky paralelní klasifikace metodou lineární kombinace klasifikátorů (kapitola 8.3.2 jsou uvedeny v tabulkách 15 a 16. V tabulce je uvedena konfuzní matice, celková úspěšnost a chybovost klasifikace. Na obrázku 11 je zobrazena závislost úspěšnosti klasifikace na počtu použitých dokumentů.



Obrázek 10: Závislosti úspěšnosti klasifikace klasifikátorů na počtu použitých dokumentů.

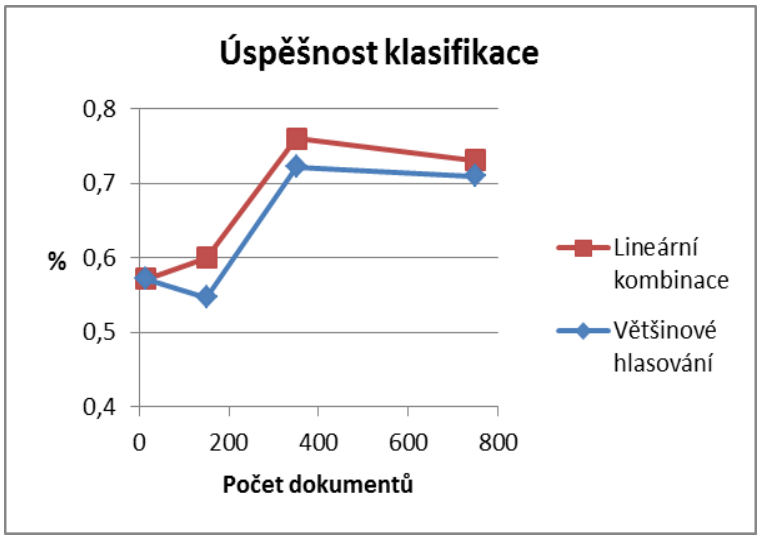
Na obrázku 12 je zobrazen graf úspěšnosti klasifikace při použití dvou klasifikátorů a největší úspěšnosti 72% se dosáhlo u použití NB a MaxEnt klasifikátorů. Na obrázku 13 je zobrazen graf úspěšnosti klasifikace při použití tří klasifikátorů. V tomto grafu je vidět přehled jednotlivých vah klasifikátorů. Úspěšnost nad 70% označuje růžová plocha v grafu a nejlepší úspěšnosti 73% se dosáhlo při experimentu při nastavených váhách $w_{NB} = 0{,}8$, $w_{MaxEnt} = 0{,}1$ a $w_{SVM} = 0{,}1$ (viz tabulky 15 a 16). Nejlepší úspěšnost klasifikace vždy průměrně dosahuje lehce nad 70%, s téměř všemi zadanými váhami (viz obr. 13).

Paralel váhy	**DataCTKLight (14 dok.)**			**DataCTKShort (150 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,1429	0,1429	0,0714	0,2	0,0867	0,0333
	0,0714	0,2143	0	0,08	0,1867	0,0677
	0,1429	0	0,2143	0,0333	0,1	0,2133
ACC	0,5714			0,6		
ERR	0,4286			0,4		

