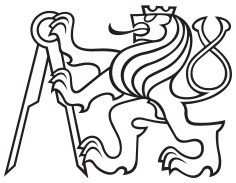
Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Aspektově orientovaná analýza sentimentu

Bc. Jakub Macháček

Vedoucí: Ing. Božena Mannová, Ph.D.
Obor: Softwarové inženýrství
Květen 2016

Poděkování

Děkuji Ing. Boženě Mannové, Ph.D. za vedení této práce. Také děkuji všem přátelům, kteří mě při práci podporovali.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Boženy Mannové, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Praze, 27. května 2016

Abstrakt

Tato diplomová práce představuje základní teoretické poznatky z oblasti strojového učení a zpracování přirozeného jazyka, které jsou následně využity pro návrh a implementaci systému schopného extrakce subjektivních informací z krátkých textových komentářů vztahujících se k předdefinovné množině entit. Systém by měl umět určit, ke kterému aspektu komentované entity se každý názor vyjadřuje, stejně tak jako jeho polaritu a lingvistický výraz.

Klíčová slova: aspektově orientovaná analýza sentimentu, strojové učení, zpracování přirozeného jazyka, klasifikace, shluková analýza, vnoření slov

Vedoucí: Ing. Božena Mannová, Ph.D.
Software Engineering and Networking,
Karlovo náměstí 13,
Praha 2

Abstract

This diploma thesis introduces basic theoretical knowledge from the fields of machine learning and natural language processing which is then used to design and implement a system capable of extracting subjective information from short text comments regarding a set of predefined entities. The system should be able to recognize towards which aspect of the commented entity is each opinion expressed, as well as polarity and linguistic expression of that opinion.

Keywords: aspect-based sentiment analysis, machine learning, natural language processing, classification, cluster analysis, word embedding

Title translation: Aspect-Oriented Sentiment Analysis

Obsah

Obrázky

Tabulky

Kapitola 1
Úvod

Tato diplomová práce se zabývá aspektově orientovanou anlýzou sentimentu – problémem extrakce subjektivních informací z krátkých textových komentářů. Problematika spadá do oblasti zpracování přirozeného jazyka, ve které není momentální stav řešení na zcela uspokojivé úrovni, ačkoliv je mu věnováno značné úsilí.

V kapitole 2 bude popsána motivace, která stojí za strojovým zpracováním dat tohoto typu a zároveň budou definovány konkrétní cíle práce.

Významnou oblastí informatiky, jež se intenzivně využivá ve zpracování přirozeného jazyka, je strojové učení. V kapitole 3 bude tato oblast přiblížena. Zvýšená pozornost bude věnována znalostem využitelných v aspektově orientované analýze sentimentu.

V kapitole 4 budou popsány některé základní úlohy zpracování přirozeného jazyka a následně bude ukázáno, jak je lze v kombinaci se strojovým učením využít pro řešení aspektově orientované analýzy sentimentu.

Návrhem vlastního systému schopného analýzu provádět se bude zabývat kapitola 5. Jeho implementace bude stručně popsána v kapitole 6 a dosažené výsledky prezentovány v kapitole 7.

V kapitole 8 bude shrnuto, jakých cílů se v práci podařilo dosáhnout a jaké další kroky budou podniknuty pro vylepšení stávajícího řešení.

Kapitola 2
Analýza sentimentu

Se stále rostoucím počtem uživatelů se internet postupně stává čím dál intenzivnějším prostředkem mezilidské komunikace. Kromě soukromých konverzací se na internetu nachází také obrovské množství krátkých veřejně přístupných textových komentářů. Prostřednictvím sociálních sítí (*Twitter*, *Facebook* aj.) píší lidé své názory o nejrůznějších tématech. Také internetové obchody nabízejí svým zákazníkům možnost psát komentáře k zakoupeným výrobkům.

Pro mnoho různých společností i jednotlivce mají tato data velkou hodnotu. Pro obchodníky je podstatná zpětná vazba o jednotlivých výrobcích, které prodávají. Na základě ní mohou například vyřadit ze sortimentu výrobky, se kterými nemají zákazníci dobré zkušenosti. Podnikatelům může průzkum veřejného mínění pomoci identifikovat příležitosti výhodných investic či předpovědět budoucí trendy.

Posuzovat veřejné mínění manuálně, tedy čtením jednotlivých příspěvků, však znamená procházení velkého množství třeba i nesouvisejících dat, což je časově i finančně velmi náročná úloha. Z tohoto důvodu vzniká potřeba tyto příspěvky automatizovaně zpracovávat a dolovat z nich subjektivní informace (tzv. *sentimenty*). Tato potřeba se stala motivací ke vzniku nové disciplíny, *analýzy sentimentu*, známé také jako *dolování názorů*. V základním pojetí se analýzou sentimentu myslí predikce polarity textového příspěvku, nejčastěji jako *pozitivní*, *negativní* nebo *neutrální*. Princip znázorňuje tabulka 2.1. V širším pojetí je analýzou sentimentu chápou všechny úlohy, jejichž cílem je z příspěvků dolovat subjektivní informace. V tomto smyslu bude chápána i v této práci.

2.1 Aspektově orientovaná analýza sentimentu

Zatímco v některých oblastech použití je predikce celkové polarity příspěvku postačující, v jiných případech jsou k dispozici příspěvky obsahující názory o aspektech různých entit. Příkladem můžou být recenze výrobků, u kterých zákazníci hodnotí jednotlivé vlastnosti zakoupeného výrobku – kvalita, vzhled, cena, jednotlivé podčásti a další. Pro výrobce může být užitečné dozvědět se o slabinách nebo naopak přednostech jeho produktu. Na základě této zpětné vazby je možné učinit strategická rozhodnutí o výrobě nebo investicím.

Pro dobré pochopení hromadného mínění z takových komentářů je nutné

Vstup	Výstup
Součástí balení je také stručná příručka.	neutrální
Tento film byl pro mne velkým zklamáním, čekal jsem víc.	negativní
Líbí se mi jejich rovný přístup k zákazníkovi.	pozitivní

Tabulka 2.1: Princip analýzy sentimentu

provést důkladnější analýzu, protože pouhé zjištění celkové polarity příspěvku může být zavádějící. Pak se hovoří o *aspektově orientované analýze sentimentu*, která má za úkol detekovat, o jakých entitách, popř. aspektech entity, se autor v příspěvku vyjadřuje a jaký k nim zaujímá postoj.

Některé internetové obchody sice umožňují vkládání strukturovaných recenzí formou oddělených textových polí pro hodnocení jednotlivých aspektů výrobku, ale uživatelé tuto možnost často nevyužívají. V některých oblastech takové řešení ani není možné, např. pokud se analyzuje veřejné mínění z internetových diskuzí.

2.2 Vymezení zadání

Cílem této práce je vytvořit systém, který by uměl provádět aspektově orientovanou analýzu sentimentu nad velkými soubory textových příspěvků. Pro jeho použití bude nejprve nutné část příspěvků ručně projít a vytvořit nad nimi anotace. Systém následně použije anotované příspěvky k natrénování vnitřního modelu a ten použije k analýze zbývajících příspěvků. Příspěvky, které jsou před zahájením samotné analýzy ručně anotovány, tvoří tzv. *trénovací sadu*. Anotováním se rozumí připojení očekávaných sentimentů k jednotlivým příspěvkům.

