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

Diplomová práce

Automatická detekce argumentace

Plzeň, 2017 Bc. Barbora Hourová

Prohlášení

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

V Plzni dne 11.5.2017

Bc. Barbora Hourová

Poděkování

Děkuji mému vedoucímu práce, Doc. Ing. Josefu Steinbergerovi, Ph.D., za rady a vstřícný přístup.

Také děkuji rodině a příteli za podporu a trpělivost.

Abstract

Automatic detection of argumentation

This thesis deals with automatic detection of argumentation, or specifically stance. In this case it means, that we have two statements and we try to determine, whether the statements agree or conflict with each other. One statement is always a theme and the second one is commentary to that theme. Argumentation is detected by supervised machine learning. This thesis propose a method, which classifies the commentaries with use of features. Feature is a vector representation of the commentary. Each commentary is represented by its atrribute, e.g. first word. We classify the comments to three classes: FAVOR, AGAINST and NONE (neither in favor or against). The method is tested on data corpus, where each commentary was annotated manually. The commentaries were downloaded from czech news server, from article discussions.

Abstrakt

Tato práce se zabývá automatickou detekcí argumentace, konkrétněji detekcí postoje (angl. stance). Zde to znamená, že máme dva výroky, a snažíme se určit, jestli spolu souhlasí, nebo jsou v rozporu. Jeden výrok je vždy téma, druhý je komentář k tomuto tématu. Argumentaci detekujeme pomocí strojového učení s učitelem. V práci je navržena metoda, která za pomoci příznaků (angl. Features) klasifikuje jednotlivé komentáře. Příznaky jsou vektorová reprezentace komentáře. Každý komentář je reprezentován pouze nějakou svou vlastností, například prvním slovem. Klasifikuje se do tří tříd: PRO, PROTI a NIC (není pro ani proti). Metoda se testuje na datovém korpusu, ve kterém byly ručně anotovány jednotlivé třídy. Komentáře byly stažené ze zpravodajského serveru, z diskuzí ke článkům.

Obsah

1 Úvod

Tato práce se zabývá automatickou detekcí argumentace. Argumentace je obor, který se zabývá strukturou diskuze. Věnuje se také stanovisku diskutérů. Podmnožinou analýzy argumentace je detekce postoje. Ta se snaží určit, jestli konkrétní výrok souhlasí s předem daným tématem. Například se snažíme určit, jestli věta „To prezident Zeman řekl správně" je pro nebo proti tématu „Miloš Zeman". Argumentaci budeme zkoumat v komentářích ke zpravodajskému textu.

Člověk sentiment, případně postoj určí většinou bez přemýšlení, na základě svých zkušeností. U počítačů se metody zhruba rozdělují na dvě skupiny. První jsou slovníkové metody, které rozhodují na základě velké databáze slov. Druhá skupina jsou metody strojového učení – takové, které se nejdříve natrénují na velkém množství dat, a pak jsou schopné označit zadaný výrok. V této práci používám metodu z druhé skupiny.

Detekce argumentace má široké využití. Analyzují se obvykle sociální sítě a diskuzní fóra, recenze (například filmů nebo restaurací), a také příspěvky z jiných druhů komunikace (například email). Analyzovat se dají i složitější a delší dokumenty, například zápisy z parlamentu nebo soudní dokumenty. Výsledky analýzy se dají použít různými způsoby, mimo jiné:

- v právním oboru pro lepší orientaci a rozhodování,
- pro lepší řízení diskuzí a porad (například v oblasti finančnictví),
- průzkum trhu (zjištění obecného mínění o značce nebo konkrétním výrobku),
- politické průzkumy (například předvolební průzkum, kde zjišťujeme, jak velkou pravděpodobnost úspěchu má daná politická strana/kandidát)

Analýzou sentimentu i argumentace už se různé výzkumy zabývaly. Nejvíce výzkumů se provádí na anglických textech. Vzniklé analyzátory ale nejsou snadno aplikovatelné na české texty. I v češtině byly provedené výzkumy a vznikly analyzátory, většina z nich se ale zabývá spíše analýzou sentimentu, než argumentace.

V první části této práce se seznámíme podrobněji se sentimentem, argumentací, postojem a s jejich analýzou. Projdeme existující výzkumy v tomto poli. Seznámíme se také s metodami strojového učení, které lze pro analýzu argumentace použít. Druhá část ze zabývá vytvořením programu, který detekuje argumentaci v českých datech.

První z cílů práce je vytvořit analyzátor, který detekuje argumentaci v českých datech. Bude používat strojové učení. V tomto případě strojové učení s učitelem, to znamená, že pro natrénování jsou potřeba už označená data. Pro češtinu, nakolik je autorce známo, žádný vhodný korpus dat neexistuje. Druhý cíl práce je proto vytvoření dostatečně rozsáhlého korpusu dat a jejich označkování. Vytvořený korpus se použije pro trénování analyzátoru a pro ověření jeho výsledků.

2 Metody strojového učení

Strojové učení je obor, který se zabývá tím, jestli a jak stroje dokáží napodobovat člověka v tom, že se učí. Učení v tomto případě obvykle chápeme jako schopnost stroje zlepšovat se, aniž by nové schopnosti byly explicitně naprogramovány. Přesněji, jak pojem definoval Mitchell (1997): Říkáme, že počítačový program se učí ze zkušenosti E (experience[1]), vzhledem ke skupině úkolů T (task) a ke způsobu měření výkonu P (performance), pokud se jeho výkon P při úkolech T zlepšuje se zkušenostmi ze skupiny E.

Tím způsobem, jak se dnes strojové učení používá, ho konkrétněji chápeme takto: je to schopnost z množství dat vyvozovat obecná pravidla, a ta potom aplikovat na nová, dosud neviděná data. Algoritmy strojové učení jsou schopny (v omezené míře) se přizpůsobit novým podmínkám. Například tím způsobem, že stroj na klasifikaci textů je schopen zpracovat data v novém jazyce, pokud dostane k dispozici dostatečné množství trénovacích dat.

Obor strojového učení souvisí s umělou inteligencí, ze které se původně vyvinul. Dá se říci, že strojové učení je jeden z přístupů, jak řešit umělou inteligenci. Souvisí také s dolováním dat (anglicky data mining). Obě skupiny metod zpracovávají velké množství dat s cílem zjistit z nich důležité informace, nebo něco předpovědět o nových datech. Ale strojové učení se snaží spíše reprodukovat schopnosti, které už člověk ovládá, tzn předem známé zkušenosti. Například překlady nebo klasifikace textů. Dolování znalostí se naopak snaží zjistit zcela nové informace, které ani člověku předtím nebyly dostupné, například předpovědět, která oblast prodeje bude nejvíce perspektivní.

Strojové učení je vhodné především k netriviálním úlohám. Je to totiž spíše heuristika, která se k ideálnímu výsledku pouze blíží. Dává pouze přibližné řešení, které nemusí být úplně optimální. Proto se nehodí na jednoduché úkoly, kde je známé použitelné algoritmické řešení. Ale je schopné dát použitelné výsledky na úlohy, pro které algoritmus neexistuje, nebo je algoritmus neúnosně náročný, ať už časově nebo výpočetně.

V porovnání s člověkem mají metody strojového učení často menší úspěšnost. Ale na složitých datech ani člověk nemusí mít stoprocentní úspěšnost. Uvedu zde jeden příklad za všechny: Habernal a kol. (2013) zkoumal automatickou klasifikaci textů. Trénovací data označovali ručně dva anotátoři. Ale ani ti se vždy neshodli, v některých

1 Protože se toto téma řeší v celosvětovém měřítku, tak se často používají anglické pojmy, kterým je mezinárodně rozumět. Proto budu u českých pojmů uvádět i anglické alternativy.

případech byl pro rozhodnutí třeba třetí, nebo až čtvrtý další anotátor. Nejlepší výsledek anotátora (pomocí F skóre, viz dále) byl 92 %, nikoliv 100%, jak by se dalo očekávat.

2.1 Druhy strojového učení

2.1.1 S učitelem a bez učitele

Učení s učitelem (anglicky supervised learning) potřebuje dopředu označená data. Program při trénování hledá vztahy v datech takové, aby trénovací data označil tak, jaká je požadovaná hodnota. Příkladem je klasifikace dokumentů do tříd „sport" a „politika". Trénovací dokumenty jsou označené, do které třídy patří. Klasifikátor se natrénuje tak, aby dokumenty označené jako „sport" zařadil všechny do jedné třídy, a dokumenty označené „politika" do druhé.

Učení bez učitele (anglicky unsupervised learning) také potřebuje trénovací data, ta ale nejsou označená. Program při trénování sám rozpozná vlastnosti dat a jejich vztahy, tak aby pro podobná data dával podobné výsledky. Příkladem je sdružování (anglicky clustering) podobných dokumentů do tříd, jejichž počet není dopředu znám. Například předložíme nějaký počet dokumentů, a program určí, že existují tři třídy dat - „sport", „politika", „bydlení". Příklady takových textů viz Obrázek 2.1

Obrázek 2.1: ukázkové texty ze zpravodajského serveru www.idnes.cz na témata sport (1.), politika (2.), bydlení (3.)

Existují také kombinované metody, kterým se část dat zadává označená a část neoznačená. Další možností, která stojí na pomezí učení s učitelem a bez učitele, je takzvané známkované učení. Při tom se programu sice dávají neoznačená data, ale jeho výsledek se hodnotí určitou známkou – nakolik je to správný výsledek. Podle tohoto hodnocení program upravuje vnitřní vztahy tak, aby se blížil požadovanému ohodnocení.

2.1.2 Učení dávkové a inkrementální

Pokud program používá dávkové učení, tak všechna trénovací data potřebuje najednou, před začátkem tréningu. Pokud bychom později chtěli naučenému programu předložit nová testovací data, tak je třeba ho natrénovat celý znova.

Inkrementální učení znamená, že program je schopen se „přiučit" i později. Například napřed natrénuje klasifikaci dokumentů na „sport" a „politika". Potom

klasifikuje testovací data. A potom mu předložíme nová trénovací data pro „automobily". Program si doplní vnitřní strukturu tak, že bude schopen klasifikovat všechny tři třídy - „sport", „politika" a „automobily".

2.1.3 Klasifikace, clusterování a regrese

Klasifikace je rozdělení dat do několika tříd, které musí být předem známé. Vždy se jedná o učení s učitelem, protože trénovací data musí být označena správnou třídou, do které patří.

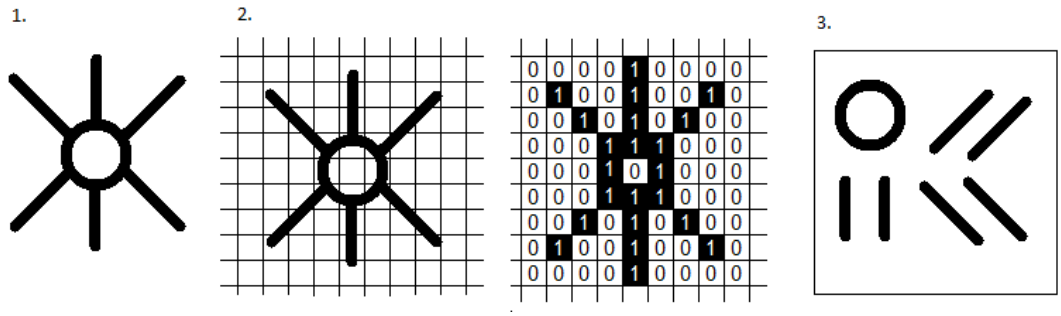
Clusterování je sdružování dat do shluků (anglicky cluster) podle toho, jak jsou si jednotlivé dokumenty podobné. Počet shluků můžeme dopředu určit, ale definice vlastních shluků jsou neznámé. Program se snaží, aby vzdálenosti a odlišnosti mezi jednotlivými shluky byly co největší. Naopak, uvnitř jednotlivého shluku musí být dokumenty co nejpodobnější. Ke clusterování se používá učení bez učitele. Protože dopředu neznáme, které shluky v datech jsou, tak nemůžeme trénovací data ohodnotit.

Regrese je předpovídání reálné hodnoty – tzn pro určité vlastnosti zkoumaného prvku je výsledkem jedno reálné číslo (Hastie a kol. 2009). Toto číslo může znamenat skoro cokoliv, například teplotu ve °C zítra v 9:00, pravděpodobnost, že pacient s danými příznaky má rakovinu, nebo míra podobnosti dvou genů.

2.2 Reprezentace dat

Rozpoznávat/klasifikovat můžeme například předměty, obrazy, dokumenty, zvukové záznamy, jevy, atd. Obecně říkáme, že rozpoznáváme objekty. Objekty v reálném světě jsou obvykle příliš složité na to, abychom je reprezentovali kompletně. Proto se reprezentují zjednodušenými modely. Snažíme se splnit dva protichůdné požadavky, a to, že model má co nejpřesněji reprezentovat reálný objekt, ale zároveň má být reprezentace co nejjednodušší. Objekty můžeme reprezentovat jedno-, dvou- nebo vícerozměrnými signály. Například obrazy se reprezentují dvourozměrnými vizuálními signály.

Jedna varianta modelu je reprezentovat objekt vektorem příznaků – každý prvek vektoru je hodnota jednoho příznaku. V anglických textech se pro příznak používá slovo feature. Obrázek 2.2 ukazuje příklad takové reprezentace. Druhá varianta je strukturní reprezentace. V té se objekt reprezentuje pomocí takzvaných primitiv, a vztahů mezi nimi (Matoušek a kol. 1994). Například v piktogramu slunce jsou primitiva kruh a čára, a vztah je „dotýká se", viz Obrázek 2.2. Dále v práci se budeme zabývat pouze reprezentací pomocí vektoru příznaků. Ta je vhodnější pro reprezentaci textů.



Obrázek 2.2: Piktogram slunce (1.) reprezentovaný buď dvourozměrným vektorem příznaků, kde každý příznak je bit na určité pozici a hodnota příznaku je buď 1=černá nebo 0=bílá (2.) nebo strukturně pomocí 6 čar, jednoho kruhu a vazby „dotýká se" (3.)

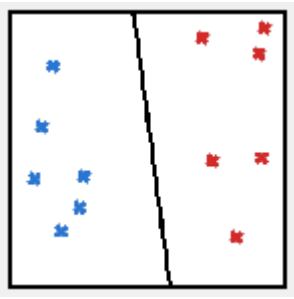
V málo případech jsou příznaky přímo hodnoty z reálného světa – například teplota ve stupních Celsia. Jindy mohou být reálné hodnoty zařazeny do kategorií, a objekt reprezentován pouze jednou z těchto kategorií (například barva objektu „modrá", přestože reálná hodnota může být libovolný odstín modré, například bleděmodrá.). U reprezentace textových dokumentů se často používají příznaky pouze v binární podobě. To znamená, že u příznaku rozeznáváme pouze, zda příznak je nebo není v textu obsažen, nepočítáme jednotlivé výskyty. Jako příklad uvedeme příznak „slova velkými písmeny" ve frázi „a PROTO říkám, NEKUŘTE". Hodnota příznaku je 1 (příznak je obsažen v textu), nezajímá nás skutečný počet slov velkými písmeny v textu (2 slova). Příznaky se, kvůli počítačovému zpracování, vyjadřují číselně. Ke každé kategorii se přiřadí určité číslo, například „modrá" v příkladu by byla reprezentována hodnotou 1, červená = 2, zelená = 3 a žlutá = 4.

2.3 Algoritmy strojového učení pro klasifikaci

V této kapitole uvedu některé z algoritmů strojového učení, které se běžně používají pro klasifikaci.

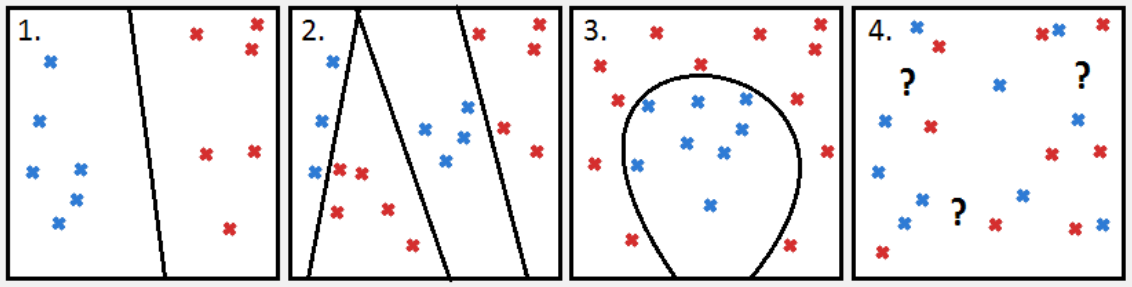
2.3.1 Lineární metody

Lineární metody fungují na jednoduchém principu. Klasifikované prvky se zobrazují jako body v prostoru. V tomto prostoru hledáme nadrovinu, která třídy odděluje. Snažíme se, aby tato nadrovina byla co nejvzdálenější od obou tříd, neboli od všech jejich bodů. Touto nadrovinou rozdělíme prostor na dvě části, kde každá část je klasifikovaná jako jedna třída. Toto znázorňuje Obrázek 2.3. Zástupce těchto metod je metoda nejmenších čtverců.



Obrázek 2.3: Dvě lineárně oddělené třídy

Lineární metody jsou jednoduché, ale nefungují pro všechny případy. Obrázek 2.4 znázorňuje různé třídy, z nichž některé jsou lineárně oddělitelné a jiné hůře nebo vůbec. Lineární metody se dají aplikovat také na transformovaný prostor – takový, který má větší dimenzi než původní zadání. To značně rozšiřuje jejich použitelnost (Hastie a kol. 2009).



Obrázek 2.4: třídy: - kompaktní, disjunktní, vzdálené, lineárně oddělitelné (1.)

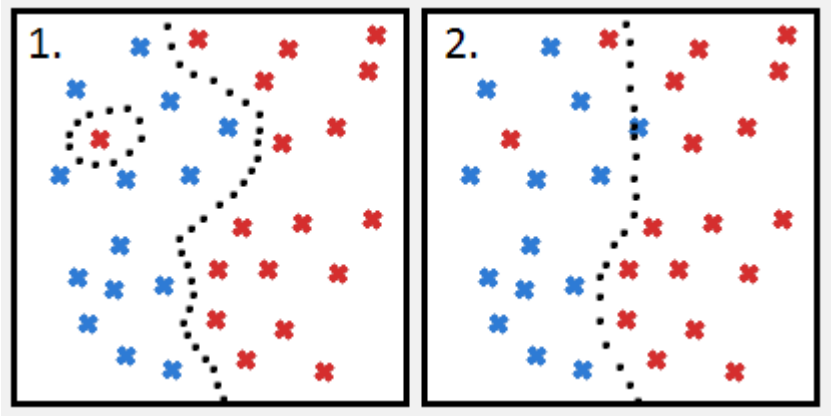
- nekompaktní, blízké, lineárně oddělitelné (2.)

- se složitou oddělující nadrovinou (3.)

- prolínající se, obtížně se určuje oddělující nadplocha (4.)