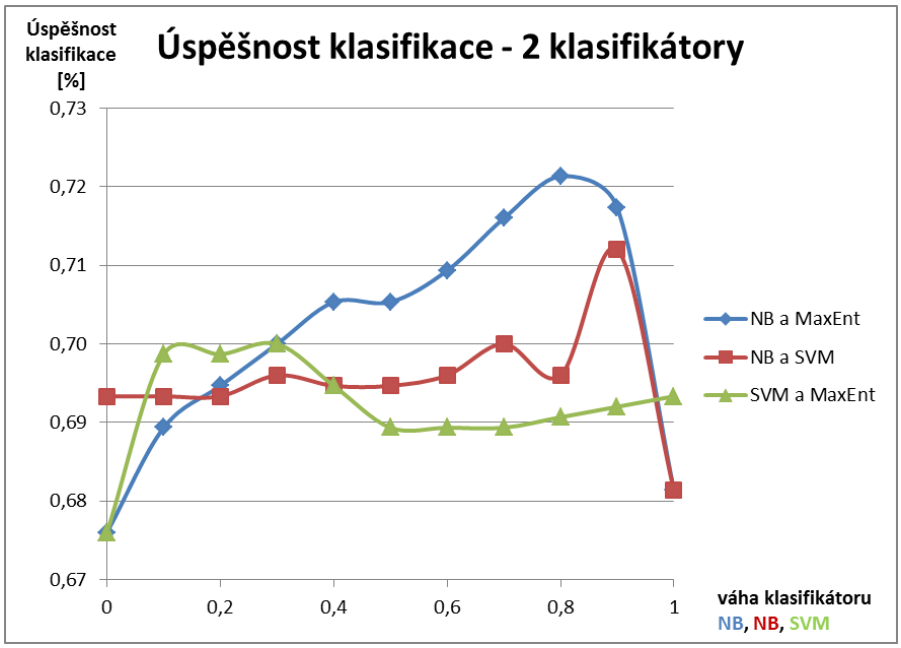
Tabulka 15: Úspěšnost klasifikace metodou lineární kombinace klasifikátorů pro hodnoty vah: $w_{NB} = 0{,}8$, $w_{MaxEnt} = 0{,}1$ a $w_{SVM} = 0{,}1$ pro 14 a 150 dokumentů.

Paralel váhy	**DataCTKMedium (350 dok.)**			**DataCTK (750 dok.)**		
	poz	neg	neut	poz	neg	neut
Konfuzní Matice	0,1216	0,0405	0,0622	0,2373	0,0253	0,0813
	0,0134	0,1865	0,0649	0,024	0,1427	0,064
	0,0162	0,0432	0,4514	0,0493	0,0267	0,3493
ACC	0,7595			0,7308		
ERR	0,2405			0,2692		

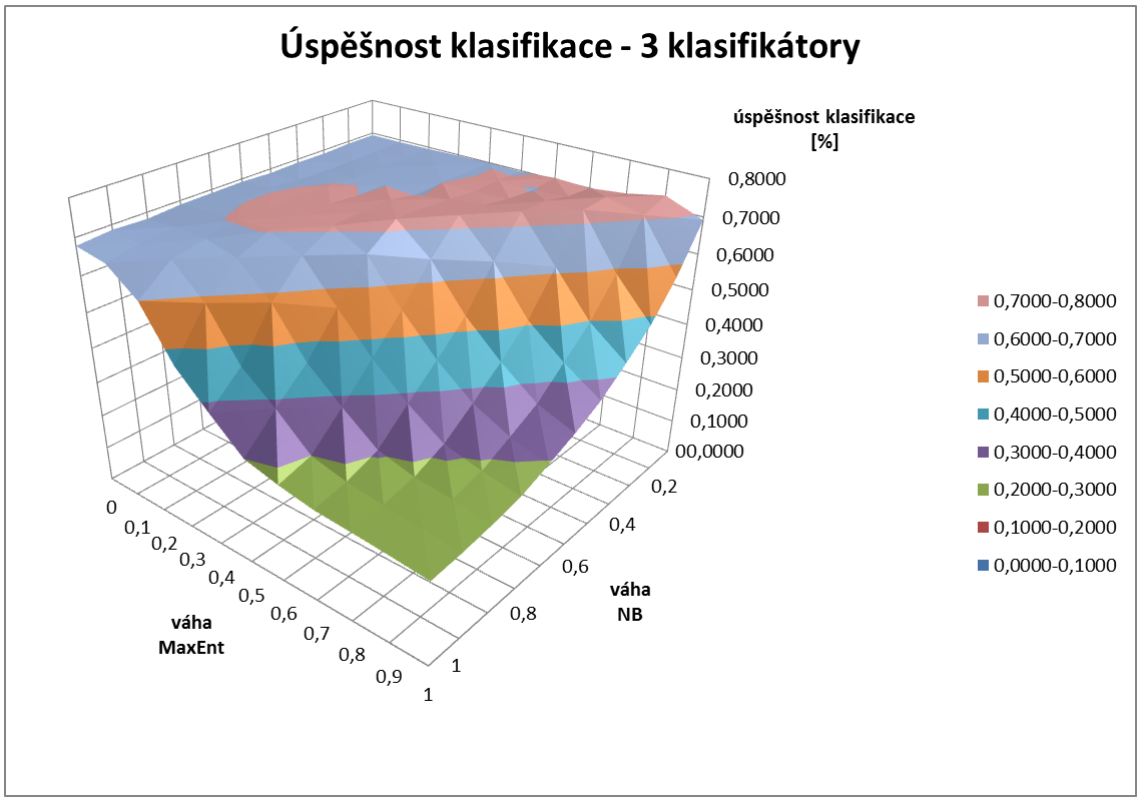
Tabulka 16: Úspěšnost klasifikace metodou lineární kombinace klasifikátorů pro hodnoty vah: $w_{NB} = 0{,}8$, $w_{MaxEnt} = 0{,}1$ a $w_{SVM} = 0{,}1 = 0{,}4$ pro 350 a 750 dokumentů.