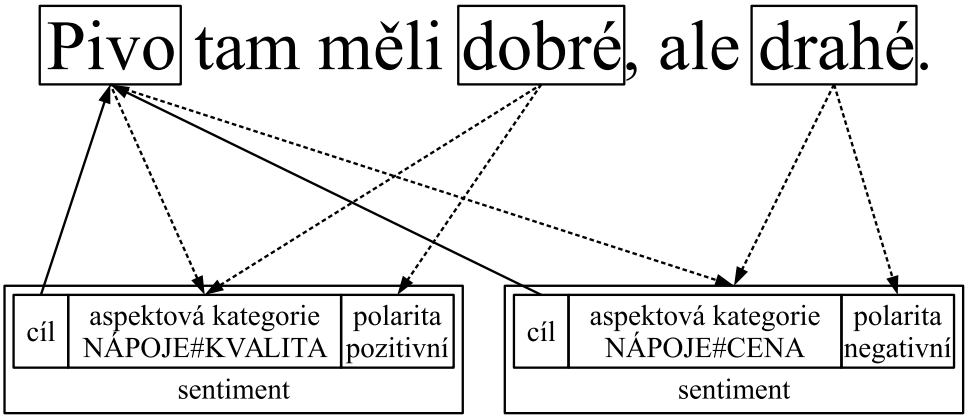
Sentiment je formálně definován jako trojice (aspektová kategorie, cíl, polarita). *Aspektová kategorie* popisuje, k jakému aspektu které entity je sentiment vyjádřen. Je tvořena spojením entity a atributu a značí se `ENTITA#ATRIBUT`. Množina entit, stejně jako množina atributů, je předem známá a konečná. Každý atribut se obecně pojí jen s některými entitami.

Aspektová kategorie je pouhý identifikátor, který je pro daný aspekt vždy stejný bez ohledu na to, jakou frází je v textu vyjádřen. V této práci budou fráze, k nimž se vztahuje nějaký sentiment, nazývány *cíle sentimentu*.

Polarita sentimentu může být *pozitivní*, *negativní* nebo *neutrální*. Neutrální polarita zahrnuje také mírně pozitivní a mírně negativní. Vztah mezi aspektovou kategorií, cílem a polaritou sentimentu ilustruje příklad na obrázku 2.1.

Přesnost systémů provádějících analýzu sentimentu se obvykle zvyšuje s množstvím textu, ve kterých se sentimenty mají detekovat. Tato práce se bude zabývat analýzou sentimentu odděleně pro každou větu příspěvku. Dá se ovšem předpokládat, že při použití systému na delší texty bude jeho přesnost vyšší.

Systém bude umět řešit několik různých úloh podle toho, jaká konkrétní data bude chtít uživatel ze sady příspěvků extrahovat. Některé úlohy předpo-



Obrázek 2.1: Schéma sentimentů nad ukázkovou větou

kládají, že jim budou na vstupu kromě samotného textu příspěvku předány také nějaké anotace. To může být výhodné v případě, kdy je má uživatel k dispozici, neboť jejich použití může vést k přesnějším výsledkům. Systém bude umět řešit následující úlohy:

1. **CAT: Detekce aspektových kategorií**. Úkolem této úlohy je pro každou vstupní větu vrátit množinu aspektových kategorií, k nimž se ve větě vztahuje alespoň jeden sentiment.
2. **TAR: Detekce cíle sentimentu**. Úkolem je pro každou vstupní větu vrátit množinu cílů, k nimž se ve větě vztahuje alespoň jeden sentiment.
3. **CAT+TAR: Detekce aspektových kategorií a cílů**. Tato úloha je kombinací předchozích dvou úloh. Úkolem je ke vstupní větě vrátit všechny dvojice (aspektová kategorie, cíl), tedy všechny sentimenty bez určení polarity. Cíl nabývá prázdné hodnoty, pokud není ve větě vyjádřen.
4. **POL1: Predikce polarity 1**. Úloze jsou kromě vstupního textu předány také již detekované aspektové kategorie. Úkolem je pro každou kategorii predikovat polaritu sentimentu.
5. **POL2: Predikce polarity 2**. Tato úloha je modifikací předchozí úlohy. Místo samotných aspektových kategorií jsou však úloze předány dvojice (aspektová kategorie, cíl) a úkolem je predikovat polaritu sentimentu pro každou dvojici.

Tabulka 2.2 ukazuje očekávané výstupy výše definovaných úloh pro ukázkovou vstupní větu. Podle charakteru příspěvků bude rozlišováno mezi těmito atributy datových sad:

- *Jazyk* – přirozený jazyk, kterým jsou příspěvky psány
- *Doména* je oblast, ze které datová sada pochází. V rámci domény by se měly příspěvky datových sad vyjadřovat o stejné nebo podobné množině entit.

Servírka byla nepříjemná, ale jídlo nám chutnalo a bylo za hubičku.		
CAT	**TAR**	**CAT+TAR**
OBSLUHA#OBECNĚ	Servírka	(OBSLUHA#OBECNĚ, Servírka)
JÍDLO#KVALITA	jídlo	(JÍDLO#KVALITA, jídlo)
JÍDLO#CENA		(JÍDLO#CENA, jídlo)

Vstupní anotace	**POL1**
OBSLUHA#OBECNĚ	negativní
JÍDLO#KVALITA	pozitivní
JÍDLO#CENA	pozitivní

Vstupní anotace	**POL2**
(OBSLUHA#OBECNĚ, Servírka)	negativní
(JÍDLO#KVALITA, jídlo)	pozitivní
(JÍDLO#CENA, jídlo)	pozitivní

Tabulka 2.2: Příklad správného výstupu pro jednotlivé úlohy aspektově orientované analýzy sentimentu

Systém by měl být navržen tak, aby po vynaložení minimálního manuálního úsilí vykazoval uspokojivé výsledky nezávisle na jazyku a doméně zpracovávaných dat.

Kapitola 3

Strojové učení

Analýza sentimentu je většinou řešena s využitím *strojového učení*. Strojové učení je oblastí informatiky, jejímž cílem je umožnit strojům učit se bez jejich explicitního naprogramování. Učením se rozumí zpracování množiny *vzorků*, na jejichž základě je poté stroj schopen učinit určité předpovědi. Rozlišují se tři základní problémy:

- *učení s učitelem*,
- *učení bez učitele*,
- *zpětnovazebné učení*.

Každou instanci datového vzorku si lze formálně představit jako *vektor příznaků* (*feature vector*), který zároveň tvoří rozhraní daného vzorku. Učící algoritmus pracuje výhradně s ním a nesnaží se data analyzovat mimo daný vektor nebo hledat vnitřní strukturu v rámci jedné hodnoty příznaku. Výběr vhodných příznaků je tedy klíčový pro dosažení maximální přesnosti modelu.

Jednotlivé vzorky si lze také představit jako body v n-rozměrném prostoru, kde n je počet příznaků a každý rozměr právě jednomu příznaku odpovídá. Tento prostor je nazýván *příznakový prostor* (*feature space*).

Algoritmy strojového učení se podle způsobu práce s jednotlivými vzorky dat dělí na

- *dávkové* (*batch*),
- *inkrementální* (též *stochastické*, *online*).

Dávkové algoritmy představují tradiční způsob pracování s daty, kdy se parametry modelu vypočítají na základě zpracování všech vzorků a v případě přidání nových vzorků je pro aktualizaci modelu nutné celou proceduru opakovat. Inkrementální algoritmy naproti tomu zpracovávají vzorky sekvenčně a model je aktualizován s každým přidaným trénovacím vzorkem. Tento přístup umožňuje podstatně rychlejší zpracování velkého množství dat, ale může dosahovat menší přesnosti. Pokud je provedeno více průchodů učení, je možné tuto nevýhodu zmírnit. Inkrementální algoritmy jsou navíc citlivé na pořadí vstupních vzorků.