2.3.2 K-nejbližších sousedů

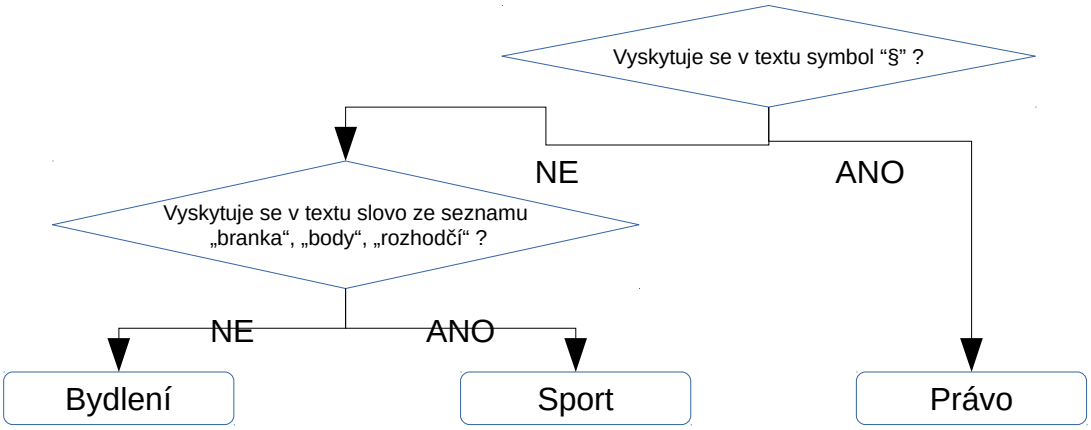
Na rozdíl od lineárních metod, k sousedů nemusí mít lineární hranici. Třída neznámého prvku se určuje podle toho, do které třídy patří nejbližší prvky z jeho okolí. Čím méně sousedů se používá, tím přesněji kopíruje hranici v trénovacích datech, ale tím náchylnější je k nepřesnosti testovacích dat, viz Obrázek 2.5. V porovnání s lineárními metodami je sice přesnější, ale méně stabilní (Hastie a kol. 2009).



Obrázek 2.5: Přibližné znázornění hranice vytvořené metodou k nejbližších sousedů pro k=1 (1.) a pro k= 7 (2.)

2.3.3 Rozhodovací stromy

Tyto algoritmy rozhodují o cílové třídě na základě rozhodovacího stromu. Ten se vytváří podle následujících kroků. Vezmeme jeden z atributů (příznaků) dat. Podle hodnot, které může nabývat, rozdělíme podstromy – pro každou hodnotu jeden podstrom. V dalším kroku vybereme jiný atribut, a rozdělíme data podle něj. Ve chvíli, kdy už se data nedají dělit, tak se vytvoří list. Jeho hodnota je třída dat, která v této podmnožině dat převažuje. Obrázek 2.6 je ukázka velmi zjednodušeného rozhodovacího stromu.

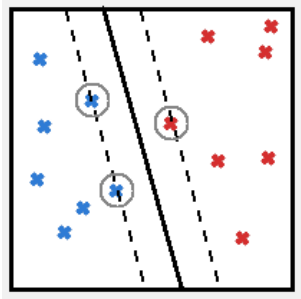


Obrázek 2.6: Zjednodušený rozhodovací strom pro klasifikaci na třídy „právo", „sport" a „bydlení".

2.3.4 SVM

SVM je zkratka anglického pojmu „support vector machines”. Je to metoda, která se snaží najít optimální separující nadrovinu. To znamená, že pokud lze třídy oddělit

úplně (tak že žádný prvek z trénovacích dat není klasifikován špatně), tak hledá nadrovinu, která je odděluje úplně. Pokud třídy nejsou v daném prostoru lineárně oddělitelné, tak hledá takovou rovinu, aby celková chyba byla co nejmenší. Nadrovina znamená, že data transformujeme do prostoru, který má více dimenzí než původní zadání. V tomto prostoru separující nadrovinu hledáme. Support vector by se dalo volně přeložit jako „podpůrné vektory". Vektory proto, že data se v daném prostoru vyjadřují jako body nebo vektory. Podpůrné proto, že k hledání separující nadroviny se používá několik podpůrných bodů. Podpůrné body jsou takové, které jsou nejblíž separující nadrovině a pomáhají určit, kde je vzdálenost mezi dvěma třídami největší. Obrázek 2.7 ilustruje dvě množiny se separující nadrovinou (v tomto případě přímkou).



Obrázek 2.7: Znázornění metody SVM. Kroužkem jsou označené podpůrné vektory. Čárkovaně je označený pruh okolo nadroviny, který metoda maximalizuje.

2.3.5 Maximum entropy

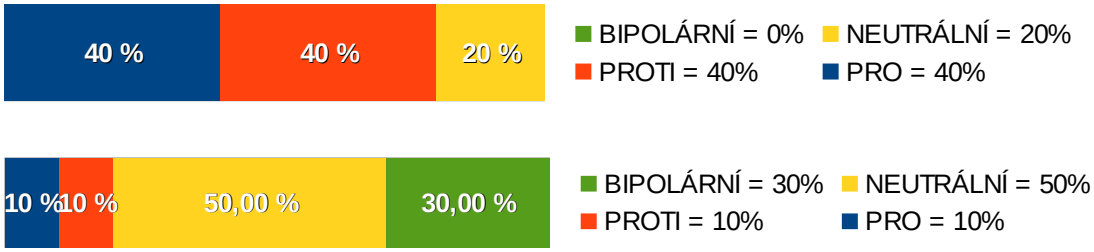
Algoritmus maximální entropie se snaží o co nejvíce rovnoměrné rozložení odhadů. Jinými slovy, modeluje všechno, co víme, ale důsledně se vyhýbá předpokládání něčeho, co nevíme (Berger a kol. 1996). Maximální entropie se dá dobře demonstrovat na příkladu. Mějme 4 třídy postoje: PRO, PROTI, NEUTRÁLNÍ, BIPOLÁRNÍ. Bipolární znamená, že text obsahuje zároveň souhlasný i nesouhlasný postoj. Pro účely příkladu budeme odlišovat tento postoj od neutrálního. Nejprve se dozvíme, že PRO a PROTI mají stejné zastoupení v datech. Můžeme tedy modelovat, že data jsou rozložená například takto (Obrázek 2.8):

Obrázek 2.8: Možné rozložení tříd, při splnění podmínky „počet PRO = počet PROTI"

To ale nevypadá moc logicky, nějaké neutrální texty se v datech určitě vyskytují. Lepší (rovnoměrnější) rozložení je určitě toto (Obrázek 2.9):

Obrázek 2.9: Rovnoměrné rozložení tříd, při splnění podmínky „počet PRO = počet PROTI"

Dále se z dat dozvíme, že neutrální texty spolu s třídou PRO dávají dohromady 60% dat. Opět máme několik možností, jak data rozdělit. Možné jsou například následující dvě rozdělení (Obrázek 2.10):



Obrázek 2.10: Možná rozložení tříd při splnění daných omezení

V tu chvíli se nabízí otázka, které rozdělení je nejrovnoměrnější. A ve chvíli, kdy budeme znát definici, tak jak máme toto rovnoměrné rozdělení vypočítat. Právě metoda maximální entropie toto řeší a umožňuje vypočítat nejrovnoměrnější rozdělení.

Ve skutečnosti jde o to, že máme stanovena nějaká omezení, která se dají vyjádřit pomocí rovnic. Tato omezení vyplývají z trénovacích dat a z vybraných příznaků (feature). Například to, že PRO a PROTI mají stejné zastoupení, se dá vyjádřit vzorcem:

$$p_{PRO} - p_{PROTI} = 0 \qquad (1)$$

kde p_{PRO} je pravděpodobnost zařazení do třídy PRO, analogicky funguje p_{PROTI}.

V triviálních úlohách (viz první část příkladu) se dá řešení vypočítat analyticky. Ale v reálných úlohách je to téměř nemožné. V metodě maximální entropie se pro každý příznak stanoví parametr. Potom vypočítáme parametry tak, abychom maximalizovali určitou funkci. Když je tato funkce v maximu, tak i entropii můžeme považovat za maximální. Příznaků (a tím pádem i parametrů) je mnoho. Proto by dokonce i iterativní optimalizace byla příliš výpočetně náročná. Proto se v algoritmu maximální entropie používá zjednodušení – s každým novým parametrem považujeme ostatní parametry za konstantní, a optimalizujeme pouze v jedné dimenzi (Berger a kol. 1996).

2.3.6 Naive Bayes

Naivní Bayesovský klasifikátor, jak už název napovídá, používá Bayesovu větu o podmíněné pravděpodobnosti jevu. Bayesova věta (viz vzorec 2) říká, že umíme vypočítat pravděpodobnost jevu A, za podmínky, že B je pravda. Vypočítáme to z pravděpodobnosti jevu B za podmínky A, a pravděpodobností jevů A a B nezávisle na sobě.

$$P(A|B)=\frac{P(B|A)*P(A)}{P(B)} \qquad (2)$$

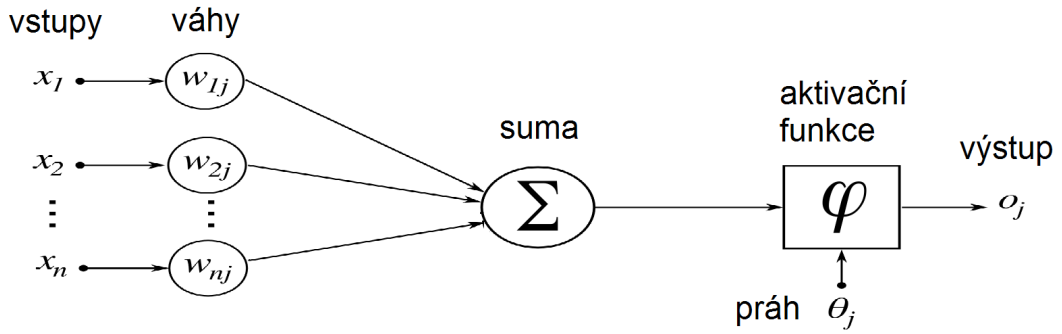
Z dat dokážeme určit pravděpodobnosti jednotlivých tříd nezávisle na sobě. Dokážeme také vypočítat pravděpodobnost B, že daný příznak má hodnotu X, pokud víme, že patří do dané třídy (jev A). Pro každou hodnotu příznaku a pro každou třídu vypočteme pravděpodobnost P(A|B), že text patří do třídy, pokud má příznak danou hodnotu. Tento výpočet je trénování klasifikátoru.

Při vlastní klasifikaci se dělá zjednodušující „naivní" předpoklad, že jednotlivé příznaky jsou na sobě nezávislé. Pro každou hodnotu každého příznaku zjistíme pravděpodobnosti jednotlivých tříd. Z dílčích pravděpodobností vypočteme celkovou pravděpodobnost pro každou třídu, a vybereme nejpravděpodobnější třídu.

Přestože naivní Bayesovský klasifikátor dělá zjednodušení, tak má efektivitu srovnatelnou se složitějšími klasifikátory, například SVM. Naopak, kvůli zjednodušení je velmi dobře škálovatelný, co se týče dimenze dat, neboli počtu příznaků. Na rozdíl od mnoha dalších metod používá analytické výpočty, takže má lineární složitost.

2.3.7 Neuronové sítě

Neuronové sítě napodobují biologické neuronové sítě. Skládají se z neuronů, a vazeb (synapsí) mezi nimi. Každý neuron má několik vstupů, aktivační neboli přenosovou funkci a jeden výstup. Obrázek 2.11 znázorňuje model neuronu. Výstupy a vstupy neuronů jsou propojené synapsemi. Hodnoty, které jsou výstupem celé sítě, závisí na přenosové funkci neuronů, a vahách jednotlivých spojení.



Obrázek 2.11: Model umělého neuronu.

Přenosová funkce je daná typem sítě. Může to být například jednotkový skok, hyperbolický tangens nebo mnoho jiných, různě složitých funkcí. Tato funkce transformuje vstupy neuronu na výstupy. Váhy spojení se nastavují během fáze trénování. Neuronové sítě se mohou učit s učitelem i bez učitele. Existují také neuronové sítě, kde učení neprobíhá, váhy jsou nastavené napevno už od začátku.

Učení s učitelem probíhá tak, že se síti předloží nějaký vstup, a požadovaný výstup. Síť vygeneruje svůj výstup, a potom iterativně upravuje váhy tak, aby se skutečný výstup blížil požadovanému výstupu. V každém kroku se váhy spojení upraví o nějakou hodnotu. Učení končí, když je rozdíl skutečného výstupu od požadovaného výstupu v dané toleranci.

Pokud se síť učí bez učitele, tak je cílem nastavit váhy tak, že pro podobné vstupy dává síť podobné výstupy. Princip učení je ten, že když se aktivují dva neurony zároveň, tak se synapse mezi nimi posílí.

Neuronové sítě jsou určené také topologií. Vazby mohou být pouze s dopředným šířením, nebo rekurentní. Sítě mohou být jedno- nebo vícevrstvé. Nejjednodušším zástupcem jednovrstvých sítí je jednoduchý perceptron s jedním neuronem.

2.4 Vyhodnocení úspěšnosti klasifikátoru

Pokud máme klasifikátor a data, tak nás zajímá také to, jak dobrý ten klasifikátor je. Pro určování úspěšnosti daného klasifikátoru se používá několik metod. Nejprve k nim definuji pojmy.

- Chyba typu I (anglicky false positive, FP) je falešně pozitivní chyba. To znamená, že případ, který do dané třídy nepatří, klasifikujeme jako že do třídy patří.
- Chyba typu II (anglicky false negative, FN) je falešně negativní chyba. To znamená, že klasifikátor určil, že prvek do třídy nepatří, přestože tam ve skutečnosti patří.
- Pravdivě pozitivní (anglicky true positive, TP) je výsledek v případě, že klasifikujeme prvek jako patřící do třídy a je to pravda.
- Pravdivě negativní (anglicky true negative, TN) výsledek funguje analogicky – prvek do třídy nepatří a klasifikátor ho do třídy nezařadí.

2.4.1 Metriky

Zde jsou uvedené nejčastěji používané metriky. Používají se samozřejmě ještě další metriky, které zde nejsou uvedené.

1. Přesnost (anglicky precision) udává, kolik z těch prvků, které klasifikátor vybral, do třídy opravdu patří. Vzorec je:

$$precision = \frac{TP}{(TP + FP)} \tag{3}$$

2. Úplnost (anglicky recall) udává kolik z těch prvků, které do třídy opravdu patří, klasifikátor vybral. Vzorec:

$$recall = \frac{TP}{(TP + FN)} \tag{4}$$

3. Accuracy (obvykle správnost) udává, kolik z celkového počtu prvků klasifikátor klasifikoval správně. Vzorec:

$$accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{5}$$

4. F1 skóre kombinuje precision a recall ve vzorci:

$$F_1 = \frac{2 * precision * recall}{(precision + recall)} \tag{6}$$

2.4.2 Přetrénování

Přílišná snaha o co nejlepší klasifikaci na trénovacích datech může vést k přetrénování (anglicky overfitting). To znamená, že klasifikátor se naučí spíše jednotlivé vzorky dat, než nějaká obecná pravidla. Výsledky na testovací datech jsou potom podstatně horší než na trénovacích datech.

2.5 Výběr dat pro testování

Máme metriky pro zjišťování úspěšnosti klasifikátoru. Na jakých datech ale budeme klasifikátor testovat? K výběru testovací množiny se používají různé metody. Část z nich zde zmíním.

2.5.1 Jednoduché rozdělení dat

Rozdělení dat na trénovací a testovací. Trénovacích dat by mělo být více, používá se například rozdělení 20% = testovací, 80% = trénovací množina. Na trénovacích datech se klasifikátor natrénuje, a potom se změří jeho úspěšnost na testovacích datech. Toto je jednoduchá varianta vhodná pro případy, kdy máme dostatek dat. Když bychom měli málo dat, tak se může stát, že trénovací data nebudou vhodný reprezentativní vzorek dat.

2.5.2 Křížové ověření

Metoda křížového ověření (anglicky cross validation) funguje tak, že se data rozdělí na K stejných částí (proto se používá název K-fold cross validation). Jedna z těchto částí se ponechá pro testování, na zbylých částech se natrénuje klasifikátor. Potom se určí jeho úspěšnost na vybrané testovací části. Potom toto provedeme znova, ale vybere se jiná část pro testování. Postup opakujeme pro všechny části. Křížová validace zmírňuje problém s nedostatkem dat.

Výsledek křížové validace částečně závisí na rozdělení dat – podle toho, jakým způsobem rozdělíme data na trénovací a testovací, podle toho se mohou lišit výsledky křížové validace. Například, v nejhorším případě, data obsahují stejný počet prvků ze tří tříd a používáme klasifikátor, který zařazuje všechny prvky do největší třídy. Pokud bychom použili 3-násobnou křížovou validaci, a do každé části dat zahrnuli všechny prvky z jedné třídy, klasifikátor by všechny testovací prvky zařadil špatně. Možnost nevhodného rozdělení dat by eliminovala kompletní křížová validace. To je průměr výsledků přes všechna možná rozdělení na K částí. Ta je obvykle ale příliš výpočetně náročná. (Kohavi 1995)

Hraniční verzí křížové validace je, když K se rovná počtu prvků, takže testujeme vždy jen na jednom prvku. Toto se anglicky nazývá leave-one-out. Tato verze je vždy kompletní (z principu musíme otestovat všechny prvky).

Pokud nechceme dělat kompletní křížovou validaci, ale chceme zmenšit riziko nevhodného rozdělení dat, tak můžeme křížovou validaci opakovat několikrát. Při každém pokusu rozdělíme data na K částí jiným způsobem. To dává lepší odhad, ale za cenu větší náročnosti.

Další možností je stratifikovaná křížová validace. Ta se provádí pouze jednou, ale data se rozdělují speciálním způsobem. Data se rozdělují pro každou třídu zvlášť. Tím se zaručí, že v testovacích i trénovacích datech bude přibližně stejné rozdělení jednotlivých tříd, jako v původních datech.

2.5.3 Bootstrap

Metoda Bootstrap také vybírá trénovací prvky z celkových dat. Důležité je, že jich vybírá N (velikost korpusu dat), ale vybírá s opakováním (Hastie a kol. 2009). Část prvků (přibližně 37%) tedy není vůbec vybraná, a ta se použije k testování.

2.6 Výběr příznaků

Na začátku kapitoly jsem psala o tom, že reálný objekt je reprezentován množinou příznaků. Pro správnou klasifikaci je důležité vybrat ty příznaky, které dávají u klasifikace nejlepší výsledky. Například, předměty mohou mít vlastnosti: délka, šířka, hloubka, barva, hmotnost, tvar, objem,... Ale pokud klasifikujeme předměty na „dlouhé" a „krátké", tak ze všech těchto vlastností stačí vybrat jednu, a to délku. Snažíme se vybrat takovou množinu příznaků, která je relevantní, ale není redundantní. Pro zjištění těch správných příznaků určíme trénovací data některou z metod, a na nich zjišťujeme, které příznaky jsou nejúspěšnější při klasifikaci.

Snažíme se vybírat příznaky takové, aby byly relevantní vzhledem k danému úkolu. Například pro klasifikaci na dlouhé a krátké předměty je příznak „barva" naprosto nerelevantní. Také je ale cílem vybírat příznaky tak, aby nebyly redundantní. Například, pokud známe hustotu materiálu a objem předmětu, tak jsme schopni zjistit váhu. Proto příznak „váha" pravděpodobně pro klasifikátor bude zbytečný.

2.6.1 Výběr, extrakce a konstrukce

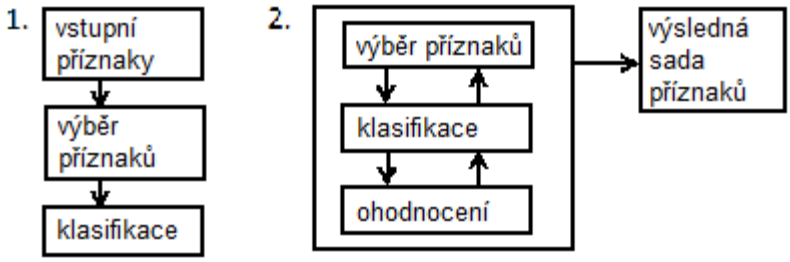
Pro výběr podmnožiny vhodných příznaků se v angličtině používá pojem feature selection. To se liší od metody extrakce příznaků (anglicky feature extraction), kde se vytvářejí nové příznaky z existujících pomocí funkčních transformací. Další jiná metoda je konstrukce příznaků (anglicky feature construction), která se snaží doplnit chybějící informace o vztazích mezi příznaky, a tím vytváří nové příznaky (Liu a Motoda 2002).