Obrázek 11: Závislosti úspěšností paralelních klasifikací na počtu použitých dokumentů.



Obrázek 12: Graf úspěšnosti při použití dvou klasifikátorů.



Obrázek 13: Graf úspěšnosti při použití tří klasifikátorů.

9.6.2 Metoda většinového hlasování

Výsledky paralelní klasifikace metodou většinového hlasování (kapitola 8.3.2) jsou uvedeny v tabulce 17. V tabulce je uvedena úspěšnost klasifikace. Na obrázku 11 je zobrazena závislost úspěšnosti klasifikace na počtu použitých dokumentů.

	DataCTKLight (14 dok.)	**DataCTKShort (150 dok.)**	**DataCTKMedium (350 dok.)**	**DataCTK (750 dok.)**
ACC	0,5714	0,5467	0,7216	0,7093
ERR	0,4286	0,4533	0,2784	0,2907

Tabulka 17: Úspěšnost klasifikace metodou většinového hlasování.

9.6.3 Doby zpracování klasifikace

Doby jednotlivých klasifikací u jednotlivých klasifikátorů s daným počtem vstupních dokumentů jsou uvedeny v tabulkách 18 a 19. Z tabulky je zřejmé, že paralelní klasifikací dobu klasifikace urychlíme, ale úspěšnost klasifikace je vždy podobná a pohybuje se okolo 70% při větší množině vstupních dokumentů. V tabulce 20 je uvedeno info o stroji, na kterém probíhaly experimenty.

Čas klasifikace		**DataCTKLight** (14 dok.)		**DataCTKShort** (150 dok.)	
		Čas [s]	**ACC** [%]	**Čas** [s]	**ACC** [%]
samostatně	NB	0,311	0,5	0,644	0,547
	MaxEnt	0,683	0,643	10,472	0,534
	SVM	0,590	0,286	3,471	0,507
	celkem	**1,584**		**14,587**	
paralelně	NB	0,354		1,289	
	MaxEnt	0,804		10,987	
	SVM	0,685		4,185	
	celkem	**0,806**	0,4286	**11,175**	0,5267
Rozdíl času		0,778		3,412	

Tabulka 18: Doby zpracování klasifikace pro 14 a 150 dokumentů.