Jméno	Definice $L(p, y)$
Kvadrátová ztráta	$(p - y)^2$
Kvantilová ztráta	$\tau(y - p)\mathbb{I}(y \geq p) + (1 - \tau)(p - y)\mathbb{I}(y \leq p)$
Závěsová (hinge)	$\max(0, 1 - yp)$
Logistická ztráta	$\log(1 + \mathrm{e}^{-yp})$
Křížová entropie	$-t \ln(p) - (1 - t) \ln(1 - p)$, kde $t = \frac{1+y}{2} \rightarrow t \in \{0, 1\}$

Tabulka 3.1: Ztrátové funkce – argument p v definicích reprezentuje predikovanou hodnotu, zatímco $y \in \{-1, +1\}$ je skutečná hodnota.

3.1 Učení s učitelem

Učení s učitelem je typ úlohy, ve které jsou všechna data z tzv. *trénovací sady* předem označena očekávaným výsledkem. Cílem algoritmu je na základě vstupu a výstupu všech vzorků najít obecné pravidlo, se kterým by pak bylo možné odhadnout výstup dalších dat, která už označená nejsou. Formálně řečeno, model problému je funkce f, kde $f(x)$ je odhadovaný výsledek pro příznakový vektor x. Cílem algoritmu je najít takovou funkci f, která pro co nejvyšší počet různých příznakových vektorů vrací správný výstup.

Podle typu výstupu se učení s učitelem dělí především na *regresi* a *klasifikaci*. Regresní úloha mapuje vstupy na spojitou množinu, zatímco klasifikační na diskrétní množinu výstupů.

3.1.1 Ztrátová funkce

Přesnost modelu je pro konkrétní parametry stanovena součtem chyb, kterých se model dopustí u jednotlivých vzorků trénovací sady. Algoritmus se snaží najít takové parametry, pro které je součet chyb nejnižší. Velikost chyby jednotlivých vzorků je stanovena tzv. *ztrátovou funkcí* (anglicky *loss* nebo také *cost function*). V závislosti na povaze úlohy se obecně volí různé ztrátové funkce. Ty, které se obvykle používají při řešení úloh učení s učitelem, zobrazuje tabulka 3.1.

3.1.2 Klasifikace

Klasifikace je souborem úloh, jejichž principem je přiřazování tříd k instancím datových vzorků. Množina tříd je vždy konečná a předem známá. Rozlišují se především

- *binární* klasifikace,
- *vícetřídní* (*multi-class*) klasifikace,
- *víceznačková* (*multi-label*) klasifikace.

Binární klasifikací se rozumí problém přiřazení právě jedné ze dvou tříd každé instanci datového vzorku. V tomto případě jsou třídy často označovány jako *negativní* (0) a *pozitivní* (1). Vícetřídní klasifikace je obdobou klasifikace

binární, ale počet tříd, které mohou být každé instanci přiřazeny, je vyšší než dva. *Víceznačková* klasifikace je problém přiřazení 0 až n tříd (z celkového počtu n) každé instanci datového vzorku.

Nadroviny (prostory s $n-1$ dimenzemi), které v n-rozměrném příznakovém prostoru od sebe oddělují poloprostory odpovídající jednotlivým třídám, se nazývají *rozhodovací hranice* (*decision boundaries*).

Klasifikátorem je nazývána entita, která je schopná provádět klasifikaci. Pod tímto termínem je obvykle míněna funkční implementace některého klasifikačního algoritmu.

Některé klasifikační algoritmy mají schopnost nepredikovat u daného datového vzorku pouze jeho příslušnost k nějaké třídě (příp. třídám), ale odhadnout pravděpobnost příslušnosti ke všem jednotlivým třídám. Klasifikátory s touto vlastností jsou označovány jako *pravděpobnostní klasifikátory*.

3.1.3 Transformace klasifikačních úloh

Binární klasifikace je základní a nejjednodušší klasifikační úlohou. Některé algoritmy ani přímo neřeší ostatní klasifikační úlohy, protože ty jsou svou povahou složitější. I takové algoritmy je ale možné použít pro řešení dalších klasifikační úloh, pokud se použije některá tranformační metoda.

Nejběžnějším algoritmem tranformace vícetřídní klasifikace na binární je *one-vs.-all*. Předpokladem této techniky je využití pravděpodobnostního klasifikačního algoritmu. V první fázi algoritmu se vytvoří samostatný binární klasifikátor pro každou třídu. Každá instance datového vzorku z trénovací sady se potom použije pro trénování všech klasifikátorů. Klasifikátoru odpovídajícímu očekávané třídě je instance předána jako pozitivní a ostatním klasifikátorům jako negativní příklad. Při klasifikaci neoznačkovaného vzorku se postupuje tak, že je vzorek předán všem klasifikátorům a každý z nich vrátí pravděpodobnost, s jakou vzorek patří do odpovídající třídy. Nakonec je vybrána třída, jejíž pravděpodobnost je nejvyšší. Nevýhodou této metody je, že prosté porovnávání výstupních pravděpobností může být nevhodné v případě nevyváženosti tříd (viz kapitola 3.4.2).

Metoda *binární relevance* umožňuje transformovat problém víceznačkové klasifikace na binární klasifikaci. Podobně jako u *one-vs.-all* je vytvořen samostatný klasifikátor pro každou třídu a i učení probíhá stejným způsobem. Rozdíl je při přiřazování tříd neoznačkovaným instancím, kdy se nevybere pouze třída s nejvyšší pravděpodobností, nýbrž všechny třídy pro než platí, že pravděpodobnost příslušnosti instance k dané třídě je vyšší než předem stanovaný práh. V případě použití nepravděpodobnostní klasifikace je postup podobný, pouze se neporovnávají pravděpobnosti s prahem.

3.1.4 Logistická regrese

Logistická regrese je jednoduchou metodou odhadu pravděpobnosti určitého jevu na základě známých znalostí (nezávislých proměnných). Tento model lze využít pro binární klasifikaci, kde patří k nejúčinnějším a nejpoužívanějším.

Nechť X je vektor všech datových vzorků trénovací sady, přičemž pro $\forall i$ je X_i vektor příznaků i-tého vzorku. Ztrátová funkce L je definována jako

$$L(a,b) = \begin{cases} -log(a) & \text{pokud } b = 1 \\ -log(1-a) & \text{pokud } b = 0 \end{cases}.$$

Logistická regrese je založena na nalezení vektoru parametrů θ tak, aby celková chyba

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(h_\theta(X_i), Y_i)$$

byla minimální. Konstanta n značí počet datových vzorků. Pro vektor $Y \in \{0,1\}^n$ platí, že pro $\forall i$ je Y_i třídou, do které patří vzorek X_i. Funkce

$$h_\theta(x) = \frac{1}{1 + \mathrm{e}^{-\theta^T x}}$$

je nazývána *hypotézou* a její výstup je chápán jako pravděpobnost, že vzorek x patří do třídy 1 (*pozitivní*). Pro klasifikaci vzorků testovací sady je použita funkce h_θ – logistická regrese je tedy pravděpobnostním modelem.

Protože úlohou *logistické regrese* je nalezení co nejvhodnějšího vektoru θ, je její řešení optimalizačním problémem. Významnou výhodou je skutečnost, že funkce celkové chyby J je vždy konvexní a je tedy možné použít *gradientní sestup* s garancí nalezení globálního minima.