2.6.2 Wrapper a filter metody

Metody pro výběr příznaků se dají zařadit do dvou skupin. První skupina jsou takzvané „obalovací" (anglicky wrapper) metody. Obalovací se jmenují proto, že

algoritmus pro výběr příznaků „obaluje“ vlastní klasifikátor, který zde funguje jako černá skříňka (Kohavi a kol. 1997). Výsledky klasifikace se používají pro ohodnocení sady příznaků. V praxi to funguje tak, že se klasifikátor spustí na testovacích datech s danou množinou příznaků a výsledky se zaznamenají. Toto se opakuje pro různé sady příznaků, a výsledky pro jednotlivé sady příznaků se porovnávají. Metoda je znázorněná na Obrázek 2.12.

Filtrační metody (anglicky filters) vybírají příznaky nezávisle na vlastním klasifikátoru. Místo toho se vybírají na základě dat. Název „filtr“ se používá proto, že příznaky se filtrují ještě před vlastní klasifikací, během předpřípravy (preprocessing) dat. Výpočetně jsou méně náročné, než obalovací metody. Ale jejich hlavní nevýhodou je, že neberou v potaz to, jak výběr příznaků ovlivní výsledky klasifikátoru. Znázornění viz Obrázek 2.12.



Obrázek 2.12: Znázornění filtračních (1.) a obalovacích (2.) metod

2.6.3 Wrapper algoritmy

Zde popíšu některé z obalovacích algoritmů, které se pro výběr příznaků používají.

Horolezecké (anglicky hill-climbing) algoritmy fungují na principu prohledávání stavového prostoru. Prostorem jsou všechny možné sady příznaků – každá sada je jeden stav. Přechody mezi stavy se provádějí přidáváním a ubíráním příznaků. V každém kroku se prozkoumá nejbližší okolí stavu. Pokud žádný z okolních stavů není lepší, tak se končí. Jinak se algoritmus přesune do nejlepšího sousedního stavu a pokračuje v hledání. Symbolický zápis algoritmu viz Tabulka 2.1. Jedna z možností je například prázdný počáteční stav bez příznaků. Přechod mezi stavy (v kroku 2) se pak děje přidáním jednoho příznaku.

1. Přiřaď v = počáteční stav 2. Prohledej okolí v: proveď všechny možné operace s v a získej tím jeho potomky 3. Ohodnoť všechny potomky 4. Přiřaď v' = potomek s nejvyšším ohodnocením 5. Pokud nový stav v' je lepší než současný stav v, tak přiřaď $v = v'$; jdi na 2. 6. Vrať v

Tabulka 2.1: Symbolický zápis horolezeckého algoritmu

Best-first (nejlepší na začátku) výběr provádí expanzi uzlů podobně, jako horolezecké algoritmy. Ale další krok nevybírá jen z nejbližších sousedů. Pamatuje si všechny dosud prohledané stavy, a z těch vybírá nejlepší. Tím umožňuje vyhnout se lokálnímu minimu.

Genetické algoritmy použijí několik nejlepších stavů, a vzájemně je kombinují pro získání další generace. Ta se bude ohodnocovat v dalším kroku. Stavy se kombinují podobně jako u biologických organismů, pomocí mutací, křížení a výběru.

Simulované žíhání na rozdíl od horolezeckého algoritmu neprohledává nejbližší stavy, ale dělá náhodný krok. Pokud je nový stav lepší, než současný, vždy ho přijme. Na rozdíl od horolezeckého algoritmu ale může dočasně přijmout i horší řešení, a tím se vyhnout lokálnímu maximu.

Existuje samozřejmě ještě mnoho jiných algoritmů pro výběr příznaků, například tabu-search a další.

3 Sentiment a argumentace

Předmětem této práce je detekce argumentace. Podrobněji se bude zabývat rozpoznáním postoje (v angličtině se obvykle používá pojem stance). Argumentace také souvisí s analýzou sentimentu. Proto tyto pojmy podrobněji rozvedeme v následující kapitole.

Analýza sentimentu, popřípadě detekce argumentace, se řadí do širší oblasti zpracování přirozeného jazyka. To je metoda umělé inteligence, která se zabývá počítačovým porozuměním textu. Také umožňuje, aby počítače dokázaly efektivně manipulovat s textem, například klasifikovat nebo sumarizovat dokumenty. Je to oblast, která se zkoumá už delší dobu, a provedené výzkumy přinesly užitečné výsledky. Stále ale existuje mnoho věcí k dalšímu zkoumání.

Počítače mají velký význam v analýze a zpracování přirozeného jazyka, zejména proto, že množství textů je obrovské, a stále rychle roste. V on-line debatách, na zpravodajských serverech, v archivech knihoven a na sociálních sítích je a stále přibývá množství nových textů, ať už dlouhých dokumentů, či komentářů o několika málo slovech. Ruční zpracování textů se využívá, ale takovéto množství dat nelze ručně zpracovat.

Jak uvádí Ikonomakis (2005), klasifikaci a porozumění textu ztěžuje například heterogenita dat. Data jsou z mnoha různých zdrojů, a i v jednom zdroji každý článek používá jiný styl jazyka, jiné fráze nebo odborné termíny. Data také mohou být zašuměná – obsahovat nepřesnosti, nespisovné výrazy, gramatické chyby (ať už stavba věty, nebo např. „i/y”). To je další výzva pro počítačové zpracování sentimentu a argumentace. Člověk také může používat ironii, která se psaném textu rozpoznává hůře.

3.1 Sentiment

Sentiment je „pocitové naladění” textu. Texty mohou být různě velké celky, od jednotlivých slov a frází, přes věty až po celé dokumenty nebo jejich skupiny. Sentiment se obvykle rozlišuje na pozitivní a negativní. Pozitivní sentiment vyjadřuje kladné, příjemné pocity autora. Pozitivní sentiment obsahuje například věta „Online diskuze mě baví.”, příkladem negativního sentimentu může být prohlášení „Nesnáším smrad z cigaret!”. V některých případech se rozlišuje i intenzita pocitu, například těchto pět kategorií: silně pozitivní, pozitivní, neutrální, negativní, silně negativní.

Existují také texty, které sentiment neobsahují, nelze ho určit, nebo obsahují jak pozitivní tak negativní sentiment. V některých případech se používá rozlišení textů na objektivní a subjektivní. Subjektivní texty obsahují sentiment (ať už pozitivní či negativní), objektivní texty sentiment neobsahují (Gamon 2004). V tom případě se

občas mluví o „neutrálním" sentimentu. Jako příklad objektivní věty bez sentimentu uvedeme konstatování „Prezident zastupuje stát navenek." V některých případech je hranice mezi objektivním a subjektivním textem nejasná, některé zdánlivě objektivní texty mohou obsahovat určitý implicitní sentiment. Toto více popisuje Greene (2009).

V malém množství se vyskytují také texty, které obsahují oba sentimenty zároveň – takzvané bipolární texty. Takové texty se obtížně zpracovávají a zhoršují výsledky automatického zpracování. Navíc, tato třída má obvykle málo zástupců. Pokud bychom ji chtěli klasifikovat (pomocí strojového učení s učitelem), tak se pro tuto třídu těžko bude trénovat klasifikátor. Dosud funkční postup byl, takové texty z trénovacích dat úplně vyřadit, a při klasifikaci tuto třídu neuvažovat (Habernal 2013).

Sentiment pomáhá pochopit, co autor textem míní. Proto je jeho analýza důležitou součástí zpracování textů. Automatické zpracování sentimentu má rozmanité způsoby využití, část z nich zde vyjmenujeme:

- Zpětná vazba od zákazníků – sentiment umožňuje správné rozlišení, kdo by ji měl zpracovávat (Gamon 2004). K tomu je důležité znát, jestli je to kritika (na základě kritiky firma může zlepšovat nedostatky) nebo chvála (určuje silné stránky které firma může propagovat).
- Politika – analýza sentimentu zjišťuje, jestli jsou reakce na politickou kampaň převážně pozitivní nebo negativní. Také může být součástí předvolebního průzkumu – zjišťuje se, který z kandidátů/stran má pozitivní a který má negativní ohlasy ve společnosti.
- Sociální média – zde může být množství příspěvků tak velké, že se nedá ručně zpracovat. Zabývají se různými tématy. Sociální média jsou obvyklé místo, kde lidé vyjadřují své názory.

  Společnosti v ČR v poslední době rozvíjejí marketing přes sociální sítě – komunikaci se zákazníky. Tato oblast se rozvíjí a firmy do ní investují. V mnoha případech se ale sentiment stále označuje ručně (Habernal 2013). To je ukázka skutečnosti, že výzkum sentimentu v češtině se ještě má kam rozvíjet.
- Oblíbenost značky nebo produktu – analýza sentimentu umožňuje firmě porovnání sebe s konkurencí.
- Úspěšnost reklamní kampaně – umožňuje zjistit, zda byla reklamní strategie úspěšná a jak moc. Na základě toho se dá rozhodovat, jestli příště zvolit stejnou nebo jinou strategii.
- Recenze filmů, restaurací a podobně – názor ostatních bývá užitečný pro širokou veřejnost při výběru, na který film nebo do které restaurace půjdou.

3.2 Argumentace

Detekce argumentace (v angličtině používaný pojem argumentation mining) stojí na pomezí teorie argumentace, zpracování přirozeného jazyka a dolování informací. Zabývá se diskuzemi - zjištěním jejich struktury a určením stanoviska diskutérů.

U struktury jde jednak o vnější strukturu, a jednak o vnitřní strukturu argumentů. Vnější struktura určuje, co je základní jednotkou diskuze, a jaké jsou jejich vzájemné vztahy. Za základní jednotku se dá považovat argument. Vzájemné vztahy argumentů může být koordinace, subordinace, nebo argumenty dohromady tvoří vícenásobný argument. (Mochales 2011) vzájemnou strukturu argumentů zobrazuje jako strom. To znamená, že argumenty jsou zároveň premisami (předpoklady) dalších argumentů, a ze všech argumentů v textu se tvoří celkový závěr.

Vnitřní struktura zjišťuje, z čeho se skládají jednotlivé argumenty. (Mochales 2011) používá celkem jednoduchou definici – argument se skládá z jedné nebo více premis, a jednoho závěru. Jednoduchý postup proto je, nejprve určit, které věty vůbec jsou součástí argumentů. V diskuzi totiž mohou také být věty, které jsou neutrální, neslouží k argumentaci. Například, v diskuzi na téma „Existuje Bůh?", je nesouvisející věta „Včera jsem četl noviny.". U vět, které patří do argumentace, se následně určí, jestli je to premisa nebo závěr.

Těžším úkolem potom je, určit hranice argumentu – kde začíná a kde končí, které věty jsou jeho součástí. K tomu lze mimo jiné použít metodu sémantické souvislosti – věty, které se věnují stejnému tématu, patří do jednoho argumentu.

Podle (Mochales 2011) existují dva hlavní směry argumentace: formální a neformální. Formální argumentaci a logiku používáme například při odvozování důkazů v matematice. Neformální argumentace se vyskytuje v běžných každodenních debatách, i ve složitějších textech. Neformální argumentace nemusí mít přesně definované ani úplné argumenty. A používají se v ní také zdánlivě nelogické postupy, jako je opakování toho samého argumentu, nebo charisma řečníka. Při rozpoznávání argumentace v přirozeném jazyce musíme brát ohledy na tyto specifičnosti neformální argumentace.

Výzkum (Mochales 2011) ukazuje, že lepších výsledků lze dosáhnout na lépe strukturovaných datech. Výzkum probíhal na dvou různých korpusech dat . Jeden byl obecný korpus dat z různých zdrojů (časopisy, soudní záznamy, diskuze,...). Druhou částí byla sada dokumentů od Evropského soudu pro lidská práva (anglicky European Court of Human Rights, ECHR). Tato část byla podstatně lépe strukturovaná a používá jednotný způsob argumentace. Na druhé části vzrostla správnost klasifikátoru (accuracy).

Detekce argumentace se zabývá strukturou celé diskuze. Proto se dá použít pouze na složitějších a delších textech. Na jedné větě, jediném komentáři nebo tweetu detekce argumentace nezíská dostatek dat. A například vztah mezi argumenty ani nelze určit, protože máme k dispozici vždy jen jeden argument najednou. Je nutné podotknout, že detekce argumentace se nezabývá správností argumentů, ani tím, jestli jsou úspěšné, logické nebo správné. Jde pouze o určení, kterou část textu autor použil k argumentaci.

Detekce argumentace má využití například v právu – pomáhá k lepší správě případů a jejich prohledávání. Umožňuje také například vizualizovat strukturu diskuze.

Další oblast využití je u novinových článků, ke kterým se vedou online diskuze. Automatickou sumarizaci těchto diskuzí zkoumali Kabadjov a kol. (2015) v úkolu Online Forum Summarization v rámci iniciativy MultiLing 2015. Jejich výzkum stojí na pomezí dolování argumentace, sumarizace dokumentů a analýzy sentimentu.

3.3 Postoj

Postoj (anglicky stance) je postoj autora vůči určitému tématu. Postoj je celkový subjektivní názor autora, téma může být objekt, něčí názor, myšlenková idea, postup, situace ve světě, hnutí, produkt atd. Téma je předem dané, tím se postoj liší od sentimentu (Mohammad 2016). Pokud u textu určujeme sentiment, můžeme určit zároveň cíl tohoto sentimentu. Například ve větě „Jsem rád, že volby vyhrál Miloš Zeman" je sentiment pozitivní, a cíl sentimentu je Miloš Zeman. Postoj vůči tématu „Miloš Zeman" je také pozitivní. U této věty ale můžeme zjišťovat i postoj k úplně jiným tématům, například Karel Schwarzenberg, nebo kouření. Postoj vůči Karlu Schvarzenbergovi je negativní. Postoj vůči kouření nelze určit, věta s kouřením nesouvisí.

(Mohammad 2016) uvádí, že nedostatek ukazatelů k určení postoje (pro nebo proti), nemusí znamenat, že text je neutrální. Může to znamenat pouze to, že postoj textu nejsme schopni určit. Dále však uvádí také to, že počet textů, které jsou opravdu vysloveně neutrální, je velmi malý. A z toho důvodu u použitých dat nerozlišuje texty neutrální od textů, u kterých postoj pouze nelze určit. Místo toho je všechny zařazuje do třídy „neither" (tzn. třída „žádný sentiment").

Postoj souvisí se sentimentem, ten lze využít k detekci postoje. Zopakujme, že u textu lze rozlišovat sentiment a cíl sentimentu. Důležitý není jen sentiment sám o sobě, ale důležitý není ani cíl sentimentu samotný. Důležité jsou v kombinaci. Uvedeme příklad k debatě na téma „Gay rights" - práva gayů. Když autor mluví o Bibli (tzn. téma je Bible), jeho postoj určíme těžko. Když je autor rozzlobený (tzn. sentiment věty je negativní), jeho názor také nejsme schopni určit. Ale pokud je autor rozzlobený na Bibli, můžeme celkem s jistotou tvrdit, že autor se zastává práv gayů.

Stupněm mezi rozpoznáváním sentimentu a postoje může být aspektově založená analýza sentimentu. Ta nezkoumá celkový sentiment věty, ale sentiment v souvislosti s cílovými entitami (například restaurace) a jejich aspekty (například cena, nebo zda je restaurace kuřácká). Aspektově založenou analýzu sentimentu prováděl například Brychcín a kol (2014).

Postoj souvisí samozřejmě také s argumentací. Detekce postoje se dá považovat za jeden z mnoha úkolů při rozpoznávání argumentace (v širším slova smyslu). Ale také naopak – to, že někdo argumentuje pro nebo proti něčemu, se dá využít při zjišťování jeho postoje. Jak intuice napovídá, tak ukazuje i výzkum – lepší výsledky při detekci postoje má kombinace argumentování a sentimentu, než každá vlastnost zvlášť (Somasundaran 2010).

Další z přístupů k analýze postoje využívá vztahy mezi jednotlivými promluvami (Thomas 2006). Tento přístup využívá skutečnosti, že se dají určit vztahy mezi jednotlivými texty. Zde se jako jednotka, u které určujeme postoj, používá celá promluva od jednoho autora. Pokud víme, že několik promluv je od stejného autora, téměř jistě budou mít texty stejný postoj vůči tématu. Pokud víme, že jeden autor odkazuje na druhého, tak pravděpodobně oba hájí stejné stanovisko. To už ale není tak jisté, autor může například souhlasit s jedním výrokem druhého, ale přesto mít obecně opačný názor. Příkladem může být věta „Zemana nemám rád, ale tohle řekl správně.", kde autor obecně zaujímá opačné stanovisko, než Miloš Zeman, ale v tomto případě na něj odkazuje a souhlasí s ním. Výzkum (Thomas 2006) ukazuje, že je lepší vazby mezi jednotlivými autory použít pouze jako volné vazby, které sice zvyšují pravděpodobnost, že promluvy patří do stejné kategorie, ale nakonec je možné, že každá promluva bude mít opačný názor.

Poslední dobou jsou oblíbené debaty, kde účastníci zaujmou jednu ze dvou stran vůči tématu – pro nebo proti (Somasundaran 2010). A potom v debatě hájí stanovisko své strany [2]. Pro zkoumání postoje jsou takovéto debaty vhodné. Nejlepší jsou hodně kontroverzní témata, protože o těch se hodně mluví. Proto je dostupných hodně dat ke zkoumání. Kontroverzní musí být také proto, aby od obou stran byl dostupný dostatek textů – abychom neměli texty pouze pro, nebo pouze proti danému tématu. Pokud nám jde o rozlišení tří tříd (pro, proti, neutrální), tak potřebujeme samozřejmě dostatek dat ze všech tří tříd (Mohammad 2016).

Rozpoznávání postoje je teprve v začátcích a výsledky nejsou dokonalé. Předpokládá se, že se výsledky budou zlepšovat s lepším porozuměním úkolu a se získáváním dalších dat (Mohammad 2016).

---

2 http://www.opposingviews.com/ - anglický web s debatami
http://www.idnes.cz/ - český zpravodajský web, umožňující diskuzi o tématu

4 Korpus dat

Metoda navržená v rámci této práce funguje na principu klasifikace, a to pomocí strojového učení s učitelem. Aby se dal klasifikátor natrénovat a otestovat, bylo třeba vytvořit korpus vhodných dat. Klasifikátor funguje pro češtinu, data jsou pouze v ČJ. Jako zdroj textů byl zvolen zpravodajský portál iDNES.cz. Webový portál na rozdíl od tištěných médií umožňuje stáhnout texty automaticky ve velkých množstvích. Portál iDNES.cz umožňuje svým návštěvníkům, aby se ke článkům vyjadřovali v diskuzích. Jeho diskuze jsou často používané, a k většině článků se návštěvníci opravdu vyjadřují. Právě z těchto diskuzí můžeme získat dostatečné množství komentářů pro korpus.