Čas klasifikace		**DataCTKMedium** (350 dok.)		**DataCTK** (750 dok.)	
		Čas [s]	**ACC** [%]	**Čas** [s]	**ACC** [%]
samostatně	NB	1,081	0,725	1,517	0,691
	MaxEnt	49,006	0,7	167,809	0,674
	SVM	7,803	0,724	17,130	0,692
	celkem	**57,890**		**186,456**	
paralelně	NB	0,354		1,289	
	MaxEnt	0,804		10,987	
	SVM	0,685		4,185	
	celkem	**54,226**	0,4286	**173,124**	0,5267
Rozdíl času		3,664		13,332	

Tabulka 19: Doby zpracování klasifikace pro 350 a 750 dokumentů.

Procesor	Intel(R) Core(TM)2 Duo CPU P8600 2.40GHz,
Typ systému	x64-based PC
Fyzická paměť (RAM)	4,00 GB
Operační systém	Microsoft Windows 7 Professional

Tabulka 20: Info o stroji, na kterém probíhaly experimenty.

9.7 Zhodnocení výsledků

Úspěšnost klasifikace se při používání samostatných klasifikátorů dosáhla hodnot lehce přes 70%. Největší úspěšnosti dosahoval klasifikátor SVM a poté stejně klasifikátory MaxEnt a NB. Při použití paralelní klasifikace se dosahovalo podobných výsledků metodou lineární kombinace klasifikátorů i metodou většinového hlasování, jako u samostatných klasifikátorů. Vždy se úspěšnost klasifikace pohybovala lehce nad 70%.

Doba klasifikace se paralelním zpracováním 3 klasifikátorů zmenšila oproti tomu, kdyby se klasifikátory spouštěly postupně (sériově). Při klasifikaci všech dokumentů se doba zkrátila o 15 sekund a bylo to vždy způsobeno čekáním na MaxEnt klasifikátor, který klasifikoval množinu dokumentů vždy nejdéle.

10 Závěr

Toto téma diplomové práce vzniklo ve spolupráci KIV a ČTK. Cílem práce bylo seznámit se a prozkoumat metody používané v oblasti analýzy sentimentu, na základě této analýzy potom zvolit vhodné metody a navrhnout systém pro rozpoznávání sentimentu.

Před samotným vypracováním diplomové práce bylo potřeba opatřit informace o analýze sentimentu. Většina materiálů je psaná v anglickém jazyce a čerpání informací z těchto zdrojů bylo časově náročné.

Nejdříve bylo potřeba se zaměřit na prostudování dostupných zdrojů, nalezení metod a dalších potřebných poznatků pro klasifikaci dokumentů dle sentimentu. Na základě dostupné literatury bylo rozhodnuto, že se dokumenty budou klasifikovat do tří tříd (pozitivní, negativní a neutrální) a budou použity tři klasifikátory (Naivní Bayesův, Maximální Entropie a SVM). Tyto klasifikační metody byly zvoleny z důvodu jejich vlastností (úspěšnost klasifikace a jejich rozšiřitelnost a použití, atd). Dále bylo potřeba zvolit klasifikační nástroj. Byl zvolen nástroj Minorthird, jehož součástí jsou všechny tři zvolené klasifikátory.

Původním záměrem bylo použít data z Ústavu formální a aplikované lingvistiky MFF UK, ale data bohužel nebyla poskytnuta. Data pro klasifikaci sentimentu byla poskytnuta ČTK v podobě článků o osobnostech. Bylo potřeba tyto články roztřídit a klasifikovat do daných tří tříd jednotlivé odstavce. Data pro klasifikaci byla potřebná pro vytvoření korpusu, který bohužel nebyl dostupný. Vytváření vlastního korpusu bylo časově náročné, ale bylo nutné korpus vytvořit pro zjištění úspěšnosti klasifikace. Byl vytvořen korpus o velikosti 750 dokumentů, tento byl roztříděn do tří kategorií v poměru 258 v pozitivní, 173 v negativní a 319 v neutrální kategorii. Pro tento postup třídění byla vytvořena aplikace, která ulehčuje anotování. Jedním z cílů této práce bylo zohlednit možnosti paralelního zpracování. Byla proto provedena klasifikace třemi klasifikátory najednou (paralelně) a výsledek byl určen jako kombinace dílčích výsledků.