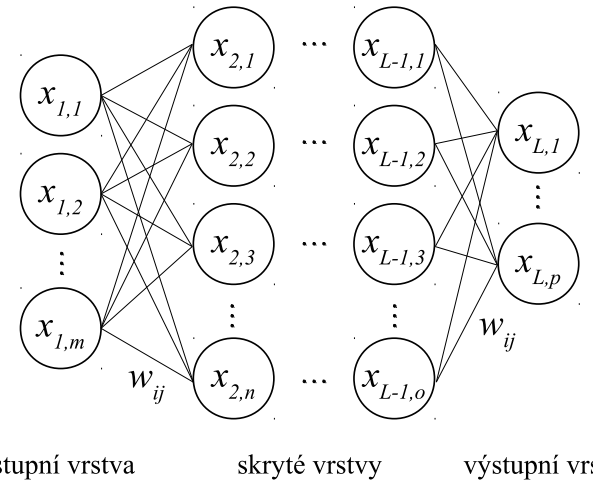
3.1.5 Podpůrné vektory

Metoda *podpůrných vektorů* (anglicky *support vector machines*, *SVM*) je založena na principu lineárního rozdělení n-rozměrného prostoru příznaků pomocí nadroviny a to takovým způsobem, že oba vzniklé poloprostory reprezentují konkrétní třídy a tudíž obsahují vzorky patřící do dané třídy. Nadrovina je vybrána tak, aby vzdálenost nejbližšího vzorku od ní byla v každém poloprostoru maximální a zároveň součet chyb byl minimální. Chybou vzorku ležícího ve špatném poloprostoru se myslí jeho vzdálenost od dělící nadroviny. Protože jde o optimalizační problém složený ze dvou kritérií, je dodatečně definována kompromisní konstanta C a úkolem podpůrných vektorů je minimalizovat výraz

$$\frac{1}{2}|w|^2 + C\sum_i E_i,$$

kde w je normálový vektor dělící nadroviny a E je vektor chyb špatně umístěných bodů. Metoda podpůrných vektorů má několik výhod:

- Díky lineárnímu přístupu je méně náchylná k přeučení.
- Její asymptotická časová složitost je $O(n^3 m)$, kde n je počet trénovacích vzorků a m je počet příznaků.
- Metodu lze použít i v případech, kdy příznakový prostor není možné účinně lineárně rozdělit. V takovém případě je možné rozšířit prostor o více dimenzí aplikováním tzv. *kernel triku* [Sch01].



Obrázek 3.1: Schéma dopředné neuronové sítě. Neurony jsou značeny $x_{a,b}$, kde a je pořadí vrstvy a b je pořadí neuronu v rámci vrstvy. Synapse jsou zobrazené jako spojnice mezi neurony w_{ij}, kde i a j značí nerony, které dané synapse spojují.

3.1.6 Umělé neuronové sítě

Umělé neuronové sítě jsou souborem modelů aproximujících funkce, jehož koncept byl inspirován biologickými neuronovými sítěmi. Síť si lze představit jako množinu výpočetních jednotek nazývané *neurony*, které jsou vzájemně propojené komunikačními cestami, *synapsemi*. Neurony mají vstupy a výstupy a jsou uspořádány do za sebou jdoucích vrstev.

První vrstva neuronů je nazývána *vstupní vrstvou* a neuronům v ní ležící jsou na vstupy přivedeny hodnoty nezávislých proměnných (příznaků). Poslední vrsta se nazývá *výstupní vrstvou* a výstupy neuronů v ní ležící jsou výsledkem aproximované funkce. V případě modelování klasifikačního problému obsahuje výstupní vrstva jediný neuron, na základě jehož výstupu jsou jednotlivým vzorkům predikovány třídy. Všechny ostatní vrstvy jsou nazývány *skryté*.

Každá synapse násobí datový tok určitou hodnotou. Existuje více druhů neuronových sítí, které se liší především typem synapsí. V *dopředné* (*feedforward*) síti jsou synapse jednosměrné a každá spojuje výstup nějakého neuronu se vstupem jiného neuronu ležícího v následující vrstvě. Schéma dopředné neuronové sítě ukazuje obrázek 3.1.

Úlohou každého neuronu je přečtení svých vstupů, provedení výpočtu nad těmito vstupy a poslání výsledku na všechny své výstupy. Spočítání výstupu je označováno jako *aktivace neuronu*, funkce

$$h(x) = g(b + w^T x),$$

kde x je vektorem hodnot na vstupech neuronu, w vektorem obsahující váhy vstupních synapsí a b je skalár nazývaný *bias*. Funkce g se označuje jako *aktivační funkce* a může být zvolena z několika variant.

Analogicky k logistické regresi je úkolem neuronové sítě nalézt takové parametry sítě (váhy synapsí, biasy neuronů), že celková chyba modelu je

minimální. Trénování sítě je tedy optimalizační problém. Zajímavou skutečností je, že pro kteroukoliv funkci lze sestrojit neuronovou síť, která by s libovolnou přesností tuto funkci aproximovala. Toto tvrzení platí i pro dopředné sítě s jedinou skrytou vrstvou a konečným počtem neuronů (*universal approximation theorem* [Cyb89]). Nalezení co nejlepších parametrů je však *nekonvexní optimalizační problém* a tato skutečnost silně limituje možnosti využití neuronových sítí.

3.1.7 Další klasifikační algoritmy

Algoritmů provádějících klasifikaci je mnoho a každý z nich obvykle přináší různé výsledky s ohledem na typ problému, který se s nimi řeší. Mezi další často používané patří například

- naivní Bayesův model (naïve Bayes),
- náhodný les (random forest),
- metoda k-nejbližších sousedů (k-nearest neighbors),
- perceptron (jednoneuronová síť),
- maximální entropie (zobecnění logistické regrese na vícetřídní klasifikaci).

3.1.8 Model strukturovaných predikcí

U všech doposud popsaných algoritmů byla předpokládána jedna společná vlastnost – výstupní predikce představovaly skalární hodnotu. Ačkoliv má tento přístup širokou škálu využití, některé úlohy mají charakter vedoucí na *strukturované predikce*. Mezi typické patří *sekvenční značkování* (*sequence labeling*), kam spadá celá řada problémů spojená se zpracováním přirozeného jazyka. V těchto úlohách se berou v úvahu závislosti mezi *značkami* tvořící výstupní sekvenci. Možnými výstupy příslušných algoritmů jsou pak všechny variace s opakováním z množiny všech značek. Mezi modely strukturovaných predikcí se řadí například

- podmíněná náhodná pole (conditional random fields) [LMP01],
- strukturované podůrné vektory (structured support vector machine),
- skryté Markovovy modely (hidden Markov models),
- Markovovy modely maximální entropie (maximum entropy Markov models).

3.2 Učení bez učitele

Učení bez učitele je typem úlohy, ve které nejsou žádná data označena očekávaným výsledkem. Cílem algoritmu je samostatně najít v datech vzory, ze kterých je poté možné něco vyvodit.

3.2.1 Shluková analýza

Shlukování je typem úlohy, ve které má být soubor objektů rozdělen do určitého počtu tříd, tzv. *shluků*, takovým způsobem, že objekty v rámci stejné třídy jsou si podobnější než objekty z různých tříd. Speciálním případem je *hierarchické shlukování*, jehož cílem je vytvoření hierarchie shluků.

3.2.2 Brownovo shlukování

Brownovo shlukování (*Brown clustering*) je hierarchickou shlukovací úlohou, která je nejčastěji aplikována na jazykové korpusy obsahující velké množství slov. Úloha byla v roce 1992 popsána v článku [BdM+92]. Na základě výběru konstanty k je cílem roztřídit všechna slova vstupního korpusu na k skupin takovým způsobem, že slova v rámci jedné skupiny se v daném korpusu objevují v podobném kontextu. Kontext je definován jako předcházející a následující slovo. Je-li tedy P_x množinou skupin, do nichž patří slova, která často předcházejí slovu x a analogicky F_x je množinou skupin, do nichž patří slova, jenž často následují po slovu x, pak pro slova a a b patřící stejné skupině by mělo platit, že P_a by měla být podobná P_b a F_a by měla být podobná F_b.