4.1 Struktura korpusu

Vzorem pro data byl korpus z mezinárodního semináře na analýzu sentimentu – SemEval (Semantic Evaluation) 2016. Jeden z úkolů na SemEval byla právě detekce postoje u výroků na sociální síti Twitter. Zde byl korpus jednoduchý textový soubor, kde na každém řádku byl jeden výrok. Na každém řádku jsou 4 pole oddělená tabulátorem: ID, téma, text a postoj příspěvku. Strukturu řádku znázorňuje Obrázek 4.1. Téma je slovo nebo fráze, ke každému tématu je v korpusu mnoho příspěvků. ID je jedinečné číslo komentáře v celém datovém korpusu, přes všechna témata. Postoj (anglicky stance) je jedna ze tří hodnot: PRO(FAVOR), PROTI(AGAINST), NIC(NONE).

Semináře se zúčastnila i Západočeská univerzita (Krejzl, Steiberger 2016). Pro tento úkol jsem v rámci diplomové práce vytvořila baseline program, který jsem dále vyvíjela. Proto bylo logické, aby nově vytvořený český korpus měl stejnou strukturu jako anglický. Program tak mohl jen s malými úpravami fungovat pro oba korpusy.

```
<ID>    <téma>      <text>                                                  <stance>
500     Atheism     Lord, You are my Hope! In You I will always trust. #SemST   AGAINST
1700    Hillary Clinton Lol why do so many people around here hate Hillary?  #SemST FAVOR
```

Obrázek 4.1: Struktura řádku a dva příklady z korpusu dat v SemEval;

V SemEval bylo pět témat, pro každé řádově stovky příspěvků. Výroků bylo celkem přibližně 3000. Počet témat jsem zmenšila, ale pro každé téma český korpus obsahuje více příspěvků. Témata používám dvě (Miloš Zeman a Zákaz kouření v restauracích), ke každému je přes 2500 příspěvků. Celkem příspěvků je 5423.

Podle vzoru SemEval bylo rozhodnuto, že prvním výrokem bude jen téma článku, nikoliv celá souvislá věta. K tématu je možné nalézt podstatně více souvisejících komentářů, než k nějaké konkrétní větě z článku. Druhý výrok je celý jeden komentář, který se může mít délku od jednoho slova (nebo emotikonu) až po několik odstavců. Maximální délka komentáře, kterou server iDNES.cz připouští, je 2000 znaků nebo 10 odstavců. Obvyklá délka příspěvků je jedna nebo několik vět.

První verze českého korpusu byla prezentována na konferenci WIKT[3] a Data a Znalosti (Steinberger a Krejzl 2016). Data byla otestována na stejném klasifikátoru, který byl použit pro SemEval, s malými úpravami. Složení první verze korpusu viz Tabulka 4.1 a výsledky testování viz Tabulka 4.2. $F1_{PRO/PROTI}$ je počítané na jednoúrovňové klasifikaci - tzn na takové, kde klasifikátor rovnou rozděluje data do tří tříd. Použil se pouze průměr $F1_{PRO}$ a $F1_{PROTI}$, průměr $F1_{NIC}$ se do této hodnoty nezahrnul.

Téma	PRO	PROTI	NIC	CELKEM
Miloš Zeman	180	170	300	750
Zákaz kouření v restauracích	170	250	390	810

Tabulka 4.1: Složení verze korpusu, prezentované na konferenci WIKT a Data a Znalosti

Téma	F1 (PRO/PROTI)	F1 (PRO/PROTI/NIC)
Miloš Zeman	.4347	.5204
Zákaz kouření v restauracích	.4562	.5400

Tabulka 4.2: Výsledky testování na první verzi korpusu.

4.1.1 Výroky

Podle zadání diplomové práce, vstupní data jsou věta (výrok) ze zpravodajského textu a komentář k tomuto výroku. Klasifikátor je postavený tak, že první výrok není třeba explicitně zadávat. První výrok je přítomen v trénovacích datech jako téma. Všechny příspěvky v korpusu jsou anotované vzhledem k tomuto tématu. Druhý výrok je vždy jeden komentář z trénovacích/testovacích dat.

Ve vytvořeném korpusu je téma sousloví. Ale dalo by se to chápat i tak, že téma je vhodná věta nebo několik vět. V korpusu je ohodnocený postoj diskutéra vůči tématu=sousloví. Ale pokud je diskutér proti tomuto tématu, téměř jistě bude i proti větě, která toto téma vyjadřuje. Jako příklad uvedu větu „[...] a příští rok tak bude zakázáno kouření v barech a restauracích.". Kdo je proti zákazu kouření, ten nebude souhlasit s touto větou. Takže se dá říci, že tématem celého korpusu je tato věta. Tím pádem klasifikátor určuje postoj testovaného komentáře vůči ní. To můžeme tvrdit dokonce i když anotace korpusu zůstane beze změny (komentáře anotované PRO, zůstanou PRO; analogicky zůstanou stejné třídy PROTI a NIC).

Pokud chceme použít současnou anotaci korpusu, tak na větu-téma máme následující omezení:

3 Workshop on Intelligent Knowledge Technologies (pracovní dílna v oblasti inteligentních a znalostních technologií)

1. Buď věta musí vyjadřovat téma samotné (například předchozí věta „[...] a příští rok tak bude zakázáno kouření v barech a restauracích.").

2. Nebo musí věta vyjadřovat kladný postoj vůči původnímu tématu. A to pokud možno vůči celému tématu (Miloš Zeman), nikoliv jen vůči části tématu nebo vůči konkrétnější verzi tématu („Miloš Zeman ustoupil Putnovi").

Ve chvíli, kdy se jedno z těchto omezení dodrží, tak je možné jako první výrok (ten, vůči kterému určujeme postoj) použít jak větu z novinového článku, tak nějaký komentář. Níže uvádím ještě několik příkladů pro ukázku a pro vysvětlení.

1. Věta *Zeman [...] dokázal, že je skvělý rétor.*

   Toto je věta ze zpravodajského článku „Skvělý rétor i podporovatel propagandy. Politiky Zemanův projev rozdělil." Kdo souhlasí s touto větou, ten je ve většině případů také PRO téma „Miloš Zeman". Postoj libovolného komentáře vůči této větě proto můžeme natrénovat na současném korpusu beze změny.

2. Komentář k tématu „Zákaz kouření v restauracích":

   10667 [...] jsem rád, ten zákon měl platit už dávno. :-) PRO

   Tento komentář se dá použít jako první výrok beze změny korpusu. Kdo souhlasí se souslovím „Zákaz kouření v restauracích" (téma korpusu), tak ten jistě také souhlasí, že zákon měl platit už dávno.

3. Komentář k tématu „Miloš Zeman":

   17346 Zeman je nejlepší hlava státu všech dob v česku R^ PRO

   Tento komentář se dá použít jako první výrok beze změny korpusu. Komentář, který je PRO téma „Miloš Zeman", ten také souhlasí s tímto komentářem.

Další možnost jak určit argumentaci mezi dvěma komentáři, by bylo vytvoření další části korpusu - takové, kde by se postoj určoval vůči nově vybranému komentáři. Tématem této nové části by tedy byl přímo komentář, nikoliv jen sousloví. Při tom bychom ale mohli narazit na problém, že k jednomu komentáři nebude dostatek odpovědí. Teoreticky by bylo možné komentář zredukovat, vypustit některá slova, nebo použít pouze hlavní myšlenku tohoto komentáře. Potom by mělo existovat více komentářů, které by se daly použít do nové části korpusu.

4.2 Témata

Při výběru témat jsem musela splnit několik požadavků. Téma musí většina lidí znát, a musí se o něm hodně diskutovat. Například Brexit (odstoupení Velké Británie z Evropské Unie) se často vyskytoval v médiích, ale diskutovalo o něm jen málo (desítky až několik málo stovek komentářů k jednomu článku). Proto nebyl použitelný jako téma

do korpusu. Navíc, v diskuzi musí být zastoupeny názory z obou stran. To jednak znamená, že lidé v diskuzi jedna musí vyjadřovat nějaké stanovisko k tématu, a jednak musí být zastoupené stanovisko pro, i stanovisko proti. Obě stanoviska by se měla vyskytovat v diskuzi pokud možno rovnoměrně, i když to téměř nelze splnit. Z tohoto důvodu nešlo použít například téma „Uprchlíci". Komentářů v diskuzích k tomuto tématu jsou tisíce, ale přes 90% vyjádřených názorů je proti. Ukázka diskuze na téma „uprchlíci“ viz Obrázek 4.2.

```
21 Uprchlíci Ať muslimáci táhnou i s celým Bohoušem.
   PROTI
22 Uprchlíci  Proč sakra? Komu se chtějí zalíbit ? Přijde mi
to jako vtip, po takové záplavě smutných zpráv .Proč se nás
nikdo nezeptá na názor? Bojím se o své malé děti. To chce
demonstraci, řekněte kdy a kde. Už nikdy se nebudeme cítit
bezpečně ;-( Ať jdou opravu do háje ! PROTI
```

Obrázek 4.2: Ukázka příspěvků v diskuzi na iDNES.cz na téma „Uprchlíci“

Na rozdíl od korpusu v SemEval jsem vybrala jen dvě témata. Předpoklad je, že bude lepší se věnovat dvěma tématům více do hloubky, než více tématům povrchně. Diskuze k jednomu článku mají řádově stovky až několik tisíc příspěvků. Ne všechny příspěvky bylo možné použít. Proto, aby bylo dostatečné množství dat, tak se k jednomu tématu použilo několik článků.

4.3 Získávání dat

Pro stažení dat z diskuzí na serveru iDNES.cz jsem vytvořila program, který z HTML struktury dat získá texty komentářů. Program stáhne celou stránku diskuze, a potom projde HTML a podle klíčových HTML tagů a atributů najde text každého komentáře. Pro toto se používá anglický pojem scraping. Diskuze se zobrazuje po stránkách, proto i program projde zadané stránky diskuze a po jedné je zpracuje. Dále je seznam článků, které jsem použila pro tvorbu korpusu.

Články k tématu „Zákaz kouření v restauracích“:

- Chystaný zákaz kouření je moc obecný. Poslanci už řeší, jak ho změkčit (http://zpravy.idnes.cz/protikuracky-zakon-projednava-snemovna-fdu-/domaci.aspx?c=A150707_224255_domaci_jkk)
  příspěvky 800-1249
- Nechte v restauracích vyhrazené prostory pro kuřáky, navrhli poslanci (http://zpravy.idnes.cz/nechte-v-restauracich-vyhrazene-prostory-pro-kuraky-navrhli-poslanci-1z9-/domaci.aspx?c=A160308_180642_domaci_kop)
  příspěvky 1250-1349

- Boj o zákaz kouření v hospodách začíná naostro. Někde na zákon nečekají (http://zpravy.idnes.cz/boj-o-zakaz-koureni-v-hospodach-zacina-naostro-nekde-na-zakon-necekaji-1kk-/domaci.aspx?c=A151207_071428_domaci_kop)
  příspěvky 1350-1699
- Úplný zákaz kouření v restauracích od května prošel. Kuřárny nebudou (http://zpravy.idnes.cz/druhy-pokus-zakazat-koureni-v-restauracich-fet-/domaci.aspx?c=A161208_210805_domaci_kop)
  příspěvky 10000-13599

Články k tématu „Miloš Zeman“:

- Všichni jsme Gogo? V Německu jsou teď všichni Syřané, prohlásil Zeman (http://liberec.idnes.cz/prezident-zeman-v-lomnici-hovoril-o-angele-merkelove-a-uprchlicich-1d4-/liberec-zpravy.aspx?c=A160226_122747_liberec-zpravy_ddt)
  příspěvky 1-727
- Zeman podepíše jmenování Putny profesorem, ale titul mu předá ministr (http://zpravy.idnes.cz/zeman-s-ministrem-fialou-o-jmenovani-putny-profesorem-p14-/domaci.aspx?c=A130522_114234_domaci_kop)
  příspěvky 15000-16676
- Trump mluvil se Zemanem. Zve ho do Bílého domu, přijal pozvání do Česka (http://zpravy.idnes.cz/trump-mluvil-se-zemanem-pozval-ho-na-navstevu-do-bileho-domu-prk-/domaci.aspx?c=A161206_153538_domaci_kop)
  příspěvky 17000-19029
- Skvělý rétor i podporovatel propagandy. Politiky Zemanův projev rozdělil (http://zpravy.idnes.cz/reakce-politiku-na-projev-prezidenta-zemana-fzh-/domaci.aspx?c=A161226_135753_domaci_pku)
  příspěvky 19100-20336
- Zeman po tragédii v Berlíně odmítl uprchlíky, Babiš kritizuje Merkelovou (http://zpravy.idnes.cz/zeman-odmitl-uprchliky-na-uzemi-ceska-d8o-/domaci.aspx?c=A161220_123031_domaci_kop)
  příspěvky 20400-21883
- Zeman má v blahopřejném dopise Trumpovi chyby, odhalil překladatel (http://zpravy.idnes.cz/v-zemanove-dopise-trumpovi-jsou-chyby-dhn-/domaci.aspx?c=A161115_094244_domaci_jj)
  příspěvky 22000-23224

Složitější bylo zpracování komentářů, které obsahují emotikony (smajlíky). Emotikon na iDNES.cz není vyjádřen jako obyčejný text, ale jako obrázek. Program

umí obrázky rozpoznat, a místo obrázku do textu vloží atribut „alt", neboli alternativní text k emotikonu. Seznam smajlíků, které iDNES.cz umožňuje, viz Tabulka 4.3

Obrázek	Alternativní text	Popis
	:-)	úsměv
	:-(	mračí se
	;-D	velký úsměv
	:-/	nespokojený
	;-(	mračí se
	:-P	vyplazuje jazyk
	;-)	mrká
	X	křížek
	8-o	brýle
	;-€	rozzuřený
	;-O	řvoucí
	[>-]	česká vlajka
	EU	vlajka EU
	?	otazník
	!	vykřičník
	R^	ruka s palcem nahoru
	Rv	ruka s placem dolů
	V	srdce
	!!	dvojitý vykřičník

Tabulka 4.3: Emotikony v diskuzích na iDNES.cz

Uvažovala jsem o stahování diskuze z mobilní verze serveru iDNES.cz. Ta ale má stejně složité HTML, jako klasická verze. Je jednodušší pouze vzhledově. Navíc se URL adresa klasické a mobilní verze diskuze neliší, proto by se programu hůře přistupovalo k mobilní verzi.

Program umožňuje několik druhů výstupu. Jedno je přímé zobrazení získaných textů. Toto slouží pouze pro kontrolu. Další formát je ten, který byl použitý v SemEval (každý komentář na jeden řádek). Pro další zpracování dat program umí další dva výstupy. Jednak je to ukládání do databáze (každý komentář jako jeden záznam). A jednak lze uložit každý komentář do samostatného textového souboru. ID komentáře je v názvu souboru. Z databáze se příspěvky samozřejmě také dají exportovat ve formátu CSV (comma separated values) nebo TSV (tab separated values). To znamená, že se

vytvoří testový soubor, kde jednotlivá pole jsou oddělená středníky nebo tabulátory. Ze stejných formátů lze importovat i textový soubor zpět do databáze.

4.4 Anotace komentářů

Po stažení se komentáře ručně ohodnotily. Nejdříve se rozdělily podle toho, jestli souvisí s daným tématem. V diskuzích je mnoho (přes 50%) příspěvků, které s tématem nesouvisejí. Například u článku „Zeman podepíše jmenování Putny profesorem, ale titul mu předá ministr“ je přibližně 500 příspěvků souvisejících a 1000 nesouvisejících. U diskuze na téma „Zákaz kouření v restauracích“ bylo více souvisejících příspěvků. U nesouvisejících se často jedná o příspěvky, které jsou velmi krátké, příliš dlouhé, obsahují pouze emotikon nebo odpovídají na jiný komentář. Příklady viz Obrázek 4.3. Nesouvisející komentáře vůbec nebyly zahrnuty do korpusu, aby korpus pro trénování byl dostatečně kvalitní. Vyřadilo se také malé množství dalších příspěvků: takové, které byly celé ve slovenštině, angličtině nebo vyjadřovaly zároveň postoj pro i proti (tzv. bipolární postoj). Tabulka 4.4 uvádí, kolik kterých příspěvků bylo z korpusu vyřazeno. Pouze související příspěvky se dále zpracovaly.

```
<ID>  <téma>      <text>
15753 Miloš Zeman Vy v tom máte praxi, že to tak dobře znáte?
16076 Miloš Zeman Ubozaku...
16160 Miloš Zeman ;-D ;-D ;-D ;-D ;-D R^
16519 Miloš Zeman Že už jich tu bylo... ;-D
```

Obrázek 4.3 Příklady komentářů, které nesouvisí s tématem.

Téma	Komentáře					
	anglicky	slovensky	mixed	nesouvisí	**souvisí**	celkem
Zákaz kouření v restauracích	0	6	5	1205	**2785**	4001
Miloš Zeman	2	20	102	5640	**2638**	8402

Tabulka 4.4: Počty komentářů. Tučně jsou označená data použitá v korpusu.

Související příspěvky se potom ohodnotily do tří tříd: NEUTRÁLNÍ, PRO a PROTI, podle toho, jaký mají postoj vůči tématu. Počty jednotlivých tříd v datech viz Tabulka 4.5. Je vidět, že diskutující více vyjadřují názor proti, než pro téma. Třída PROTI má zhruba 1,75 násobný počet než třída PRO. Hodnotily se v náhodném pořadí, aby anotátor neviděl strukturu diskuze. Ze struktury diskuze by se totiž dalo odvodit více informací o postoji, než jen ze samostatného příspěvku. To by ale znevýhodnilo automatický klasifikátor, který hodnotí každý komentář zvlášť.

Téma	Komentáře			
	PRO	PROTI	NIC	celkem souvisí
Zákaz kouření v restauracích	744	1280	761	2785
Miloš Zeman	691	1263	684	2638

Tabulka 4.5: Počty jednotlivých tříd ve finálních datech.

V první verzi korpusu se příspěvky nerozlišovaly podle toho, zda s tématem souvisí. Do tří tříd podle postoje byly rozděleny všechny příspěvky z diskuze. Příspěvky, které nesouvisely s tématem se zařadily do třídy NEUTRÁLNÍ (nevyjadřuje postoj). Takto se ale stalo, že do třídy NEUTRÁLNÍ byly zařazené i příspěvky, které jednoznačně vyjadřovaly postoj, jen k jinému tématu. Příklad takového příspěvku viz Obrázek 4.4. V pozdější verzi korpusu byly příspěvky vytříděny a byl vytvořen korpus pouze dat, která souvisejí s tématem. Porovnání výsledků klasifikace viz kapitola 8 Testování a výsledky.

```
   <ID>    <téma>                                  <text>
   11202   Zákaz kouření v restauracích            Hospody    jsou
hlavně líhně alkoholiků a jak víme alkoholismus je jedna z
největších a nejhorších závislostí která likviduje rodiny.
```

Obrázek 4.4: Příspěvek vyjadřující postoj k jinému tématu

Při anotování postoje se objevily problémy, které z úkolu dělají ne zcela silně definovaný. Pokud komentáře nějaký z těchto problémů obsahovaly, anotátoři se často neshodli. Zde je seznam problémů, příklady viz Obrázek 4.5.

- fráze, a slova, která v souvislosti s tématem mají jiný význam;
- spojitost mezi osobami. Například, pokud někdo haní Karla Schwarzenberga, nebo haní odpůrce Miloše Zemana, tak obvykle souhlasí s tématem „Miloš Zeman“.
- Když se píše o tématu nepřímo.
- Ironie
- slova, která mohou ale nemusí mít negativní/pozitivní sentiment.

```
   PROTI   15849 Miloš Zeman     Bohužel i nás všech. Je to náš
lidový prezident s nevyléčitelnou virózou.
   (u tématu Miloš Zeman „viróza" obvykle znamená „alkoholismus"
v negativním  smyslu)
   PRO     15312 Miloš Zeman     Skupina antizemanovců se nemůže
stále smířit s  tím, kdo je prezidentem, (…)
   NIC     15439 Miloš Zeman     Bohužel  místo  silné  osobnosti
zvolili alkoholika. :-/
   (není jisté, jestli zvolenou osobou je zrovna prezident)
   PROTI   15313 Miloš Zeman     Miloš je opravdu chudák,a ještě
k tomu ty vyrózy  ..... ;-D
   PROTI   15965 Miloš Zeman     Prezident zkratař... ;-D
   ("zkratař" znamená, že autor je proti prezidentovi, nebo je
to jen komentář situace?)
```

Obrázek 4.5: Příklady komentářů s problematickým hodnocením: fráze (15849), spojitost mezi osobami (15312), nepřímo zmíněné téma (15439), ironie (15313), slova s nejistým sentimentem (15965).