Klasifikace dokumentů dle sentimentu probíhala na čtyřech vstupních množinách (malý počet až celý korpus) s využitím metody křížové validace v poměru 1:5. Tato metoda zajistila dostatečný počet dokumentů pro trénování a testování klasifikátorů. Větší poměr křížové validace nebylo nutné používat, protože korpus nebyl tak velký, aby se mohly zlepšit výsledky. Výsledky klasifikace byly udávány v úspěšnosti a chybovosti

klasifikace. Úspěšnost u klasifikace pouze jednou metodou NB byla 72% u MaxEnt 70% a SVM 72%. Úspěšnost klasifikace při použití paralelní klasifikace dvou klasifikátorů byla maximálně 73% při použití klasifikátoru NB a MaxEnt. Při použití paralelní klasifikace všech tří klasifikátorů dosáhla úspěšnost maximálně 73,2% při použití lineární kombinace klasifikátorů a nastavení vah $w_{NB} = 0{,}8$, $w_{MaxEnt} = 0{,}1$ a $w_{SVM} = 0{,}1$. Byla použita ještě metoda většinového hlasování klasifikátorů a úspěšnost byla opět 72%. Celková úspěšnost klasifikace je zlepšila pouze o 1%, což je pro praktické použití zcela zanedbatelné. Na základě dosažených výsledků bych doporučil použít pouze jeden klasifikátor a to Naive Bayesův, který dosahuje dobrých výsledků (72%) a je řádově několikrát rychlejší než nejpomalejší MaxEnt (100x u celého korpusu). U paralelní klasifikace bylo důležité zjistit doby průběhu klasifikace. Podle vstupních dokumentů se doby paralelní klasifikace urychlily řádově v pár sekundách. Při použití celého korpusu byla paralelní klasifikace rychlejší o 15s, než sériové zapojení klasifikace, s úspěšností klasifikace 70%.

Pro další vývoj je připravena aplikace pro klasifikaci využívající nástroj Minorthird a je spustitelná z příkazového řádku s řadou vstupních parametrů.

Další možná rozšíření

Jako první možné rozšíření by bylo dobré se pokusit zvětšit vstupní množinu dokumentů, aby v každé třídě bylo alespoň 500 dokumentů.

V experimentu byla použita jednoduchá metoda parametrizace a to četnost výskytu slov v dokumentu. V navazujících experimentech by bylo dobré vyzkoušet jiné metody parametrizace např. pomocí inverzní dokumentové frekvence nebo information gain, nebo mutual information.

Dále by bylo možné doplnit předzpracování o lemmatizaci a určení slovních druhů.

Seznam zkratek

ČTK - Česká tisková kancelář

XML - Extensible Markup Language

NB - Naive Bayes

MaxEnt - Maximum Entropy

SVM - Support Vector Machine

MAP - Maximum A Posteriori

MPI - Message Passing Interface

SPMD - Single Program Multiple Data

GUI - Grafické uživatelské rozhraní

Literatura

[1] *Bo Pang and Lillian Lee*
Opinion Mining and Sentiment Analysis Foundations and Trends in Information Retrieval Vol. 2, 2008.

[2] *Pang, Bo and Lee, Lillian and Vaithyanathan, Shivakumar*
Thumbs up?: sentiment classification using machine learning techniques Association for Computational Linguistics, 2002.

[3] *Bing Liu*
Sentiment Analysis and Subjectivity Department of Computer Science University of Illinois at Chicago, 2010.

[4] *Alexander Pak, Patrick Paroubek*
Twitter as a Corpus for Sentiment Analysis and Opinion Mining Université de Paris-Sud, Laboratoire LIMSI-CNRS, Bâtiment 508, F-91405 Orsay Cedex, France, 2010.

[5] *Andrew O. Arnold*
Exploiting domain and task regularities for robust named entity recognition Machine Learning Department School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, 2008.