Skutečnost, že slova patřící stejné skupině se v korpusu vyskytují v podobném kontextu, implikuje podobnost daných slov. Při dostatečně velkém korpusu je tedy shlukováním možné identifikovat synonyma, mezi něž patří také slova napsaná s typickými pravopisnými chybami.

Nechť V je množinou všech slov $w_1, w_2, \ldots, w_n$ vstupního korpusu a C je funkce mapující slova na skupiny, tedy $C : V \rightarrow \{1, 2, \ldots, k\}$. Dále nechť $e(w, c)$ značí pravděpobnost, že slovo w patří do skupiny c a $q(c_1, c_2)$ značí pravděpobnost, že slovo patřící skupině c_1 následuje po slovu patřící skupině c_2. Funkce $Q(C)$ je mírou, jak moc mapovací funkce C vyhovuje danému korpusu a je definována jako

$$Q(C) = \sum_{i=1}^{n} log\, e(w_i, C(w_i))q(C(w_i), C(w_{i-1})).$$

Úlohou problému je najít takovou mapovací funkci C, že $Q(C)$ je maximální.

Nalezení optimálního řešení úlohy pro velké korpusy není kvůli její složitosti proveditelné. V praxi se nejvíce používá aproximační algoritmus popsaný v [BdM+92], jehož složitost je $O(k^2|V|)$. V inicializační části je množina slov korpusu V tranformována na seznam slov seřazený od nejfrekventovanějších slov po nejméně frekventovaná. Prvních k slov je přiřazeno samostatným k skupinám. V následujících $|V| - k$ iteracích je vždy další v pořadí nejfrekventovanější slovo přiřazeno do své vlastní skupiny, čímž se počet skupin zvýší na $k+1$. Poté je pro všechny možné kombinace párů $k+1$ skupin zkusmo provedeno sloučení. Pár, po jehož sloučení vrátí $Q(C)$ nejvyšší hodnotu, zůstane sloučený, čímž se počet skupin opět zredukuje na k a iterace tím končí.

Tento algoritmus bývá v praxi používán v modifikované verzi [MGZ04], která paralelně k původnímu algoritmu navíc sestavuje množinu binárních stromů. Na začátku je k nejfrekventovanějších slov reprezentováno k uzly

tvořících k samostatných stromů. Při každém přiřazení slova nové třídě je zároveň toto slovo reprezentováno novým stromem o jednom uzlu a analogicky při každém sloučení dvou skupin jsou stromy daných skupin sloučeny do jednoho nadstromu. Po vykonání $|V| - k$ výše popsaných iterací je dodatečně provedeno ještě $k - 1$ sloučení a celá množina slov je tak hierarchicky uspořádána do jednoho binárního stromu. Každé slovo se poté obvykle reprezentuje binárním řetězcem, který představuje cestu od kořene stromu po uzel daného slova. Tato reprezentace je velice výhodná pro praktické použití, neboť pouhým oseknutím cesty na prvních n znaků je možné zjistit příslušnost slova ke skupině pro různé hodnoty $k \doteq n^2$ ($n \in \mathbb{N}$, strom je nevyvážený) bez nutnosti opakování celého algoritmu.

3.2.3 K-means

Metoda *k-means* je založena na jednoduchém principu rozdělení objektů do k skupin na základě jejich pozic v prostoru příznaků.

V inicializační fázi algoritmu je do prostoru náhodně umístěno k pomocných bodů, tzv. *centroidů*. Algoritmus poté provádí iterace, ve kterých vždy nejprve přiřadí každý objekt nejbližšímu centroidu a následně přepočítá polohu všech centroidů tak, aby každý z nich ležel v těžišti objektů, které jsou k němu přiřazeny. Následkem změny polohy controidů se v následující iteraci obecně mění příslušnost objektů k jednotlivým centroidům a algoritmus iteruje tak dloho, než se příslušnost objektů k centroidům měnit přestane.

Vzhledem k principu, na jakém je algoritmus postaven, není možné metodu použít na nespojité prostory.

3.3 Zpětnovazebné učení

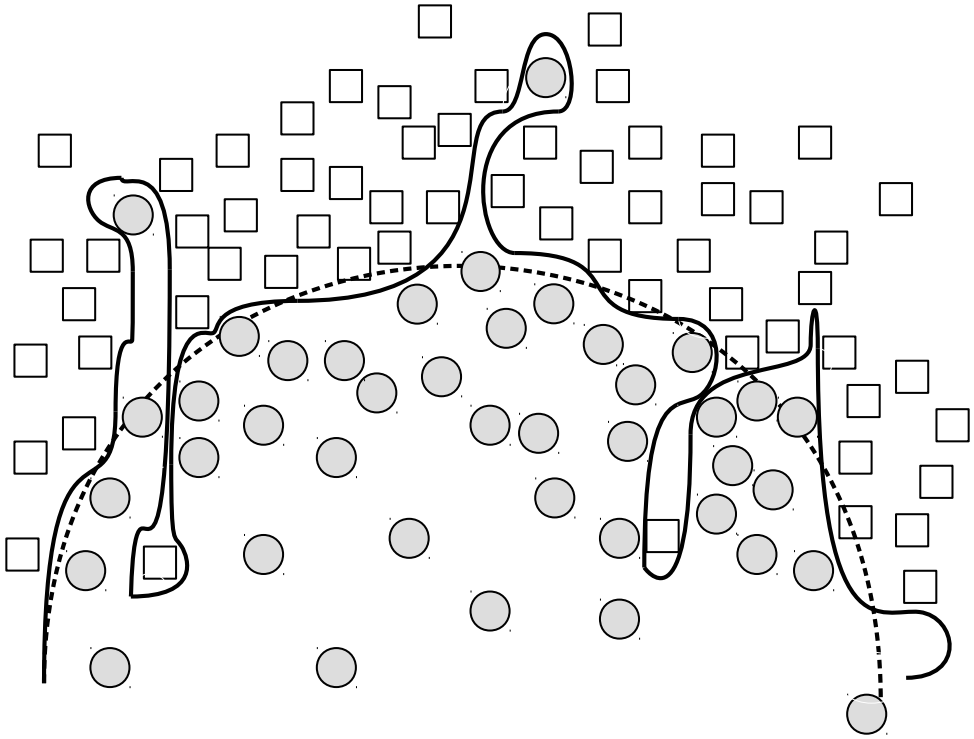
Zpětnovazebné učení (*reinforcement learning*) je typ problému, ve kterém je entita (často označována jako *agent*) umístěna do prostředí, ze kterého má možnost získávat zpětnou vazbu svých vlastních akcí. Entita však nemá žádnou informaci o tom, jak moc se blíží cíli, kterého má dosáhnout.

3.4 Úskalí spojená se strojovým učení

Pomocí strojového učení je možné relativně jednoduše řešit celou škálu problémů, které se konvenčním způsobem řeší velmi obtížně. Se strojovým učením se však také pojí určité problémy. V této části kapitoly budou popsány dva z nich – *přeučení* a *nevyváženost tříd*.