4.4.1 Shoda anotátorů

Ve zpracování přirozeného jazyka je obvyklé, že někdy nelze zcela přesně definovat kategorie, do kterých chceme data klasifikovat. Úkol v této diplomové práci není jiný. Příklady obtížně zařaditelných příspěvků jsou v předchozí kapitole, viz Obrázek 4.5. Právě proto se určuje shoda mezi anotátory. Ta určuje hlavně dvě věci:

1. Jak dobře jsou kategorie definované. To znamená, jak dobře vůbec můžeme určit kategorii.
2. Jak spolehlivá a jednoznačná je tato konkrétní anotace dat. Neboli, jak moc věříme svému ohodnocení dat.

V jednodušší verzi by stačilo určit, v kolika procentech z celkového počtu příspěvků se anotátoři shodli. To ale nebere v úvahu možnost, že by se anotátoři shodli náhodou. Například, pokud by kategorie byly jen dvě - ANO/NE, tak by byla 50% šance, že se anotátoři shodnou náhodou.

Proto se zavádí měřítko Cohenova Kappa. Ta je určená vzorcem:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \qquad (7)$$

kde p_0 je pozorovaná shoda mezi anotátory a p_e je pravděpodobnost náhodné shody.

V rámci této diplomové práce Kappa určím ve dvou případech.

1. Shoda u tématu „Miloš Zeman". Celý korpus anotovala autorka práce. Pro vypočítání shody menší část korpusu (302 příspěvků) anotoval druhý anotátor[4]. Na této části příspěvků se určí Kappa.
2. Shoda u tématu „Zákaz kouření v restauracích“ mezi dvěma anotátory. Část tématu anotovala autorka práce. Druhou část (2203 příspěvků) tohoto tématu anotovali nezávisle dva anotátoři[5]. Na druhé části je určená kappa.

Shoda u tématu „Miloš Zeman“:

		anotátor #2			
		NEU	PRO	PROTI	celkem
anotátor #1	NEU	92	14	14	120
	PRO	8	19	1	28
	PROTI	37	1	117	155
	celkem	137	34	132	303
P_0 (shoda celkem)		92	19	117	228
					0,7525 (75%)
		NEU	PRO	PROTI	P_e
Shoda náhodou		54,26	3,14	67,52	124,92
					0,4125 (41%)
Kappa	(P0 - Pe) / (1 - Pe) = (228/303 - 125/303) / (303/303 - 125/303) = (228 - 125) / (303 - 125) = 0.579				

Shoda u tématu „Zákaz kouření v restauracích“:

		anotátor #2			
		NEU	PRO	PROTI	celkem
anotátor #1	NEU	636	91	134	861
	PRO	188	272	13	473
	PROTI	365	22	485	872
	celkem	1189	385	632	2206
P_0 (shoda celkem)		636	272	485	1393
		NEU	PRO	PROTI	P_e
Shoda náhodou		464,06	82,54	249,82	796,44
Kappa	(P0 - Pe) / (1 - Pe) = (1393 - 796) / (2206 - 796) = 0.423				

4 Ing. Peter Krejzl

5 Ing. Peter Krejzl, Mgr. Martina Krejzlová

Výsledky jsou tedy $\kappa_{\text{Miloš Zeman}} = 0.579$ a $\kappa_{\text{Zákaz kouření v restauracích}} = 0.423$. Kappa s hodnotou 0 by znamenala pouze náhodnou shodu, hodnota 1 znamená úplnou shodu. Čísla okolo 0.5 se tedy dají hodnotit jako mírná shoda.

To ukazuje, že úkol není zcela dobře definovaný. Shoda u tématu „Zákaz kouření v restauracích“ je zhoršená výběrem souvisejících příspěvků. Tyto se vybíraly pouze z těch, kde oba anotátoři určili třídu NIC. To znamená, že z korpusu se odstranily pouze příspěvky, kde se oba autoři shodli, což zhoršilo shodu. Původní kappa pro nevytříděný korpus = 0.517. To ale není o tolik lepší shoda.

Shoda u tématu „Miloš Zeman“ mohla být ovlivněná také tím, že se shoda určovala pouze na podmnožině dat (300 příspěvků z 2 638).

Také je vhodné si povšimnout toho, že je celkem 35 příspěvků u zákazu kouření, kde oba autoři určili postoj, ale každý jiný. To může ukazovat na nepřesnou definici toho, co vlastně znamená, že komentář je PRO nebo PROTI. Případně to může ukazovat obtížnost rozhodnutí bez znalosti kontextu. Přehled počtů, jak byly příspěvky anotovány, viz Tabulka 4.6.

celkem	shoda (golden data)	neshoda (NIC+STANCE)	neshoda (oba anotátoři STANCE)
2202	1388	779	35

Tabulka 4.6 Porovnání počtů příspěvků podle různé shody anotátorů.

U příspěvků, kde se první dva anotátoři neshodli bylo třeba rozhodnout, kterou anotaci použít. Toto rozhodoval třetí anotátor. U příspěvků, kde jeden z anotátorů určil NIC, se vždy použil postoj určený druhým anotátorem (PRO nebo PROTI). U příspěvků, kde každý z anotátorů určil postoj, třetí anotátor rozhodl, který z postojů se použije.

4.4.2 Data se 100% shodou

Protože shoda mezi anotátory není moc velká, tak se testovala klasifikace na tzv. „golden" datech - na komentářích, kde oba anotátoři zvolili stejnou třídu. Tím se ale hodně zmenšil počet příspěvků. Příspěvků k tématu „Zákaz kouření v restauracích", na kterých se anotátoři shodli je 1388. To jsou zhruba 2/3 z celkového počtu 2202 příspěvků k tomuto tématu. Porovnání s celým souvisejícím korpusem viz kapitola 8 Testování a výsledky.

4.5 Syntaktické a morfologické předzpracování

Některé z příznaků používají nejen text samotný, ale také jeho sémantickou a syntaktickou strukturu. Proto bylo třeba komentáře více zpracovat. Pro to se použil

nástroj UDPipe[6]. Nástroj popsali Straka a kol. (2016). Jako vstup má textový soubor s jedním komentářem, a jako výstup soubor ve formátu CoNLL-U[7], kde každé slovo je na jednom řádku, a jsou k němu uvedené informace jako je slovní druh, větný člen, morfologické znaky a závislost na jiném větném členu.

Pro pomoc se zpracováním souborů byly použity jednoduché skripty ve Windows Batch.

4.6 Automatická data

V první fázi bylo anotováno jen něco přes 1000 komentářů. Protože to je stále celkem málo a protože ruční anotace je zdlouhavá, testovala jsem poloautomatické vytváření dat. Z diskuze k dalšímu článku jsem stáhla příspěvky, a ukládala jsem k nim nejen ID, ale také jejich autora. Potom jsem vybrala autory s dostatečně velkým počtem příspěvků (alespoň 20). Od každého jsem ručně anotovala 10 příspěvků. Někteří autoři vyjadřovali názor z obou stran, nebo naopak většinou žádný postoj nezaujímali. Ti byli vyřazení. Příspěvky ostatních vybraných autorů posloužily jako zdroj dat. Všechny příspěvky od jednoho autora byly ohodnoceny jako PRO nebo PROTI podle toho, který názor převažoval v ručně ohodnocených datech.

Do výsledného poloautomatického korpusu se zařadily i ručně anotované příspěvky od vybraných autorů. Výsledný korpus měl 714 PRO, 1801 PROTI a 129 NIC příspěvků. Ruční anotace namátkově vybraných příspěvků z těchto dat ale ukázala, že ve skutečnosti je zhruba 60% těchto příspěvků nesouvisející nebo neutrální. Všechny poloautomatické příspěvky jsou k tématu „Zákaz kouření v restauracích“. Porovnání úspěšnosti viz kapitola 8 Testování a výsledky.

6 dostupné z http://ufal.mff.cuni.cz/udpipe

7 popis formátu, jednotlivé informace a jejich možné hodnoty jsou vyjmenované na stránce CoNLL-U Format na webu Universal Dependencies (http://universaldependencies.org/format.html)

5 Klasifikace

Klasifikace se provádí pomocí učení s učitelem. Pro klasifikaci se používají jednak příznaky, implementované v rámci této diplomové práce, a jednak vlastní algoritmus strojového učení (v tomto případě klasifikace). Ten je implementovaný v rámci knihovny Brainy (Konkol 2014). Příznaky slouží k reprezentaci dat tak, aby byla vhodná pro klasifikační algoritmus. Popis jednotlivých příznaků uvádím níže.

Knihovna Brainy umožňuje zvolit, který klasifikátor se pro klasifikaci použije:

- Naivní Bayesovský
- SVM (Support vector machines, podpůrné vektory)
- Maximum Entropy (maximální entropie)

Tyto klasifikátory jsou implementované v rámci Brainy a v předchozích výzkumech dosahovaly dobrých výsledků. Klasifikátory SVM a Maximum Entropy jsem porovnala, výsledky viz kapitola 8.2 Varianty výpočtu.

5.1 Dvouúrovňová klasifikace

V první verzi se používala jednoúrovňová klasifikace. To znamená, že rovnou byly zadány tři třídy (PRO, PROTI, NIC), do kterých klasifikátor komentáře zařazuje. Klasifikace všech dat proběhla v jednom průchodu. Při dalších pokusech byla vytvořena druhá verze, která klasifikuje dvěma průchody. V prvním průchodu se klasifikuje pouze na STANCE (komentář vyjadřuje postoj) oproti NIC (komentář žádný postoj neobsahuje). V druhém průchodu se použijí jen ty příspěvky, které v prvním průchodu byly hodnocené jako STANCE. Ty se rozdělují na PRO a PROTI.

5.2 Feature

Data jsou reprezentována pomocí příznaků. Každý příznak je implementován jako jedna třída implementující rozhraní Feature. Proto pro skupinu implementovaných příznaků používám slovo Feature.

Každá třída Feature má minimálně tři důležité metody. První je konstruktor, ve kterém se inicializují důležité parametry, které potom ovlivňují chování příznaku. V některých případech se v něm také nastavují obecná data, která se používají při klasifikaci, ale nedají se zjistit z trénovacích dat. Příkladem může být seznam emotikonů, který je pevně daný a nemá smysl ho určovat z trénovacích dat. Dalším příkladem jsou slovníkové třídy Feature, protože slovník je také předem daný. V konstruktoru se nahrává z textového souboru.

Druhá je metoda train(), pomocí ní se feature trénuje. Třída Feature je vlastně šablona – feature template. Šablona určuje, jakým způsobem se budou příznaky z komentářů extrahovat, co je ten důležitý příznak. Teprve při trénování se určují jednotlivé konkrétní příznaky, jednotlivé hodnoty. Například, u WordFeature je šablona slovo a při trénování se určí, která konkrétní slova budou sloužit jako feature, například „souhlasím“ nebo „pravda“.

Třetí metoda, extract(), extrahuje jednotlivé konkrétní příznaky z konkrétního komentáře. To znamená, že vytváří vektor, který je částí matematické reprezentace daného komentáře. Jedna třída Feature (neboli jedna šablona) vytváří jeden vektor. Jeho délka je rovná počtu jednotlivých konkrétních příznaků, které byly v rámci třídy natrénované. Celá reprezentace objektu je složená z vektorů, které byly extrahovány ve všech třídách Feature. Extrakce příznaků probíhá tak, že se projdou všechny konkrétní příznaky, a pro každou se do vektoru nastaví jednička nebo nula, podle toho, jestli je daný znak (slovo, bigram, emotikon) v komentáři zastoupen.

Jednotlivé třídy Feature jsou popsané v kapitole 7 Feature. Výsledky testování různých Feature jsou v kapitole 8 Testování a výsledky.

5.3 Křížová validace

Dat v korpusu není mnoho (na poměry automatické klasifikace). Proto se na výsledcích může projevit heterogenita dat. Například, pokud bychom data pouze rozdělili na testovací a trénovací část, tak by se mohlo stát, že v testovacích datech bude většina příspěvků od jednoho uživatele, který se vyjadřuje jinak než všichni ostatní. Klasifikátor by potom měl špatné výsledky. Proto používám křížovou validaci a to 5-fold validaci. Data se před klasifikací automaticky rozdělí na trénovací a testovací část, kde testovací je 20% z celkového počtu. Rozdělení dat a klasifikace probíhají pětkrát, tak aby testovací data byla pokaždé jiná pětina z korpusu.

5.4 Úspěšnost

Pro ověření úspěšnosti metody se používá F1 skóre. To se vypočítává jednak pro každou třídu (PRO, PROTI, NIC), jednak průměr mezi PRO a PROTI a jednak průměr přes všechny tři třídy. Pro výsledné rozhodnutí o úspěšnějším prvku se používá F1 skóre počítané jako průměr PRO a PROTI.

Pro kontrolu a pro podrobnější zkoumání se počítají také další metriky, a to:

- počet správně/špatně zařazených příspěvků
- true positive, true negative, false positive, false negative pro všechny tři třídy
- recall a precision pro všechny třídy

Každá z uvedených metrik se spočte pro každou iteraci při křížové validaci. Pro každou metriku se také počítá průměr přes všechny iterace křížové validace. Průměrné F1 skóre se počítá jako průměr F1 skóre pro jednotlivé iterace. Nepoužívá se pro jeho výpočet průměrný recall ani precision.

5.4.1 Dvouúrovňová klasifikace

Při dvouúrovňové klasifikaci probíhá klasifikace ve skutečnosti dvakrát. Výsledné F skóre používané pro porovnání se počítá až na celkových datech, po proběhnutí obou úrovní. Ale pro kontrolu se počítá také F1 skóre pro jednotlivé úrovně.

V první úrovni se rozděluje jen na NIC a STANCE. Pro každou z těchto tříd se vypočítá a vypíše true/false positive/negative, precision, recall a F1 skóre. Celkové F1 skóre první úrovně je průměr $F1_{NIC}$ a $F1_{STANCE}$.

Ve druhé úrovni se rozděluje jen na PRO a PROTI. Opět se pro každou třídu spočítají a vypíšou metriky. Celkové F1 skóre druhé úrovně je průměr $F1_{PRO}$ a $F1_{PROTI}$ Zde je ale jedna zvláštnost. Do druhé úrovně se mohly dostat příspěvky, které mají skutečný postoj NIC. Ať už je tedy klasifikátor druhé úrovně zařadí do kterékoliv třídy, vždy to bude špatně. Tyto příspěvky totiž špatně zařadila už první úroveň. Jsou započítané do (ne)úspěšnosti první úrovně. Ve druhé úrovni se už do hodnocení nezařazují, při výpočtu metrik druhé úrovně se úplně ignorují.

6 Implementace

V této části popíšu implementační detaily a programy, které jsem vytvořila v rámci diplomové práce.

6.1 Korpus dat

Pro stahování dat byl vytvořen PHP program. Jak zmiňuji výše, program stahuje komentáře přímo ze stránek diskuze, a jednotlivé texty získá zpracováním HTML kódu. Program vždy stáhne jednu stránku, získá z ní texty komentářů, a potom pokračuje další stránkou. Pro pomoc s HTML je použité rozšíření PHP DOM (Document Object Model). To zjednodušuje práci s HTML a umožňuje lépe procházet jednotlivé elementy a přistupovat k nim.

V rámci HTML se prochází struktura, jednotlivé elementy a jejich potomky. Procházené elementy se filtrují pomocí názvu elementu (tag), atributů ID a class. Tím se vždy nalezne jeden komentář, jeho text se uloží, a potom se hledá další komentář. Knihovna Httpful umožňuje získat i vnitřní text elementu. To se využilo, protože nalezený div s komentářem mohl obsahovat ještě nějaké další elementy (například odstavce <p>). Pokud už ale vnitřní elementy obsahovaly jen text, tak rozšíření DOM umožnilo rovnou získat text, bez dalšího procházení vnitřních elementů. Ukázka procházení elementů viz Obrázek 6.1.

```
   //tabulkový layout příspěvku
   $table = $prispevek->getElementsByTagName('table')->item(0);
   $table_row = $table->getElementsByTagName('tr')->item(0);
   $td = $table_row->getElementsByTagName('td')->item(1);

   //hledame <div> s třídou 'user-text' - ten obsahuje text
vložený uživatelem
   $user_text = false;
   foreach ($td->childNodes as $div) {
           if ($div->nodeType != XML_ELEMENT_NODE)
                   continue;
           if ($div->getAttribute( 'class' ) == 'user-text') {
                   $user_text = $div;//nalezen
           }
```

Obrázek 6.1: Ukázka procházení HTML struktury a hledání textu komentáře.

Složitěji se získávají emotikony. Na iDNES.cz se emotikony zobrazují jako obrázek, a proto nestačí získat vnitřní text (obrázek žádný vnitřní text nemá). Proto se musí v komentáři najít všechny obrázky, a místo těch se na správnou pozici v textu vloží jejich atribut „alt" (alternativní text obrázku). Obrázek 6.2 ukazuje získávání textu k emotikonům.

```
foreach ($paragraph->childNodes as $child) {
   if ($child->nodeType == XML_TEXT_NODE) {
      //připojím text této části odstavce k příspěvku
      $comment_text .= trim($child->textContent);
   } else {
      if ($child->tagName == 'img'
           && $child->getAttribute('class') == 'smiley' ) {
         //připojíme k příspěvku alt obrázku
         $comment_text .= trim($child->getAttribute('alt'));
      }
   }
}
```

Obrázek 6.2: Ukázka sestavování textu komentáře z textu odstavců a atributu „alt“ obrázků.

V programu je možné nastavit následující parametry:

- url adresa diskuze, ze které se komentáře stahují,
- stránky od-do,
- ID, od kterého se budou komentáře číslovat.
- Podle toho, jaký výstup budeme používat, se volí také:
  - název souboru (v případě, že chceme vypsat všechny komentáře do jednoho souboru),
  - název tabulky v databázi, do které se komentáře ukládají.

Parametry se nastavují v konfiguračním souboru config.php.

Jedna z verzí programu umožňovala zadat uživatele, jejichž příspěvky se rovnou automaticky anotovaly jako PRO nebo PROTI. Později se ale ukázalo, že anotovat uživatele, dokud nevíme celkový počet jejich příspěvků, je neefektivní. Byl zvolen výhodnější postup, kdy se uživatelé hodnotí až po stažení všech příspěvků.

Pro spojení s databází slouží pomocná třída dbConnection, která inicializuje připojení.

Program umožňuje 4 výstupy.

1. Výpis na standardní rozhraní. Toto slouží pouze k testovacím účelům.
2. Všechny komentáře do jednoho souboru. Každý komentář je na vlastním řádku. Struktura řádku je „<ID> <TÉMA> <TEXT_KOMENTÁŘE>". Toto je stejný formát, který se používal v SemEval.
3. Každý komentář do vlastního souboru. Název souboru je ID komentáře. Toto se dále používá pro zpracování pomocí knihovny UDPipe. V jejím výstupu je totiž na každém řádku jedno slovo, takže komentáře musí být odlišené takto.
4. Do databáze. Struktura tabulky je následující: id, theme, text, url. Url se ukládá proto, aby se později dalo zjistit, ze které diskuze byl komentář získán. ID komentáře se nastavuje automaticky v databázi. Program ukládá pouze téma, text a url.