[6] *Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze*
An Introduction to Information Retrieval Cambridge University Press Cambridge, England, 2009.

[7] *Christopher D. Manning, Hinrich Schütze*
Foundations of Statistical Natural Language Processing The MIT Press Cambridge, Massachusetts London, England, 2000.

[8] *Wenyuan Dai and Gui-rong Xue and Qiang Yang and Yong Yu*
Transferring naive bayes classifiers for text classification Department of Computer Science and Engineering Shanghai Jiao Tong University, Shanghai, China, 2007.

[9] *Canasai Kruengkrai and Chuleerat Jaruskulchai*
A Parallel Learning Algorithm for Text Classification Intelligent Information Retrieval and Database Laboratory, Department of Computer Science, Faculty of Science Kasetsart University, Bangkok, Thailand, 2002.

[10] **Česká tisková kancelář**
[*online*], 2011, dostupné z <http://www.ctk.cz/>

[11] **Wiki a Tutoriál Minorthird**
[*online*], 2012, dostupné z <http://sourceforge.net/apps/trac/minorthird/wiki >

[12] **A Minorthird Tutorial for ML Researchers**
[*online*], 2011, dostupné z <http://www.cs.cmu.edu/w̃cohen/10-707/m3rd-tutorial/#subpopulation >

[13] **OpenCV**
[*online*], 2012, dostupné z <http://opencv.willowgarage.com/wiki >

[14] **Mallet**
[*online*], 2011, dostupné z <http://mallet.cs.umass.edu/ >

A Uživatelský manuál

Kapitola se věnuje uživatelskému spuštění aplikací potřebných pro klasifikaci dokumentů. Všechny nástroje a aplikace jsou připraveny na spuštění pod operačním systémem Windows.

A.1 Aplikace na předzpracování dat z ČTK a tvorbu korpusu

Aplikace na tvorbu korpusu je popsána v kapitole 7.1 a obrázku 6.

Aplikace je vytvořena jako spustitelný soubor ***jar***. Příklad spuštění:

```
java -jar TvorbaKorpusu.jar
```

Aplikace se spustí a ovládání je pomocí GUI. Nápověda aplikace, je v příslušném menu aplikace. Výstupem aplikace je adresářová struktura. Vytvoří se adresář `sent`, ve kterém jsou další adresáře:

- **poz** - adresář se soubory, anotovaných odstavců do pozitivní kategorie,
- **neg** - adresář se soubory, anotovaných odstavců do negativní kategorie,
- **neut** - adresář se soubory, anotovaných odstavců do neutrální kategorie,
- **not** - adresář se soubory, neanotovaných odstavců.

A.2 Aplikace na parametrizaci korpusu

Aplikace je vytvořena pro parametrizaci korpusu metodou četnosti výskytu slov v dokumentu. Více o parametrizaci v kapitole 7.4.

Aplikace je vytvořena jako spustitelný soubor ***jar***. Příklad spuštění:

```
java -jar ParametrizaceKorpusu.jar <vstupni-soubor> <kategorie>
```

Aplikace má dva vstupní parametry, které nejsou nutné zadat. Do aplikace tyto parametry jsou možné zvolit i po spuštění bez parametrů:

- $< vstupni-soubor >$ - vstupní soubor nebo adresář se soubory pro parametrizaci, kde jsou samostatné odstavce, nikoli xml soubory,
- $< kategorie >$ - kategorie nebo třída, do které je soubor, nebo adresář se soubory zařazen.

Výstupem aplikace je jeden soubor s parametrizovanými vstupními soubory ve formátu `<vstupni-soubor>-<kategorie>.txt`, který lze použít jako vstupní soubor pro klasifikaci Minorthird. Doporučuje se zkombinovat všechny kategorie do jednoho souboru a ten potom použít jako vstup do Minorthirdu.