3.4.1 Přeučení

Bez ohledu na použitou metodu je cílem řešení úloh strojového učení nalezení takových parametrů, pro které daný model vykazuje nejvyšší přesnost. Pokud je ale model příliš komplexní ve srovnání s množstvím trénovacích dat,



Obrázek 3.2: Přeučení modelu – přerušovanou čarou je znázorněna skutečná rozhodovací hranice a plnou čarou rozhodovací hranice přeučeného modelu.

rozsáhlé optimalizace můžou zapříčinit *přeučení modelu* (anglicky *overfitting*). V takovém případě model v rámci trénovací sady velmi dobře klasifikuje i datové vzorky obsahující chyby měření nebo vzorky zavádějícího charakteru, které se jinak špatně klasifikují. Příklad tohoto stavu je ilustrován obrázkem 3.2. Velkým problémem přeučeného modelu je pak jeho špatná schopnost zobecňování.

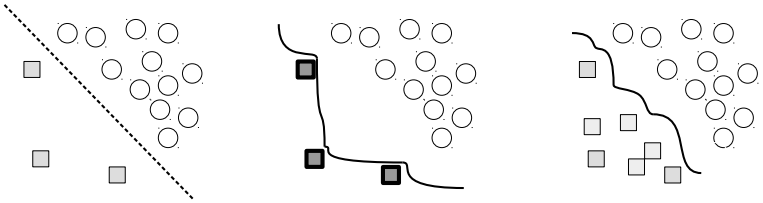
3.4.2 Nevyváženost tříd

Dalším, v praxi velmi častým, problémem strojového učení je nevyváženost tříd. Čím více se liší počty vzorků spadajících do jednotlivých tříd, tím těžší je nalézt v příznacích obecné pravidlo.

V binární klasifikaci můžou pro konkrétní testovaný příklad nastat následující situace:

Skutečnost	Predikce	Název případu	
pozitivní	pozitivní	správně pozitivní	T_P
pozitivní	negativní	falešně negativní	F_N
negativní	pozitivní	falešně pozitivní	F_P
negativní	negativní	správně negativní	T_N

Tabulka 3.2: Případy výsledků binární klasifikace



Obrázek 3.3: Srovnání rozhodovacích hranic po vytvoření umělých vzorků. Zleva: 1) očekávaná hranice, 2) hranice po klonování vzorků, 3) hranice po vytvoření nepřesných kopií

Přesnost modelu se obvykle vyjadřuje následujícími parametry:

- *Precision* je mírou udávající úspěšnost modelu v pozitivních predikcích, $P = \frac{T_P}{T_P + F_P}$.
- *Recall* je mírou udávající schopnost neopomínání pozitivních příkladů, $R = \frac{T_P}{T_P + F_N}$.

Podle konkrétního případu použití mají precision a recall z praktického hlediska různou váhu. Například při klasifikaci emailů na vyžádáné a nevyžádané zprávy, kde nevyžádaná zpráva je pokládána za pozitivní nález, je velmi důležitá vysoká hodnota *precision*. Při její nízké hodnotě by totiž vyšší počet vyžádáných zpráv byl klasifikován jako nevyžádané, což by v tomto konkrétním případě bylo závažnějším problémem než klasifikování nevyžádaných zpráv jako vyžádané.

Klasifikační algoritmy však mají tendenci mezi *precision* a *recall* nerozlišovat a spíše se zaměřují na minimalizaci celkového počtu chyb. Tento problém lze řešit dvěma základními přístupy: modifikací ztrátové funkce algoritmu nebo vyrovnáním rozdílu v počtu vzorků spadající pod jednotlivé třídy.

Ztrátovou fukci algoritmu je možné upravit tak, že při výpočtu bude reflektovat váhu podle typu chyby. Jeden F_N výsledek pak může mít stejnou váhu jako k F_P výsledků, kde k je vhodně zvolená konstanta.

Vyrovnání rozdílů v datových sadách je možné provést buď odebráním určitého množství vzorků z dominantních tříd nebo umělým vytvářením nových vzorků v málo zastoupených třídách. První případ vede k zbytečné ztrátě datových vzorků a tím pádem i k degradaci učení. Ve druhém případě je podstatná technika tvorby nových vzorků. Pokud jsou vytvářeny prostou duplikací existujících vzorků (často dosaženo zvýšením priority), pak dochází k přeučení modelu kvůli posunu rozhodovací hranice blíže k samotným vzorkům. Úspěšnější technikou je vytváření nepřesných kopií. V algoritmu *SMOTE* [CBHK02] vznikne nový vzorek tak, že na základě dvou blízkých existujících vzorků se v prostoru příznaků umístí nová instance do náhodné polohy na úsečce mezi danými existujícími vzorky. Algoritmus *SMOTE* je na obrázku 3.3 srovnán s duplikací vzorků.

Kapitola 4

Přístupy

Tato kapitola se věnuje možnostem řešení aspektově orientované analýzy sentimentu. Nejprve budou popsány základní podpůrné techniky a nástroje *zpracování přirozeného jazyka* a následně bude vysvětleno, jak k jednotlivým úlohám aspektově orientované analýzy sentimentu přistoupili tvůrci již existujících řešení. Na základě těchto informací bude v kapitole 5 navržen vlastní systém.

4.1 Základní techniky a nástroje

V této části kapitoly budou teoreticky probrány některé jednoduché techniky a nástroje, které se často využívají v analýze sentimentu. Některé z nich jsou typickými úlohami *zpracování přirozeného jazyka*.

4.1.1 Značkování slovních druhů

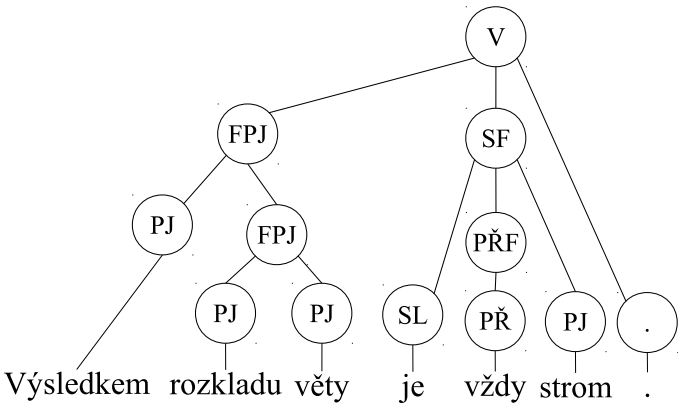
Značkování slovních druhů (*part-of-speech tagging*, *POS tagging*) je proces, ve kterém se všem slovům vstupního textu přiřadí právě jedna značka z předem definované množiny. Značky indikují především slovní druhy, ale užitečné mohou být i další informace, např. stupeň u přídavných jmen nebo čas u sloves.

Značkovače současné doby mívají úspěšnost $95 - 98\%$ správného označení jednotlivých slov. Jako úspěšné se prokázaly například tyto přístupy:

- použití *sekvenčního značkování*, konkrétněji *skrytých Markovových modelů*,
- dynamické programování,
- učení bez učitele, kdy se podobně jako u *Brownova shlukování* třídí slova podle kontextu.

Značkovače obvykle dodržují konvence pojmenování jednotlivých značek[1]. Každé slovo vstupního textu může být v analýze sentimentu obecně různě relevantní v závislosti na slovním druhu a podrobnějších informacích zjištěných ze značky.