Ruční anotace postoje se obvykle provádí v souhrnném textovém souboru (bod č. 2.). Klasifikační program ale využívá dále upravené soubory z bodu 3. Postoj komentářů zjišťuje z jednoduchého textového souboru, kde je na jednom řádku vždy ID a postoj daného komentáře. Tento soubor s názory lze vytvořit ze souhrnného textového souboru pomocí textových úprav s regulárním výrazem.

Pro sémantické a syntaktické zpracování komentářů byl testován dvojí různý software třetích stran. První varianta byla kombinace následujících dvou programů:

- Majka – rychlý morfologický analyzátor (Šmerk, Rychlý, 2009)
- SET (Syntactic Engineering Tool) (Kovář a kol., 2011)

Majka je morfologický analyzátor, který se dá použít jako knihovna nebo řádková program. Slouží k určení základního tvaru a gramatických značek (slovní druh,rod,…). Pokud je možné, aby zadaný tvar slova měl různé gramatické značky, určí všechny (například slovo „stát“ může znamenat „státní instituce“ v prvním pádě (např. „Stát zakázal kouření."), nebo sloveso „stojí“ v infinitivu (např. „budou tam stát kuřáci"). SET provádí syntaktickou analýzu, určuje závislosti slov ve větě a je schopen rozpoznat větný strom.

Druhá varianta je knihovna UDPipe (Straka a kol. 2016). Ta byla nakonec použita, protože umožňuje provést zároveň syntaktickou i morfologickou analýzu v jednom příkazu, bez mezikroků. Použití Majky a SETu by také vyžadovalo použití ještě třetího programu, který by rozhodoval, která z variant určených Majkou je ve větě správná.

Knihovna UDPipe se spouští příkazem:

```
   udpipe.exe --outfile=data/{}-out.txt --tokenize --tag --parse
czech-ud-1.2-160523.udpipe 2.txt 4.txt 10.txt 1000.txt 1003.txt
```

Prvním parametrem je název výstupního souboru, parametry --tokenize, --tag a --parse určují, že se má provést rozdělení věty na tokeny (slova), určení slovních druhů a větných závislostí. Parametr „czech-ud-1.2-160523.udpipe" je model češtiny použitý pro zpracování[8]. Zbylé parametry jsou názvy souborů, které se mají zpracovat. Obrázek 6.3 ukazuje větu, zpracovanou nástrojem UDPipe a zobrazenou ve formátu ConLL-U.

```
1     Já    já    PRON  PP-S1--1------- Case=Nom|Number=Sing|
Person=1|PronType=Prs 3     nsubj _     _
2     opravdu     opravdu     ADV   Db------------- _     3
      advmod      _     _
3     začnu začít VERB  VB-S---1P-AA---  Mood=Ind|Negative=Pos|
Number=Sing|Person=1|Tense=Pres|VerbForm=Fin|Voice=Act 0     root
      _     _
4     kouřit      kouřit      VERB  Vf--------A----  Aspect=Imp|
Negative=Pos|VerbForm=Inf   3     xcomp _     SpaceAfter=No
5     !     !     PUNCT Z:------------- _     3     punct _     _
```

Obrázek 6.3: Ukázka věty ve formátu ConLL-U

Ke spuštění programu je třeba PHP server (Apache). Pokud chceme ukládat příspěvky do databáze, tak je třeba také MySQL databáze, kde je již vytvořená tabulka s požadovanou strukturou. Přístupové údaje k databázi se nastavují v souboru dbConnection.php.

6.2 Klasifikace

Po načtení dat začne vlastní klasifikace. Ta je popsaná v následující části.

Navržená metoda byla v rámci diplomové práce implementována programem v jazyce Java. Program načte příspěvky z korpusu dat, klasifikuje je, a pomocí zvolených metrik vypočte úspěšnost. Skládá se z částí:

- pro ukládání a operace s vlastními komentáři,
- pro načítání a správu dat,
- pro klasifikaci,

8 model je možné stáhnout spolu s knihovnou UDPipe ze stránek https://ufal.mff.cuni.cz/udpipe#language_models

- pro výpočet a zobrazení výsledků.
- Poslední, celkem obsáhlou částí jsou feature, kterých je větší množství.

Zdrojové kódy jsou komentované.

Téma se do klasifikátoru nikde explicitně nezadává, není to třeba. Pokud by to bylo potřeba, tak by se dal program upravit tak, aby se téma načítalo z komentáře, protože u každého komentáře je téma uvedené. Podle tématu jsou ale rozdělená data do složek, a programu se při spuštění zadává jako parametr konkrétní složka. Tím se dá testovat každé téma zvlášť, pokud jsou rozdělená do složek, nebo i obě témata dohromady, pokud jsou všechny komentáře v jedné složce.

6.2.1 Práce s daty

Pro přechovávání komentářů slouží třídy `Word`, `Sentence` a `Comment`. `Comment` obsahuje ID komentáře a seznam vět (`Sentence`). `Sentence` obsahuje seznam slov (`Word`). Třída `Word` uchovává údaje o slově, jako je slovo samotné, lemma, slovní druh a vztah ve větě. Dále existuje třída `CommentsOpinionsList`, která uchovává načtený seznam komentářů, seznam skutečných postojů k těmto komentářům (pokud jsou známé) a seznam postojů určených klasifikátorem. Umožňuje operace s nimi, jako je přidávání komentářů a vrácení celého seznamu (komentáře + postoje) nebo jeho části. Vrácení části seznamu se používá při křížové validaci, kdy je potřeba pouze určená část dat.

Seznam postojů (ať už skutečných, nebo klasifikovaných) je `List<Integer>`. Každé číslo reprezentuje jednu třídu. V případě této diplomové práce jsou označení následující: 0=NIC, 1=PRO, 2=PROTI. Kvůli knihovně Brainy je nutné dodržet omezení, že třídy musí být číslované od nuly a nepřeskakovat čísla.

Program pracuje s daty, která byla zpracována pomocí nástroje UDPipe. Klasifikační program načítá data ze složky, kde je každý komentář v samostatném souboru. Název souboru je ve tvaru „<id>.txt", kde <id> je ID komentáře. V souboru je na jednom řádku slovo a další informace k němu. Kromě souborů s komentáři existuje jeden zvláštní soubor, ve kterém jsou uvedené postoje komentářů. Tento soubor má každý řádek ve tvaru „<id> <postoj>".

Načítání dat z textového souboru zařizuje třída `ClassificationData`. Prochází všechny soubory v zadané složce. V souboru čte řádky, a každý řádek pomocí regulárního výrazu rozdělí na slovo, a informace o něm (lemma, slovní druh...). Údaje z každého souboru uloží do třídy `Comment`. Třída `ClassificationData` vlastní jeden `CommentOpinionList`, do kterého zpracované komentáře ukládá. `ClassificationData` také umožňuje ze seznamu komentářů extrahovat testovací

data (jedna část z K pro K-fold křížové ověření) a trénovací data (zbylé části dat spojené do jednoho seznamu).

Pro jednoúrovňovou klasifikaci se data dají použít v té podobě, v jaké je poskytuje `ClassificationData`. `SupervisedClassifierTrainer` bere jako parametry dva seznamy - trénovací data `List<Comment>`, a seznam postojů (`List<Integer>`), které se dají snadno získat z `CommentOpinionList`, který nám poskytuje `ClassificationData`.

Pro dvouúrovňovou klasifikaci se ale data musí předem připravit. V první úrovni klasifikujeme pouze na STANCE a NIC. Proto je potřeba v trénovacích datech upravit seznam postojů, tak aby obsahoval jen třídy NIC a STANCE, místo NIC, PRO a PROTI. Toho se dosáhne tak, že se všechny postoje 2=PROTI přečíslují na jedna. Tabulka 6.1 ukazuje převod tříd.

Původní třída	Původní číslo	Nové číslo	Nová třída
NIC	0	0	NIC
PRO	1	1	STANCE
PROTI	2	1	STANCE

Tabulka 6.1 Převod indexů tříd pro dvouúrovňovou klasifikaci

Do druhé úrovně už potřebujeme předat jen ty příspěvky, které jsou označené jako STANCE. Proto musíme testovací data, která jsou výsledkem první úrovně, rozdělit na dvě části. Pro klasifikaci použijeme jen část označenou jako STANCE.

Trénovací data se také musí upravit. Jednak se z nich musí odstranit příspěvky třídy NIC (protože ta se ve druhé úrovni neklasifikuje). A kvůli odstranění první třídy musíme přečíslovat zbylé dvě třídy, aby číslování začínalo od nuly. Převod uvádí Tabulka 6.2

Původní třída	Původní číslo	Nové číslo	Nová třída
NIC	0	--	--
PRO	1	0	PRO
PROTI	2	1	PROTI

Tabulka 6.2 Převod indexů tříd pro druhou úroveň dvouúrovňové klasifikace

Po dokončení druhé úrovně se data z první i druhé úrovně sjednotí do výsledných dat.

6.2.2 Vlastní klasifikace

Pro klasifikaci se používá několik tříd z Brainy. Mimo jiné `SupervisedClassifierTrainer`, `Classifier`, `Classification-`

`Results` a `FeatureSet`. Klasifikátor `Classifier` se nejprve natrénuje na trénovacích datech pomocí `SupervisedClassifierTrainer`. Výsledky klasifikace jsou vrácené jako `ClassificationResults`. `FeatureSet` je třeba více popsat. `FeatureSet<T>` je generická třída implementovaná v rámci Brainy. V tomto případě je typem T mnou vytvořený `Comment`, takže používám `FeatureSet<Comment>`. Tento seznam příznaků spravuje jednotlivé třídy `Feature`. Každá vlastní třída, přidaná do setu, musí implementovat generické rozhraní `Feature<T>`, kde T je stejný typ, jako u `FeatureSet<T>`. Všechny feature vytvořené v rámci této diplomové práce implementují `Feature<Comment>`. To znamená, že vytvářejí reprezentaci komentáře (`Comment`).

Zde, pod pojmem Feature (samotné rozhraní nebo libovolná třída, implementující rozhraní Feature) rozumíme feature template. To znamená, že tím myslíme množinu všech možných hodnot, nejen jednu konkrétní.

Feature musí implementovat dvě metody: `extractFeature()` a `train()`. V metodě `train()` se projdou všechny komentáře, a zjistí se v nich znaky, specifické pro danou `Feature`. Například u příznaku „první slovo" se vytvoří seznam všech slov, která jsou první v testovacích komentářích. Některé `Feature` by z trénovacích dat nezjistily žádné informace navíc, a proto mají metodu `train()` prázdnou. Například `EmoticonFeature` je závislá pouze na seznamu emotikonů, který je pevně daný pravidly na iDNES.cz a je nezávislý na trénovacích datech.

V metodě `extractFeature()` se potom určuje číselná reprezentace pro přímo jeden daný komentář z testovacích dat. Reprezentace se určuje podle informací zjištěných v metodě `train()`. Každá `Feature` je určená číselným vektorem. Každý komentář pro každou `Feature` je určen číselným vektorem jedniček a nul. Jedničky jsou na místech, kde komentář obsahuje natrénovanou informaci. Například u příznaku „první slovo" je délka vektoru rovna počtu všech odlišných prvních slov z trénovacích komentářů. Jednička je na místě toho slova, kterým začíná právě testovaný komentář. Reprezentaci viz Obrázek 6.4.

```
1.
13530 Daně z cigaret brát, to jo, ale dovolit kouřit, to ne.
Trapas.
13531 Nikdo nezakazuje kouření. Jen si nezapálíte všude, kde se
vám zlíbí.
10867 Kouření rozhodně není o svobodě. Kouření je naopak o
otroctví cigaretě.

2.
[Daně, Nikdo, Kouření]
[0   , 1    , 2      ]

3.
13418 Kouření ve školách, na úřadech atd. je již zakázáno.
[0, 0, 1]
```

Obrázek 6.4: Příklad reprezentace komentáře pomocí jednoho příznaku „První slovo ve větě). 1. trénovací data, 2. seznam známých prvních slov ve větě, 3.testovací komentář a jeho reprezentace pomocí vektoru. Jednička na indexu 2 znamená, že věta začíná známým slovem „Kouření", které je v seznamu známých slov uvedené na pozici 2

Pro ladicí účely většina tříd implementuje metodu `toString()`. Třídy `Feature` v ní vypisují název třídy a případné důležité parametry (například slovník u slovníkových příznaků).

6.3 Křížová validace

Pro K-fold křížovou validaci se používá cyklus, který proběhne K-krát. Rozdělení dat na trénovací a testovací zajišťuje už třída `ClassificationData`. Postupně se prochází části dat od 1 do K. Daná část se vždy vyčlení pro testování. Na zbytku dat se natrénuje klasifikátor, a na vyčleněné části se otestuje jeho úspěšnost. Pro všechna opakování křížové validace se používá stejná konfigurace klasifikátoru – stejné `Feature`, jedno/dvouúrovňová klasifikace, klasifikátor a podobně. Na konci každého cyklu křížové validace se výsledky z daného cyklu uloží do instance třídy `Statistics`. Po doběhnutí křížové validace se výsledky zprůměrují a vypočítá se z nich výsledné F-skóre pro danou konfiguraci.

6.4 Hill climbing

Pro výběr příznaků se používá horolezecký algoritmus. Ten je implementován jako dva vnořené cykly. Nejprve se vytvoří dva seznamy příznaků: `selectedFeatures` a `availableFeatures`. Seznam `SelectedFeatures` obsahuje počáteční konfiguraci. Může to být buď prázdný seznam, nebo nějaká sada příznaků, například taková, která měla dobré výsledky v předchozích testech. Po doběhnutí horolezeckého algoritmu bude seznam `selectedFeatures` obsahovat nejlepší konfiguraci. `AvailableFeatures` obsahuje všechny příznaky, které chceme otestovat.

Potom probíhá vlastní horolezecký algoritmus. Vnitřní cyklus vezme jako základ `selectedFeatures`. Přidá do seznamu vždy jednu Feature z `availableFeatures` a otestuje klasifikaci. V každém cyklu použije všechny příznaky z `selectedFeatures`, a přidá k nim jednu z `availableFeatures`. Nakonec vybere tu `Feature`, jejíž přidání zlepšilo výsledek nejvíce. Tu přesune z `availableFeatures` do `selectedFeatures`. Vnější cyklus probíhá, dokud se výsledky vnitřního cyklu zlepšují.

Výstupem horolezeckého algoritmu je jednak sada příznaků, která měla nejlepší výsledky, jednak tento nejlepší výsledek, a jednak zlepšení, o kolik každá jednotlivá `Feature` zlepšila výsledek.

6.5 Výsledky

Pro měření úspěšnosti program vypočítává různé statistiky. Jejich seznam je uvedený v kapitole 5.4. Pro jejich ukládání a základní operace s nimi slouží třída `Statistics`. Pro základní hodnoty (true/false positive/negative) má třída proměnné. Před výpočtem dalších statistik se projdou všechny klasifikované příspěvky, a podle jejich skutečného a klasifikovaného postoje se do instance třídy `Statistics` nastaví základní údaje. Z nich se potom vypočítávají další statistiky. Pro generické použití má třída „rozcestníkovou“ metodu `getMeasure()`. Ta přijímá jako parametr řetězec, který rozdělí příkazem switch a tím identifikuje požadovanou metriku. Druhý parametr určuje, pro kterou z klasifikačních tříd (PRO, PROTI, NIC) metriku požadujeme. Podle parametrů vrátí jednu z proměnných třídy, respektive vrátí výsledek jedné z metod.

Pro každý běh klasifikace se rovnou vypočítají statistiky (uložené v jedné instanci třídy `Statistics`). Během křížového ověření se statistiky jen ukládají do seznamu. Po doběhnutí všech kroků křížové validace se výsledky projdou a vypíšou, aby bylo možné jednotlivé kroky snadno porovnávat. Pro průchod výslednými klasifikovanými komentáři slouží konstruktor `Statistics()`, kterému se předává výsledek klasifikace. Výpočet průměrů přes jednotlivé kroky, a výpis výsledků zajišťují statické metody `printStatisticsList()`, `printAllAndMean()`. Desetinná čísla se vypisují s přesností na 4 místa. Obrázek 6.5 ukazuje výstup z jednoho běhu programu.

STATISTIKY /pokus:	0	1	2	3	4	průměr
spravny odhad :	42	46	47	51	53	47
spatny odhad :	44	40	40	35	33	38
FAVOR TP :	7	11	13	16	15	12
FAVOR FP :	22	24	9	15	10	16
FAVOR FN :	2	7	15	12	18	10
AGAINST TP :	7	12	11	12	9	10
AGAINST FP :	20	6	12	14	11	12
AGAINST FN :	10	16	17	11	10	12
NONE TP :	28	23	23	23	29	25
NONE FP :	2	10	19	6	12	9
NONE FN :	32	17	8	12	5	14
FAVOR precision :	0,2414	0,3143	0,5909	0,5161	0,6000	0,4525
FAVOR recall :	0,7778	0,6111	0,4643	0,5714	0,4545	0,5758
AGAINST precision:	0,2593	0,6667	0,4783	0,4615	0,4500	0,4631
AGAINST recall :	0,4118	0,4286	0,3929	0,5217	0,4737	0,4457
NONE precision :	0,9333	0,6970	0,5476	0,7931	0,7073	0,7357
NONE recall :	0,4667	0,5750	0,7419	0,6571	0,8529	0,6587
FAVOR F1 score :	0,1842	0,2075	0,2600	0,2712	0,2586	0,2363
AGAINST F1 score :	0,1591	0,2609	0,2157	0,2449	0,2308	0,2223
AVERAGE F1 score :	0,1717	0,2342	0,2378	0,2580	0,2447	0,2293

Obrázek 6.5: Ukázka statistik, které vypisuje program.

7 Feature

Zde jsou popsané jednotlivé příznaky implementované jako třídy Feature. U jednotlivých tříd se popisuje, jak se tvoří vektor příznaků, jestli se trénují, a v některých případech i důvod vytvoření konkrétního příznaku. Jednotlivé příznaky jsou rozdělené do skupin podle podobnosti, každá skupina je implementována v samostatném balíku (package).

7.1 Základní příznaky

Jsou v balíku „features“.

7.1.1 WordFeature

Základní třída je WordFeature. Jako jediná byla použitá v programu baseline pro úkol v SemEval. Délka vzniklého vektoru je počet všech známých slov v korpusu (řádově tisíce). Má jeden parametr – threshold – který určuje, kolikrát se musí slovo v trénovacích datech vyskytovat, aby bylo použito jako příznak. Délka vektoru tedy nemusí být vždy počet všech slov v datech, může to být jen počet častých slov v datech, v závislosti na threshold. Výsledná reprezentace komentáře neudržuje pořadí slov. To znamená, že například komentář „nevlastní hospodu“ má stejnou reprezentaci jako „hospodu nevlastní“. Při trénování příznaku se projdou všechna slova ve všech komentářích a vytvoří se interní seznam známých slov. Při extrakci se potom zjišťuje, která známá slova komentář obsahuje.

7.1.2 WordNoDiaFeature

WordNoDiaFeature komentář reprezentuje také pomocí slov, ale ze slov byla odstraněna diakritika. V datech se často vyskytují komentáře, kde autor nepoužívá diakritiku. Tím se ale stane, že jedno slovo je v datech někdy s diakritikou a někdy bez ní (např. „kouření" a „koureni"). Pro WordFeature to znamená dvě různá slova. WordNoDiaFeature se snaží tento nedostatek odstranit.