A.3 Klasifikace s Minorthird

Aplikace tvoří hlavní část diplomové práce pro klasifikaci dokumentů nástrojem Minorthird, jak je uvedeno v kapitole 8.1. Aplikace je tvořena z klasifikačního nástroje Minorthird a ovládá se z příkazového řádku s řadou parametrů, které lze nalézt v uvedené kapitole nebo v nápovědě aplikace (parametr `-help`).

Pro spuštění aplikace je využíván spustitelný soubor ***jar***, kterému se musí přidat i vstupní parametry.

A.3.1 Jednoduchá klasifikace

Spuštění jednoduché klasifikace (NB, SVM, MaxEnt samostatně) z parametrizovaných dat `-data dataCTK.txt` a uložení výsledků do `-saveAs klasf-dataCTK` s použitím křížové validace v poměru 1:5 `-splitter k5`.

```
java -jar ParalelniKlasifikaceMinorthird.jar -data dataCTK.txt
-splitter k5 -saveAs klasf-dataCTK
```

A.3.2 Paralelní klasifikace

Spuštění paralelní klasifikace (NB, SVM, MaxEnt) `-paralelClasf` z parametrizovaných dat `-data dataCTK.txt` a uložení výsledků do `-saveAs klasf-dataCTK` s použitím křížové validace v poměru 1:5 `-splitter k5`. Paralelní klasifikaci je zapotřebí zadat ještě alespoň 2 váhy (v tomto případě váhy pro NB a SVM klasifikátor `-vahaNB 0.8 -vahaSVM 0.1`).

```
java -jar ParalelniKlasifikaceMinorthird.jar -data dataCTK.txt
-splitter k5 -saveAs klasf-dataCTK -paralelClasf -vahaNB 0.8
-vahaSVM 0.1
```

A.3.3 Hledání optimálních parametrů

Spuštění aplikace pro hledání optimálního nastavení vah klasifikátorů `-testParamVahy` (NB, SVM, MaxEnt) z parametrizovaných dat `-data dataCTK.txt` a uložení výsledků do `-saveAs klasf-dataCTK` s použitím křížové validace v poměru 1:5 `-splitter k5`.

```
java -jar ParalelniKlasifikaceMinorthird.jar -data dataCTK.txt
-splitter k5 -saveAs klasf-dataCTK -testParamVahy
```

B Struktura přiloženého CD

Na přiloženém CD se nachází následující adresářová struktura.

- adresář `doc`:
  - adresář s touto prací, pdf této práce
  - adresář `obr` - obrázky použité v této práci
  - adresář `TeX` - zdrojové LaTeXové soubory této práce
- adresář `klasifikace`:
  - proběhlé experimenty s výsledky klasifikací
  - konečné výsledky a grafy klasifikací
- adresář `data`:
  - dostupný pouze na CD ve verzi pro ČTK
  - adresář `sentiment` - adresář s poskytnutými daty od ČTK
  - adresář `dataKlasifikace` - adresář s daty, na kterých probíhali experimenty
  - adresář `poz` - adresář se soubory anotovanými do pozitivní kategorie
  - adresář `neg` - adresář se soubory anotovanými do negativní kategorie
  - adresář `neut` - adresář se soubory anotovanými do neutrální kategorie
- adresář `projects`:
  - projekty z programu Eclipse, ve kterém byly tvořeny aplikace
  - `korpus` - adresář s aplikací na tvorbu korpusu
  - `parametrizace` - adresář s aplikací na parametrizaci korpusu
  - `Minorthird` - adresář s aplikací na klasifikaci pomocí nástroje Minorthird
- adresář `jar`:
  - všechny spustitelné aplikace
- adresář `install`:
  - instalační balíček neupravené verze Minorthird
- adresář `javadoc`:
  - java dokumentace pro vytvořené aplikace
- adresář `src`:
  - `korpus` - adresář se zdrojovými kódy aplikace na tvorbu korpusu

– `parametrizace` - adresář se zdrojovými kódy na parametrizaci korpusu
– `Minorthird` - adresář se zdrojovými kódy na klasifikaci pomocí nástroje Minorthird