[1] viz www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html



Obrázek 4.1: Strukturální rozklad věty

4.1.2 Lematizace a stematizace

Množství tvarů, jež mohou slova nabývat, je jednou z komplikací zpracování přirozených jazyků. Slova jsou proto obvykle předzpracována před jejich samotnou interpretací. Ve většině případů jsou všechna velká písmena převedena na malá. Často se však také více různých tvarů stejného slova (např. *krásný*, *krásná*, *krásných*) nahradí společným slovem. Většinou se v tomto ohledu za tvary stejného slova považují i slova jiného druhu (např. *krásný* a *krásně*) a jejich nahrazování základním tvarem (tzv. *lematem*) je nazýváno *lematizace* (*lemmatization*). V některých případech může být výhodné sjednocovat také všechna příbuzná slova (mající společný kořen) a proces hledání kořene je nazýván *stematizace* (*stemming*).

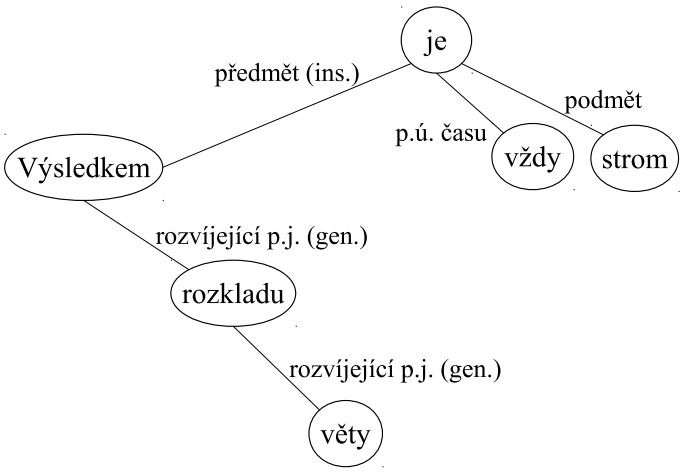
Pro účely stematizace i lematizace je možné vyhradit vyhledávací tabulku, která mapuje různé tvary slov na odpovídající kořeny nebo lemata. Manuální tvorba takové tabulky je však velmi pracná, zvláště ve flektivních jazycích. Z tohoto důvodu se hledání v tabulce kombinuje také se zkusným odtrháváním předpon a přípon. Ještě lepších výsledků dosahují systémy, které nejdříve provádějí *POS značkování* vstupních vět a značku každého slova použijí jako vodítko pro získání kořene či lematu.

4.1.3 Gramatický rozklad vět

Dolování sémantiky textu je možné usnadnit gramatickým rozkladem jednotlivých vět na *derivační stromy* (*parse trees*). V praxi jsou věty nejčastěji rozkládány *strukturální gramatikou* (*phrase structure grammar*) nebo *závislostní gramatikou* (*dependency grammar*).

Výsledkem rozkladu strukturální gramatikou je strom, jehož koncové uzly reprezentují jednotlivá slova věty. Nekoncové uzly reprezentují části věty, přičemž kořen představuje větu celou. Hrany vedoucí od rodičovských uzlů k synovským nemají žádné atributy. Příklad rozkladu ukazuje obrázek 4.1.

Rozklad věty závislostní gramatikou je zjednodušenou variantou, ve které se modelují jen závislosti mezi jednotlivými slovy. Synovský uzel vždy představuje slovo, které je závislé na slově reprezentované rodičovským uzlem.



Obrázek 4.2: Závislostní rozklad věty

Koncovými uzly jsou tedy jen ta slova, na kterých nezávisí žádné další slovo, zatímco kořen představuje slovo, na kterém (přímo či nepřímo) závisí všechna ostatní slova – sloveso. Závislostní rozklad je výpočetně jednodušší operací. Obrázek 4.2 ilustruje jednoduchý příklad rozkladu podle závislostní gramatiky.

4.1.4 Rozpoznávání pojmenovaných entit

Problém *rozpoznávání pojmenovaných entit* (*Named-entity recognition – NER*) spočívá v detekci výrazů v textu patřící pod určité kategorie. V praxi se jako kategorie definují např. jména osob, měst nebo institucí, ale také čísla, datum a čas, peněžní hodnoty a podobně.

Problém je typicky řešen rozdělením na dvě podúlohy: detekci slov označujících nějaké entity, a následnou predikci kategorií, do kterých daná slova spadají. Úloha může být řešena manuálním vytvořením gramatiky s dostatečně velkou sadu pravidel, ale její tvorba je velmi pracná. Další možností je využití strojového učení (většinou modelováno jako *podmíněné náhodné pole*), ale pro dosažení uspokojivých výsledků je nutné disponovat velkým množstvím označkovaných trénovacích dat.

V oblasti analýzy sentimentu se může na slova spadající pod stejnou kategorii nahlížet buď jako na synonyma nebo alespoň jejich příslušnost ke stejné kategorii nějakým způsobem zohlednit.

4.1.5 Tezaury a lexikony

Tezaurus je termín označující synonymický slovník. V tomto významu se někdy používá i termín *lexikon*. Skutečný význam lexikonu je však obecnější – je to libovolný seznam slov, která nemusí být nutně podobného významu. Tezaury i lexikony mohou být ve zpracování přirozeného jazyka užitečné v podobném významu jako pojmenované entity.

Speciální variantou je lexikon obsahující pozitivní a/nebo negativní slova. Ten může kromě samotného výčtu slov obsahovat také číselné vyjádření, jak

moc jsou daná slova pozitivní či negativní.

4.1.6 Stop slova

Za *stop slova* jsou v přirozeném jazyce považována taková slova, která jsou velmi frekventovaná, ale přitom nenesou (přinejmenším ne v daném kontextu) žádnou sémantickou informaci. Mezi tato slova patří například determinátory (*nějaký*, *ten*, *tamten*, *můj*, *tento*, aj.) nebo spojky. Stop slova jsou obvykle odstraněna před samotnou analýzou textu. Seznam vhodných stop slov (také označováno jako *stop seznam*, anglicky *stop list*) se však neliší pouze podle jazyka, ale také dle domény a cíle analýzy. Při použití nevhodné sady stop slov se může přesnost modelu také zhoršit.

4.1.7 Model multimnožiny slov

Multimnožina slov (anglicky *Bag-of-words*) je oblíbeným konceptem při modelování přirozeného jazyka. Jeho základní myšlenkou je zanedbání pořadí slov ve větě a gramatických konstrukcí. V úvahu tak připadají jen slova jako taková, jež se použijí jako binární příznaky. Vektor příznaků je pak v základní variantě tvořen tolika příznaky, kolik unikátních slov se nachází ve vzorcích.

Přesnost tohoto modelu je možné zvýšit generováním tzv. *n-gramů*. Termín n-gram označuje sousloví o n slovech. Vektor příznaků je rozšířen o všechny n-gramy nalezené ve vzorcích, čímž se s rostoucím n zvyšuje velikost vektoru. Pro většinu případů užití je výhodné generovat *unigramy* ($n = 1$) a *bigramy* ($n = 2$). S vyšším řádem se strojové učení kvůli rostoucímu počtu příznaků nejenom zpomaluje, ale také vykazuje nižší přesnost. Další možností je při generování n-gramů povolit přeskakování určitého počtu slov, čímž se mezi výslednými příznaky objeví i kombinace slov, které nejsou v původní sekvenci těsně za sebou a počet příznaků stoupne ještě více.

4.2 Reprezentace slov

Významnou oblastí výzkumu týkajícího se *zpracování přirozeného jazyka* je způsob reprezentace slov a jejich interpretace. Základní jednotka významu se v ligvistice označuje jako *morfém*. V některých jazycích (např. v čínštině) je každé slovo tvořeno jediným morfémem. Častěji jsou však slova složená z více různých morfémů. Jejich extrakce není jednoduchá, proto se v analýze sentimentu obvykle neprovadí a za základní jednotku významu je pokládáno celé slovo.