7.1.3 WordStemFeature

Tento příznak používá nástroj HPS (High Precision Stemmer, Brychcín a Konopík, 2015). Ten ze slova ořízne počáteční a koncové znaky tak, aby se vytvořil základní tvar. WordStemFeature funguje podobně jako LemmaFeature (viz dále), ale základní tvar se získává jiným způsobem.

7.1.4 OneFeature

Toto je příznak použitý pro doplnění vektoru. Netrénuje se a vždy má délku jedna. Hodnota, která se ve vektoru nastavuje, se zadává jako parametr v konstruktoru. V některých případech tento příznak zlepšuje výsledky i přesto, že nezávisí na datech.

7.1.5 BigramFeature

BigramFeature je podobná, jako WordFeature, ale používá dvojice slov. Při trénování vytvoří seznam známých dvojic slov – každá dvojice slov je jeden konkrétní příznak. Při extrakci příznaků v právě testovaném komentáři hledá známé dvojice slov. Těmi je potom komentář reprezentován. BigramFeature má také parametr threshold, který určuje, kolikrát se dvojice slov musí v trénovacích datech vyskytnout, aby se zahrnula do známých dvojic.

Tato Feature má za cíl naučit se fráze, které nemají jen jedno slovo. Například fráze „odnaučený kuřák“, která má přibližně opačný význam, než samotné slovo „kuřák“.

7.1.6 BigramBagFeature

Tato Feature je podobná, jako BigramFeature. Také trénuje dvojice slov, a reprezentuje pomocí nich komentář. Také má parametr threshold. Ale zde nezáleží na pořadí slov ve dvojici – používá se tzv. princip Bag-of-words. To může být výhodné v češtině, která má celkem volný slovosled. Například věty „Hospodský čepuje." a „Čepuje hospodský." mají stejnou reprezentaci. To odpovídá skutečnému stavu, protože obě tyto věty mají zhruba stejný význam.

7.1.7 UppercaseFeature, UppercaseHalfFeature

Toto jsou binární příznaky. To znamená, že může mít jen dvě různé hodnoty, délka vektoru je 1. Komentář je reprezentován jen jako 1 nebo 0.

Pokud komentář obsahuje alespoň jedno slovo velkými písmeny, tak se příznak u UppercaseFeature nastaví na hodnotu 1, jinak 0. Tato Feature se netrénuje. Slova psaná velkými písmeny obvykle znamenají silné emoce, a tím pravděpodobně i silný postoj.

UppercaseHalfFeature je analogická jako UppercaseFeature. Liší se v tom, že příznak je nastaven na 1 až v případě, že slov velkými písmeny je více než polovina komentáře. To by mělo znamenat ještě silnější sentiment, ale také se takovéto komentáře méně vyskytují v datech.

7.1.8 NthWordFeature

Tento reprezentuje komentář pomocí jednoho slova. A to slova, které je na N-té pozici. N se zadává v konstruktoru. V metodě train() vytváří seznam všech známých slov, které jsou na N-té pozici. Ten potom určuje výslednou reprezentaci komentáře v metodě extractFeature(). Tato Feature je použitá pouze s N=1, to znamená s prvním slovem věty.

7.2 Syntaktické příznaky

Tato skupina používá údaje, které poskytuje UDPipe. Výzkum (Gamon 2004) ukázal, že hlubší informace o textu mohou vylepšit výsledky klasifikátoru, proto s nimi experimentuji i v rámci této diplomové práce. Jsou v balíku featuresSynactic.

7.2.1 DependencyRelationFeature

Tato Feature příspěvek reprezentuje jako množinu větných členů (nebo vztahů ve větě. Feature se netrénuje, ale v konstruktoru se ze slovníku načtou existující závislosti ve větě. V diplomové práci používám závislosti používané nástrojem UDPipe. Seznam závislostí a je daný ConLL-U formátem (Universal Dependencies 2016). Často je jako závislost uveden větný člen, ale většinou ještě s nějakou dodatečnou informací. Například, jestli je to podmět aktivní (např. ve větě „Kouření škodí zdraví“), nebo pasivní (např. ve větě „Kouření bude zakázáno“). Větnou závislostí nemusí být jen větné členy, můžou to být jiné vztahy. Například vztah „root“ (kořen věty) může, ale nemusí být přísudek.

7.2.2 LemmaFeature

LemmaFeature používá všechna slova z komentáře, ale v základní podobě. Základní tvar znamená například první pád u podstatných jmen, infinitiv u sloves a pod. Tím se redukuje rozměr vektoru (neboli počet všech možných kombinací), protože různá slova mají stejné lemma. Při trénování se vytvoří seznam všech základních tvarů. Při extrakci se nastavuje jednička u všech základních tvarů ze seznamu, které se v příspěvku vyskytují.

7.2.3 Predicate-/Subject-/ObjectFeature

Tyto Feature z komentáře používají pouze jeden větný člen (přísudek/podmět/předmět). Předmětů ve větě může být víc, v takovém případě jsou zahrnuté všechny. Funguje podobně, jako WordFeature, ale pouze pro jeden větný člen. Při trénování se zjistí, jaká všechna slova se vyskytují jako daný větný člen. Při extrakci příznaku se nastaví ve vektoru jednička u těch slov ze seznamu, které se vyskytují v příspěvku jako daný větný člen.

7.2.4 SubjectPredicateFeature

V této Feature komentář je reprezentován jako dvojice podmět-přísudek. Při trénování vytvoří dva seznamy - všechny známé přísudky a všechny známé podměty. Při extrakci nastavuje jedničku na pozici podmětu a přísudku.

7.2.5 PartOfSpeechFeature

Toto je příznak větného členu. Funguje podobně, jako DependencyRelationFeature, ale místo slovních druhů používá větné členy. Při trénování vytvoří seznam známých slovních druhů, a potom podle nich reprezentuje komentář.

7.2.6 SentenceLengthFeature

Tento příznak reprezentuje komentář podle délky věty. Ve třídě SentenceLengthFeature jsou nastavené intervaly. Počet intervalů určuje délku výsledného vektoru. Metoda se netrénuje. Při extrakci se vypočítá průměrná délka věty v příspěvky. Pak se zjistí interval, do kterého průměrná délka patří, a v tom se nastaví jednička.

7.3 Slovníkové příznaky

Toto jsou obvykle binární příznaky. Není třeba je trénovat, protože slovníky jsou externí. Pokud komentář obsahuje alespoň jedno slovo z daného slovníku, tak je příznak nastaven na 1. Slovníky už byly vytvořeny v předchozích výzkumech, a dávaly dobré výsledky. Proto jsem je také zahrnula do experimentů. Třída pro slovníky je jen jedna, ale lze jí zadat libovolný slovník. Název třídy je DictionaryBinFeature. Příznak je binární. Rozpoznává pouze, zda je nějaké slovo ze slovníku přítomno v komentáři. Nikoliv, které slovo to je. Použité slovníky sentimentu:

- vysoce pozitivní,
- pozitivní,
- negativní,
- vysoce negativní,
- zvyšující význam sentimentu v jiném slově
- snižující význam sentimentu v jiném slově
- negace sentimentu v jiném slově

Obrázek 7.1 ukazuje příklady slov z vysoce negativního slovníku.

```
arogan%
blbec
kritic%
lež
zkorumpovan%
zlo
```

Obrázek 7.1: Příklady slov ze slovníku s vysoce negativním sentimentem.

7.4 Příznaky, využívající interpunkci

Tyto příznaky využívají zejména slova, která knihovna UDPipe ohodnotila jako PUNCT (punctuation), neboli interpunkci. Taková slova jsou obvykle jednoznaková, a jsou to například „.“, „?“, nebo „-“.

7.4.1 EmoticonFeature

Tento příznak rozpoznává, jestli ve větě je obsažen emotikon. Emotikon je určený sekvencí znaků. Vzhledem k tomu, že dělení věty na slova provádí nástroj UDPipe, tak emotikon bývá rozdělený do dvou nebo více slov. Seznam emotikonů je načten v konstruktoru, používá se seznam všech emotikonů, které nabízí server iDNES.cz. EmoticonFeature generuje vektor o takové délce, kolik existuje různých druhů emotikonů. Každému emotikonu odpovídá jedna pozice vektoru.

7.4.2 QuotedFeature

Toto je binární příznak, který rozpoznává, zda příspěvek obsahuje uvozovky. Uvozovky jsou obvyklé v jednom ze dvou případů. Buď je v uvozovkách jedno slovo, které je míněné ironicky, nebo autor někoho cituje a v uvozovkách je delší fráze nebo věta“. Tuto Feature není třeba trénovat.

7.4.3 QuestionFeature a ImperativeFeature

Toto jsou oba binární příznaky, na rozpoznání toho, jestli se v příspěvku vyskytuje otázka nebo rozkazovací věta. Rozpoznává se podle toho, jestli věta obsahuje znak „?“ respektive „!“. Tyto Feature není třeba trénovat. QuestionFeature byla přidána proto, že komentáře, které souhlasí s tématem „Miloš Zeman“ mají často tvar otázky ve smyslu „Proč všichni kritizují Miloše Zemana?“ a podobně. Tázací i rozkazovací věty se u obou témat vyskytují celkem často.

7.5 Morfologické příznaky

Tyto třídy využívají příznaky z pole FEAT (features - morfologické vlastnosti) z CoNLL-U formátu. V tomto poli jsou uvedené vlastnosti, jako je např rod a pád podstatných jmen, čas a osoba u sloves, nebo stupeň u přídavných jmen. Žádná z těchto tříd se netrénuje, protože jsou závislé pouze na možných hodnotách jednotlivých morfologických znaků. Feature jsou v balíku featuresMorphological.

7.5.1 NegativeFeature

NegativeFeature je binární příznak, který zjišťuje, jestli je ve větě použito negativní slovo. Možné hodnoty jsou dvě: pozitivní a negativní. Negace ve větě by mohla znamenat, že autor vyjadřuje vůči něčemu postoj, není neutrální.

7.5.2 GenderFeature

GenderFeature rozpoznává rod slova. Existují tři možné hodnoty rodu: Masc (mužský), Fem (ženský), Neut (střední). Pokud je slovo s nějakým rodem nalezeno, nastaví se na dané pozici jednička. Rod se rozpoznává pouze u větných členů, které jsou předané v konstruktoru jako seznam. Pokud by se rozpoznával rod u všech slov, velmi pravděpodobně by skoro všechny příspěvky měly stejnou reprezentaci - každý příspěvek by obsahoval slova všech tří rodů. Testované kombinace byly:

- root - přísudek
- nsubj, nsubjpass, csubj - podmět
- obj, dobj, iobj - předmět
- nmod - přívlastek

7.5.3 TenseFeature

TenseFeature rozpoznává čas slova. Čas se většinou vyskytuje pouze u sloves. Rozlišujeme tři možné hodnoty: Pres (present, přítomný), Past (minulý) a Fut (future, budoucí). Feature rozpoznává čas všech slov v příspěvku. V jednom příspěvku proto může zaznamenat víc různých časů. Příklad slova, které má určený čas viz Obrázek 7.2

```
ID slovo      základ     DRUH  příznaky              vět. člen
4  ponechal   ponechat   VERB  Number=Sing|Tense=Past     root
```

Obrázek 7.2: Příklad slova s vyznačeným časem v ConLL-U formátu. Na první řádku je popisek hodnot, druhý řádek je vlastní slovo a jeho vlastnosti.

7.5.4 DegreeFeature

DegreeFeature rozpoznává stupeň slov - obvykle přídavných jmen. Má tři možné hodnoty: Pos (positive, první stupeň), Cmp (comparative, druhý stupeň), Sup (superlative, třetí stupeň). Tato feature se snaží využít skutečnost, že pokud autor komentáře něco porovnává, nebo se o něčem vyjadřuje v superlativech, tak pravděpodobně vyjadřuje názor. Příkladem mohou být příspěvky „Já volil v obou kolech Karla. Věřím, že by byl lepší prezident." nebo „Zeman ječ nejslušnější z celé té plejády rádoby demokratů"

7.5.5 FirstPersonFeature

Toto je binární feature, která rozpoznává, jestli se v textu vyskytuje slovo v první osobě. Je použitá proto, že pokud autor mluví o sobě, nebo za sebe, tak je větší pravděpodobnost, že vyjadřuje subjektivní názor. Příklad takového komentáře: „No vy tomu zákazu tleskáte a já ne."

8 Testování a výsledky

V této kapitole popíšu, jaké konfigurace programu byly testovány, a porovnám jejich výsledky. Testovány byly jednak různé množiny dat, jednak různé varianty výpočtu, a jednak různé příznaky.

8.1 Data

Zde jsou porovnané výsledky na různých množinách dat.

8.1.1 Související data

Zde porovnávám výsledky pro první a druhou verzi korpusu. V první verzi byly všechny příspěvky, včetně těch, které nesouvisely s tématem, ale vyjadřovaly postoj. Právě to pravděpodobně způsobilo horší výsledky. Klasifikátor se proto špatně natrénoval na příznaky, které vyjadřují postoj. Tabulka 8.1 ukazuje porovnání první verze korpusu (všechna data) s druhou verzí (pouze související data). Ukazuje, že vytřídění nesouvisejících dat jednoznačně zlepšilo výsledky klasifikace.

F1 skóre		Data	
Téma	Počet úrovní	Pouze související	Vše (včetně nesouvisejících)
Miloš Zeman	Jedna úroveň	**0,4774**	0,2599
	Dvě úrovně	**0,4381**	0,2838
Zákaz kouření v restauracích	Jedna úroveň	**0,5192**	0,4353
	Dvě úrovně	**0,5310**	0,4463
Průměr		**0,4808**	0,3563

Tabulka 8.1: Porovnání výsledků na všech datech oproti pouze souvisejícím datům. Použitý klasifikátor: SVM. Tučně jsou označené lepší výsledky.

8.1.2 Data se shodnou anotací dvou anotátorů

Zde je porovnání „golden" dat (těch, kde oba anotátoři určili stejný postoj) a všech dat, která byla ručně ohodnocena. Na komentářích se shodou anotátorů jsou výsledky horší, pravděpodobně kvůli menšímu počtu příspěvků. Tabulka 8.2

Otestovat klasifikaci pouze na příspěvcích se shodou šlo pouze u tématu „Zákaz kouření v restauracích", protože u tématu „Miloš Zeman" není dostatek příspěvků, které hodnotilo více anotátorů.

F1 skóre		Data	
Téma	Počet úrovní	Všechna související	Se shodnou anotací
Zákaz kouření v restauracích	Jedna úroveň	**0,5192**	0,4696
	Dvě úrovně	**0,5310**	0,4847
Průměr		**0,5251**	0,4772

Tabulka 8.2 Porovnání všech souvisejících dat, s daty kde byla shoda anotátorů.

8.1.3 Automatická data

Zde jsou výsledky klasifikace na datech, která byla získána poloautomaticky. Testování s těmito daty přineslo horší výsledky, než testování pouze s ručně anotovanými daty. Viz Tabulka 8.3. To by mohlo být způsobené jednak už zmíněnou chybou, kdy se neutrální/nesouvisející příspěvek zařadil do třídy PRO nebo PROTI. Důvodem by také mohlo být stále celkem malé množství dat, vzhledem k jejich kvalitě.

F1 skóre		Data	
Téma	Počet úrovní	Ruční anotace	Automatická anotace
Zákaz kouření v restauracích	Jedna úroveň	**0,5192**	0,4762
	Dvě úrovně	**0,5310**	0,4615
Průměr		**0,5251**	0,4688

Tabulka 8.3: Porovnání automaticky anotovaných dat s ručně anotovanými daty. Tučně jsou označené lepší výsledky.

8.2 Varianty výpočtu

Program umožňuje zvolit několik variant výpočtu. Výsledky testování jsou uvedené v této kapitole.

8.2.1 Klasifikátory

Provedla jsem několik pokusů s klasifikátory SVM a MaxEnt, které Brainy poskytuje. V pokusech se používala vždy jen základní třída Feature – WordFeature. Výsledky viz Tabulka 8.4. Klasifikátory měly celkem podobné výsledky. Proto by volba klasifikátoru zřejmě celkovou úspěšnost téměř neovlivnila. Nakonec byl zvolen SVM, který měl výsledky ve většině případů o málo lepší.

F1 skóre		Klasifikátor	
Téma	Počet úrovní	SVM	MaxEnt
Miloš Zeman	Jedna úroveň	**0,4347**	0,4323
	Dvě úrovně	0,4381	**0,4482**

Zákaz kouření v restauracích	Jedna úroveň	**0,5192**	0,5130
	Dvě úrovně	**0,5310**	0,5079
Průměr		**0,4808**	0,4753

Tabulka 8.4: Porovnání klasifikátorů SVM a Maximum Entropy. Tučně je označený lepší výsledek.

8.2.2 Počet úrovní klasifikace

Jak jsem již zmiňovala, program může klasifikovat buď do jedné ze všech tří tříd najednou (jednoúrovňově), nebo nejprve rozliší, za příspěvek obsahuje postoj, a potom teprve určí, jaký ten postoj je (dvouúrovňově). Jedna i dvě úrovně byly testovány pro obě témata, a s několika různými příznaky. Výsledky testování viz Tabulka 8.5. Tučně je označený lepší výsledek z každého řádku.

Téma	Příznaky	Jedna úroveň	Dvě úrovně
Zákaz kouření v restauracích	WordFeature	0.5192	**0.5310**
	BigramFeature	0.4429	**0.4438**
	Object	0.3780	**0.4243**
Miloš Zeman	WordFeature	0.4347	**0.4380**
	BigramFeature	**0.3841**	0.3680
	Object	0.3405	**0.3807**
Průměr		0.4166	**0.4310**

Tabulka 8.5: Porovnání jedno- a dvouúrovňové klasifikace.

8.3 Příznaky

Bylo implementováno 27 příznaků, každý z nich umožňuje různé konfigurace. Testovaly se v samostatně, v různých kombinacích (podle balíků, perspektivní příznaky, a další) i vše najednou. V tabulkách dále je porovnání výsledků klasifikace. Tabulka 8.6 uvádí porovnání vybraných základních příznaků pro obě témata.

Tabulka 8.7 uvádí porovnání na vybraných skupinách příznaků. Při testování skupin příznaků byl použit horolezecký algoritmu, který z počáteční skupiny vybere jen několik příznaků. V tabulce je uvedená jak výchozí skupina příznaků (ty, ze kterých vybíral horolezecký algoritmus), tak výsledná podmnožina příznaků s nejlepším výsledkem. V základních příznacích byla vždy dostupná WordNoDiaFeature a jeden celý balík dalších příznaků. U dvouúrovňové klasifikace jsou uvedené příznaky z druhé úrovně. V první úrovni u těchto výsledků byly vždy použité LemmaFeature a NthWordFeature. Nejlepší výsledky jsou označené tučně.

Téma →	Zákaz kouření v restauracích		Miloš Zeman	
Počet úrovní klasifikace →	Jedna	Dvě	Jedna	Dvě
Příznaky				
WordFeature	0,5171	0,5323	0,4774	0,4744
WordNoDiaFeature	0,5317	0,5470	**0,5036**	0,4907
WordStemFeature	0,5374	0,5415	0,4860	**0,4950**
LemmaFeature	**0,5425**	**0,5490**	0,4897	0,4866

Tabulka 8.6: Porovnání výsledků se základními příznaky. Tučně jsou označené nejlepší výsledky pro každý sloupec.

Zákaz kouření v restauracích			
	Počet úrovní klasifikace →	Jedna	Dvě
Dostupné příznaky	Vybrané příznaky (s nejlepším výsledkem)		
WordNoDia, Slovníkové (celý balík)	W, DictionaryBin(int)	0,5448	
	W, Dictionary(int), Dictionary(inv)		0,5515
WordNoDia, Syntaktické (celý balík)	W, DependencyRelation, Lemma	**0,5476**	
	W, Lemma, SentenceLength		**0,5615**
Word, WordNoDia, Lemma, WordStem	Lemma	0,5425	
	Word, Lemma, WordStem		0,5614
Miloš Zeman			
	Počet úrovní klasifikace →	Jedna	Dvě
Základní příznaky (dostupné)	Vybrané příznaky (s nejlepším výsledkem)		
WordNoDia, Interpunkce (celý balík)	WordNoDia, Question, Imperative	**0,5098**	
	WordNoDia, Imperative		0,5029
WordNoDia, Morfologické (celý balík)	WordNoDia	0,5036	
	WordNoDia, FirstPerson		**0,5033**
Word, WordNoDia, Lemma, WordStem	WordNoDia, WordStem	0,5086	
	WordNoDia, WordStem		0,4973

Tabulka 8.7: Porovnání výsledků s různými kombinacemi příznaků. V případech u tématu „Zákaz kouření...", kdy ve vybraných byla WordNoDiaFeature, je označená písmenem W. Pro kratší zákaz bylo v názvech tříd vynechané slovo Feature. Slovník „int" znamená slova zvyšující intenzitu sentimentu, „inv" slova invertující sentiment.

9 Možné kroky do budoucna

Program dosahuje určitých výsledků, ty ale zatím nejsou úplně dobře použitelné v praxi. Bylo by možné ho vylepšit. Zde navrhuji několik kroků a vylepšení, které by bylo možné implementovat v budoucnu.

Jak uvádí Mochales (2011), pokud je příspěvek součástí větší diskuze, tak znalost struktury diskuze může vylepšit výsledky klasifikátoru. V této diplomové práci byla struktura diskuze záměrně ignorována. Jeden z důvodů bylo zadání, které uvádí, že metoda má mít dva vstupy – výrok ze zpravodajského textu, a komentář. Cílem nebylo klasifikovat složitější diskuzi, ale pouze jeden komentář. Dalším důvodem bylo to, že jsem se soustředila spíše na vytváření vhodného korpusu dat a vytváření vhodných příznaků, než na rozpoznávání struktury diskuze. V dalším vývoji by bylo možné implementovat složitější příznaky, které nepracují jen s jedním komentářem, ale s celou diskuzí. Například by mohly používat autora komentáře, vlákna diskuze a časovou posloupnost komentářů.

Některé příspěvky v korpusu jsou psané bez diakritiky, často se objevují překlepy a nespisovná slova. V další verzi by se mohlo přidat předzpracování dat, které by doplňovalo diakritiku a opravoval chyby. Program Majka, testovaný pro předzpracování dat, umí obnovit diakritiku. Existují také další programy, které toto umožňují.

Další možností je naprogramovat a otestovat další třídy příznaků. Program umožňuje přidat a otestovat libovolné další Feature, takže je možné experimentovat s dalšími nápady.

Korpus dat je možné vylepšit několika způsoby. Jedním z nich by bylo, přesněji definovat třídy PRO, PROTI a NIC a zvýšit shodu anotátorů. Tím by se data měla stát konzistentnější, což by mělo zlepšit výsledky klasifikátoru.

Třída PROTI je v datech zastoupena podstatně více, než třídy PRO a NIC. Třídy PRO a NIC jsou zastoupené zhruba stejně, ale třída PROTI je téměř dvojnásobná. Dalo by se experimentovat s vyvážením korpusu - buď přidáním dalších komentářů ze třídy PRO, případně NIC, nebo ubíráním komentářů PROTI. Vyvážení korpusu se dá udělat buď jednorázově, na stejný počet, nebo je možné experimentovat s postupným ubíráním komentářů PROTI (například po 100 příspěvcích).

Výsledkům by také mohla pomoci další data. Korpus vytvořený v rámci diplomové práce obsahuje dvě témata a přes 5000 příspěvků ze serveru iDNES.cz. Je možné přidat další komentáře na jiná témata nebo z jiných zdrojů. Také je možné přidat další komentáře na současná témata a zvýšit počet trénovacích dat.

10 Závěr

Metoda pro automatické rozpoznání argumentace byla navržena a otestována. Byla otestována na datech, vytvořených v rámci této diplomové práce. Pro použití v praxi je ale potřeba podniknout další kroky.

Autorka této práce se z dostupné literatury seznámila se současným stavem v detekci argumentace, a s dosud provedenými experimenty. Dále shromáždila komentáře pro datový korpus, a většinu jich anotovala. Všechny příspěvky zpracovala a vytvořila z nich korpus dat pro trénování a testování. Navrhla a implementovala příznaky, použité pro reprezentaci dat a algoritmus, který z těchto příznaků vybírá ty vhodné. Navrhla a implementovala metodu, která automaticky rozpoznává argumentaci komentářů. Metodu otestovala s různou konfigurací a výsledky předložila v této práci.

Navržená metoda využívá strojového učení, konkrétně klasifikace. Klasifikuje se do tří tříd: PRO, PROTI a NIC. Používá se klasifikátor, který se natrénuje na datech. Po natrénování se mu předloží testovací komentáře, a klasifikátor je zařadí do jedné z tříd. Komentáře jsou reprezentovány pomocí vektorů příznaků. Každý příznak byl implementován pomocí třídy „Feature".

Metoda byla testována na různých částech dat, v různých konfiguracích a s různými metodami výpočtu. Nejlepší byly výsledky na datech, která souvisí s tématem a jsou anotovaná ručně (ne automaticky). I přes pouze mírnou shodu anotátorů vycházelo lépe použít všechna související data, než jen ta, kde se anotátoři shodli. Dvouúrovňová klasifikace, (tedy ta, kde se nejprve určilo, zda příspěvek vyjadřuje postoj, a teprve potom se určovalo, jestli je postoj PRO nebo PROTI) vycházela o něco lépe, než klasifikace rovnou do jedné ze tří tříd.

Z testovaných příznaků vycházely dobře zejména: Word, WordNoDia, WordStem, Lemma, Imperative, Dictionary. Mezi použitelnými příznaky se vyskytovaly jak ty, které využívaly syntaktické předzpracování, tak ty, které pracují s obyčejným textem. Oba druhy příznaků vykazovaly podobná zlepšení. Změna dat se obvykle na výsledcích projevila víc, než změna příznaků.

Nejlepší dosažené F_1 skóre je 0,5615, a bylo dosažené u tématu „Zákaz kouření v restauracích" a na příznacích: WordFeature, LemmaFeature, SentenceLengthFeature. Základní (baseline) skóre se získalo na konfiguraci: SVM klasifikátor, jednoúrovňově, všechna ručně anotovaná související data, WordFeature. Základní F_1 skóre pro téma „Kouření" je 0.519.

Z výsledků by se dalo soudit, že pro český text je automatická detekce argumentace poměrně náročný úkol, který si rozhodně zaslouží další pozornost.

Reference

IKONOMAKIS Emmanouil K., S. Kotsiantis, V. Tampakas, 2005 *Text Classification Using Machine Learning Techniques* [online]. 2005. [cit. 2016-10-26]. Dostupné z: *https://www.researchgate.net/publication/228084521_Text_Classification_Using_Machine_Learning_Techniques*

MOHAMMAD Saif M., Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, Colin Cherry. 2016 *SemEval-2016 Task 6: Detecting Stance in Tweets* [online]. 2016. [cit. 2016-10-26]. Dostupné z: *http://aclweb.org/anthology/S/S16/*

GAMON Michael, 2004 *Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis* [online]. 2004. [cit. 2016-10-26]. Dostupné z: *https://www.microsoft.com/en-us/research/publication/sentiment-classification-on-customer-feedback-data-noisy-data-large-feature-vectors-and-the-role-of-linguistic-analysis/*

KABADJOV Mijail, Udo Kruschwitz, Massimo Poesio, Josef Steinberger, Emma Barker. 2015 OnForumS: A Shared Task on On-line Forum Summarisation [online]. [cit. 2017-10-26]. Dostupné z: *http://multiling.iit.demokritos.gr/pages/view/1531/task-onforums-data-and-information*

BRYCHCÍN Tomáš, Michal Konkol, Josef Steinberger, 2014 *UWB: Machine Learning Approach to Aspect-Based Sentiment Analysis* [online]. [cit. 2016-10-26]. Dostupné z: *http://nlp.kiv.zcu.cz/publication/25*

MOCHALES Raquel, Marie-Francine Moens. Argumentation Mining [online]. 2011. [cit. 2016-10-26]. Dostupné z: https://www.researchgate.net/publication/225336483

SOMASUNDARAN Swapna, Janyce Wiebe, 2010 *Recognizing Stances in Ideological On-Line Debates* [online]. 2010. [cit. 2016-10-26]. Dostupné z: *https://www.researchgate.net/publication/234801682_Recognizing_stances_in_ideological_on-line_debates*

THOMAS Matt, Bo Pang, Lillian Lee, 2006 *Get out the vote: Determining support or opposition from Congressional floor-debate transcripts* [online]. 2006. [cit. 2016-10-26]. Dostupné z: *http://dl.acm.org/citation.cfm?id=1610075.1610122*

HABERNAL Ivan, Tomáš Ptáček, Josef Steinberger, 2013 *Sentiment Analysis in Czech Social Media Using Supervised Machine Learning* [online]. 2013. [cit. 2016-10-26]. Dostupné z: *http://nlp.kiv.zcu.cz/publication/32*

GREENE Stephan, Philip Resnik, 2009 *More thanWords: Syntactic Packaging and Implicit Sentiment* [online]. [cit. 2016-10-30]. Dostupné z: *http://www.aclweb.org/anthology/N09-1057*

MITCHELL Tom M., 1997 *Machine Learning* McGraw-Hill Science/Engineering/Math. ISBN 0070428077

HASTIE Trevor, Robert Tibshirani, Jerome Friedman, 2009 *The Elements of Statistical Learning* 2nd edition [online]. [cit. 2016-11-12]. Dostupné z: *http://statweb.stanford.edu/~tibs/ElemStatLearn/*

BERGER Adam L., Stephen A. Della Pietra, Vincent J. Della Pietra, 1996 *A Maximum Entropy Approach to Natural Language Processing* [online]. [cit. 2016-11-17]. Dostupné z: *http://dl.acm.org/citation.cfm?id=234289*

KREJZL Peter, Josef Steinberger, 2016 *UWB at SemEval-2016 Task 6: Stance Detection* [online]. [cit. 2017-01-08]. Dostupné z: *https://aclweb.org/anthology/S/S16/*

ŠMERK Pavel, Pavel Rychlý, 2009 *Majka – rychlý morfologický analyzátor* [online]. [cit. 2017-01-08]. Dostupné z: *https://is.muni.cz/publication/935762/cs*

KOVÁŘ Vojtěch, Aleš Horák, Miloš Jakubíček, 2011 *Syntactic Analysis Using Finite Patterns: A New Parsing System for Czech* [online]. [cit. 2017-01-08]. Dostupné z: *https://is.muni.cz/publication/932481*

STRAKA Milan, Hajič Jan, Straková Jana, 2016 *UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing*. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovinsko, Květen 2016.

KOHAVI Ron, 1995 *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection* [online]. [cit. 2017-02-10]. Dostupné z: *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.529*

KOHAVI Ron, George H. John, 1997 *Wrappers for feature subset selection* [online]. [cit. 2017-02-18]. Dostupné z: *http://www.sciencedirect.com/science/article/pii/S000437029700043X*

LIU Huan, Hiroshi Motoda, 2002 *Feature Selection, Extraction and Construction* [online]. [cit. 2017-02-18]. Dostupné z: *http://www.ar.sanken.osaka-u.ac.jp/~motoda/papers/fdws02.pdf*

CoNLL-U Format. 2016 *Universal Dependencies* [online]. [cit. 2017-03-24]. Dostupné z: *http://universaldependencies.org/format.html*

BRYCHCÍN Tomáš, Konopík Miloslav, 2015 *HPS: High Precision Stemmer* [online]. [cit. 2017-04-26]. Dostupné z: *http://nlp.kiv.zcu.cz/publication/67*

KONKOL Michal, 2014 *Brainy: A Machine Learning Library* [online]. [cit. 2017-04-26]. Dostupné z: *http://nlp.kiv.zcu.cz/publication/34*

KREJZL Peter, Josef Steinberger, 2016 *Stance detection in online discussions* [online]. [cit. 2017-04-26]. Dostupné z: *http://nlp.kiv.zcu.cz/publication/83*

Seznam obrázků

Seznam tabulek

Příloha - Přehled výsledků testování

Zákaz kouření v restauracích - jednoúrovňová klasifikace		
Dostupné příznaky	Vybrané příznaky	F1 skóre
WordFeature	WordFeature	0,5171
WordNoDiaFeature	WordNoDiaFeature	0,5317
WordStemFeature	WordStemFeature	0,5374
LemmaFeature	LemmaFeature	0,5425
BigramFeature	BigramFeature	0,4372
WordNoDiaFeature, Slovníkové	WordNoDiaFeature, DictionaryBinFeature(int)	0,5448
WordNoDiaFeature, Morfologické	WordNoDiaFeature, FirstPersonFeature	0,5369
WordNoDiaFeature, Interpunkce	WordNoDiaFeature	0,5317
WordNoDiaFeature, Syntaktické	WordNoDiaFeature, DependencyRelationFeature, LemmaFeature	0,5476*
Word, WordNoDia, Lemma, Stem	LemmaFeature	0,5425
Základní (balík features)	WordFeature, BigramFeature	0,5180
* Nejlepší výsledek. Výsledky pro jednotlivé třídy pro tuto kombinaci příznaků: F1 NIC = 0,3736 F1 PRO = 0,4740 F1 PROTI = 0,6212		

Miloš Zeman - jednoúrovňová klasifikace		
Dostupné příznaky	Vybrané příznaky	F1 skóre
WordFeature	WordFeature	0,4774
WordNoDiaFeature	WordNoDiaFeature	0,5036
WordStemFeature	WordStemFeature	0,4860
LemmaFeature	LemmaFeature	0,4897
BigramFeature	BigramFeature	0,4174
WordNoDiaFeature, Slovníkové	WordNoDiaFeature	0,5036
WordNoDiaFeature, Morfologické	WordNoDiaFeature	0,5036
WordNoDiaFeature, Interpunkce	WordNoDiaFeature, QuestionFeature, ImperativeFeature	0,5098*
WordNoDiaFeature, Syntaktické	WordNoDiaFeature	0,5036
Word, WordNoDia, Lemma, Stem	WordNoDiaFeature, WordStemFeature	0,5086
Základní (balík features)	WordFeature, UppercaseFeature	0,4826
* Nejlepší výsledek. Výsledky pro jednotlivé třídy pro tuto kombinaci příznaků: F1 NIC = 0,3109 F1 PRO = 0,4066 F1 PROTI = 0,6130		

Miloš Zeman - dvouúrovňová klasifikace

Miloš Zeman - dvouúrovňová klasifikace		
V 1. úrovni byly vždy použité LemmaFeature a NthWordFeature (první slovo)		
Dostupné příznaky	Vybrané příznaky	F1 skóre
WordFeature	WordFeature	0,4744
WordNoDiaFeature	WordNoDiaFeature	0,4907
WordStemFeature	WordStemFeature	0,4950
LemmaFeature	LemmaFeature	0,4866
WordNoDiaFeature, Slovníkové	WordNoDiaFeature, DictionaryBinFeature(int)	0,4994
WordNoDiaFeature, Morfologické	WordNoDiaFeature, FirstPersonFeature	0,5033*
WordNoDiaFeature, Interpunkce	WordNoDiaFeature, ImperativeFeature	0,5029
WordNoDiaFeature, Syntaktické	WordNoDiaFeature, SubjectFeature	0,5016
Word, WordNoDia, Lemma, Stem	WordStemFeature, WordNoDiaFeature	0,4973
Základní (balík features)	WordFeature, OneFeature, NthWordFeature	0,4865
* Nejlepší výsledek.		

	Miloš Zeman - dvouúrovňová klasifikace, podrobné výsledky jednotlivých úrovní a tříd								
V 1. úrovni byly vždy použité LemmaFeature a NthWordFeature (první slovo)									
	F1 SKÓRE								
Vybrané příznaky	1. úroveň	2. úroveň	Třídy 1. úrovně		Třídy 2. úrovně		Třídy po sloučení úrovní		
	Obě třídy	Obě třídy	NONE	STANCE	PRO	PROTI	PRO	PROTI	NIC
WordFeature	0,5586	0,5865	0,3055	0,8118	0,4239	0,7491	0,3050	0,6439	0,3055
WordNoDia	0,5586	0,6077	0,3055	0,8118	0,4506	0,7648	0,3265	0,6548	0,3055
WordStem	0,5586	0,6106	0,3055	0,8118	0,4677	0,7535	0,3418	0,6482	0,3055
Lemma	0,5586	0,5998	0,3055	0,8118	0,4484	0,7511	0,3310	0,6421	0,3055
WordNoDia, FirstPerson	0,5586	0,6221	0,3055	0,8118	0,4889	0,7553	0,3580	0,6486	0,3055
WordNoDia, Imperative	0,5586	0,6238	0,3055	0,8118	0,4767	0,7709	0,3458	0,6600	0,3055

Zákaz kouření v restauracích - dvouúrovňová klasifikace

Zákaz kouření v restauracích - dvouúrovňová klasifikace		
V 1. úrovni byly vždy použité LemmaFeature a NthWordFeature (první slovo)		
Dostupné příznaky	Vybrané příznaky	F1 skóre
WordFeature	WordFeature	0,5323
WordNoDiaFeature	WordNoDiaFeature	0,5470
WordStemFeature	WordStemFeature	0,5415
LemmaFeature	LemmaFeature	0,5490
WordNoDiaFeature, Slovníkové	WordNoDiaFeature, Dictionary(int), Dictionary(inv)	0,5515
WordNoDiaFeature, Morfologické	WordNoDiaFeature	0,5470
WordNoDiaFeature, Interpunkce	WordNoDiaFeature, QuotedFeature	0,5520
WordNoDiaFeature, Syntaktické	WordNoDiaFeature, LemmaFeature, SentenceLengthFeature	0,5615*
Word, WordNoDia, Lemma, Stem	LemmaFeature, WordFeature, WordStemFeature	0,5614
Základní (balík features)	WordFeature, UppercaseHalfFeature	0,5323
* Nejlepší výsledek.		

	Zákaz kouření v restauracích - dvouúrovňová klasifikace, podrobné výsledky jednotlivých úrovní a tříd								
V 1. úrovni byly vždy použité LemmaFeature a NthWordFeature (první slovo)									
	F1 SKÓRE								
Vybrané příznaky	1. úroveň	2. úroveň	Třídy 1. úrovně		Třídy 2. úrovně		Třídy po sloučení úrovní		
	Obě třídy	Obě třídy	NONE	STANCE	PRO	PROTI	PRO	PROTI	NIC
WordFeature	0,5844	0,6516	0,3472	0,8216	0,5364	0,7669	0,4060	0,6585	0,3472
WordNoDia	0,5844	0,6698	0,3472	0,8216	0,5623	0,7772	0,4268	0,6671	0,3472
WordStem	0,5844	0,6630	0,3472	0,8216	0,5559	0,7701	0,4240	0,6589	0,3472
Lemma	0,5844	0,6729	0,3472	0,8216	0,5677	0,7780	0,4303	0,6676	0,3472
WordNoDia, Lemma, SentenceLength	0,5844	0,6875	0,3472	0,8216	0,5857	0,7893	0,4512	0,6717	0,3472
Lemma, Word, WordStem	0,5844	0,6882	0,3472	0,8216	0,5838	0,7926	0,4479	0,6750	0,3472