4.2.1 Distribuční vektory

Nechť V je množina všech slov nacházejících se v určitém korpusu C. *Distribuční vektor* slova $w_i \in V$ je vektor D_i o $|V|$ dimenzích, kde pro každé $w_j \in V$ je na j-tém indexu vektoru D_i (značeno D_{i_j}) pravděpodobnost, s jakou se slovo w_j nachází v maximální vzdálenosti k od náhodného výskytu slova w_i

v korpusu C. Distribuční vektor každého slova pak tedy reprezentuje rozložení pravděpodobnosti s jakou jsou jednotlivá slova slovníku V sousedy daného slova do vzdálenosti k. Platí tedy, že

$$\text{pro} \forall w_i \in V : \sum_{j=1}^{|V|} D_{i_j} = 2k.$$

Užitečnou vlastností distribučních vektorů je, že míra jejich podobnosti (kosinová podobnost) reflektuje podobnost kontextu, ve kterém se příslušná slova nachází a tím pádem i podobnost slov samotných. Rozložení pravděpodobnosti pro každý vektor D_i je možné odvodit z manuálního spočtení sousedních slov všech výskytů odpovídajícího slova w_i v daném korpusu C.

4.2.2 Vnoření slov

Distribuční vektory definové výše nejsou v praxi příliš použitelné. Důvodem jsou jejich prostorové ($|V|^2$, kde $|V|$ v praxi bývá v řádech sta tisíců až milionů) a výpočetní nároky.

Koncept *vnoření slov* (*word embedding*) spočívá v razantním omezení dimenze vektorů na hodnotu d, která se obvykle pohybuje v řádu stovek. Místo počítání sousedů je pro každé slovo $w_i \in V$ vytvořen vektor E_i s náhodnými hodnotami a s využitím strojového učení jsou vektory iterativně upravovány. V každé iteraci se vždy zjistí množina sousedních slov N_i do vzdálenosti k od zkoumaného slova w_i a vektory všech slov $N_i \cup \{w_i\}$ jsou upraveny tak, že jsou si navzájem podobnější než na začátku iterace a naopak vektory všech ostatních slov $V \setminus N_i \setminus \{w_i\}$ jsou upraveny tak, že jsou méně podobné vektoru E_i. Model byl poprvé popsán v [BDVJ03].

Přestože je jejich dimenze relativně nízká (oproti distribučním), mají tyto vektory vzhledem ke způsobu jejich vytvoření poměrně silnou vypovídající hodnotu o podobnosti slov, která reprezentují. Operace prováděné s vektory jsou díky nízké hodnotě d velmi rychlé.

Jednotlivá slova slovníku reprezentovaná svými vektory si lze také představit jako body v eukleidovském prostoru o d dimenzích, ve kterém je možné odvozovat zajímavé analogie. Bylo demonstrováno[MYZ13], že s dostatečně velkým korpusem lze úkoly typu „Slovo *Berlín* je ve vztahu r se slovem *Německo* a slovo x je ve vztahu r se slovem *Česko*. Najděte slovo x.“ řešit jednoduchým aplikováním vektorové aritmetiky:

$$E(\text{Berlín}) - E(\text{Německo}) + E(\text{Česko}) \approx E(\text{Praha}).$$

Pokud se na tento prostor aplikuje algoritmus k-means, je také možné získat množinu k skupin navzájem podobných slov, podobně jako u *Brownova shlukování* (viz kapitola 3.2.2).

Velkou výhodou techniky vnoření slov je skutečnost, že jde čistě o učení bez učitele. Pro všechny jazyky s nezanedbatelným počtem mluvčích existuje obrovské množství elektronického textu, které je možné použít pro vytvoření daných vektorů bez dodatečné manuální práce.

Model vnoření slov nabyl na popularitě po zveřejnění varianty *word2vec* [MCCD13] [MSC+13].

4.3 Detekce aspektových kategorií

V této části kapitoly budou představeny některé postupy, které se používají pro detekci aspektových kategorií. Každá aspektová kategorie je tvořena spojením konkrétní entity s nějakým jejím atributem, značeno `ENTITA#ATRIBUT`. Kategorie je možné detekovat atomicky, případně odděleně vyhledat entity a atributy a ty vzájemně pospojovat do aspektových kategorií.

Úlohu komplikuje několik skutečností. Aspekty mohou být ve větě explicitně vyjádřeny entitou i atributem (***výkon laptopu*** *je nedostačující*) nebo vyjádřeny jen entitou (***laptop*** *ztěží spustí středně náročné hry*). V některých případech nemusí být vyjádřena ani entita (*středně náročné hry si nezahrajete*). Ve všech těchto případech by však měl systém vrátit aspektovou kategorii `LAPTOP#VÝKON`. Někdy je atribut určen slovem, které se pojí s polaritou sentimentu (*notebook byl* ***levný*** $\rightarrow$ `LAPTOP#CENA`). Fráze označující aspekty však nemusí být vždy nutně součástí sentimentu (***cena laptopu*** *nehrála při výběru žádnou roli* – cena je zde zmíněna, ale autor k ní nevyjadřuje žádný názor).

4.3.1 Tématické modelování

Tématické modelování (*topic modeling*) je úlohou, která u textových dokumentů zjišťuje, o jakých tématech se v nich píše. Na základě označkovaných dokumentů je cílem pro každé téma spočítat frekcenci, s jakou se jednotlivá slova nachází v dokumentech zahrnující dané téma. Porovnáním frekvence jednotlivých slov v neoznačkovaném dokumentu s frekvencemi slov pro konkrétní téma lze odhadnout míru, s jakou dokument dané téma obsahuje.

Úloha se nejčastěji řeší algoritmem *pLSA* (*probabilistic Latent Semantic Analysis*) [Hof99] nebo *LDA* (*Latent Dirichlet Allocation*) [BNJ03].

Tématické modelování se často využívá pro porovnání dokumentů po obsahové stránce, kdy je například možné nabídnout čtenáři po dočtení konkrétního článku jiný článek s podobnou tématikou.

Detekci aspektových kategorií lze řešit jako tématické modelování, kde tématům odpovídají jednotlivé aspektové kategorie a dokumentům odpovídají věty, ve kterých se mají kategorie detekovat. Nevýhodou tohoto přístupu však je, že pro dosažení uspokojivých výsledků je nutné mít k dispozici velké množství trénovacích dat a věnovat velké úsilí ladění parametrů modelu.

4.3.2 Použití strojového učení

Text vstupní věty je také možné pokládat za samostatné příznaky a formou n-gramové *multimnožiny slov* řešit jako klasifikační úlohu. Text obvykle nejdříve prochází předzpracovací fází, kde je mu odebrána interpukce, velká písmena jsou převedena na malá a podobně. Tento přístup byl použit například v [BKS14].

V [ZWX15] byla k detekci použita jiná metoda. Slova vstupní věty se převedla na *word2vec* vektory, jejich zprůměrováním získán vektor reprezentující celou větu a ten se následně použil jako vstup *dopředné neuronovové sítě* se dvěma skrytými vrstvami. Výstupní neurony sítě reprezentovaly jednotlivé aspektové kategorie. Výsledky této metody však byly prezentovány jen pro velmi malý počet kategorií, přičemž každá kategorie odpovídala samostané entitě, atributy nebyly brány v úvahu.

4.4 Detekce cílů

Detekce aspektových kategorií měla ke vstupním větám přiřazovat značky, které jednoznačně identifikovaly, o jakých aspektech byl v daných větách vyjádřen názor, zatímco přesná fráze, jakou byl aspekt popsán, nebyla vyžadována. Detekce cílů je přesně opačný problém, kdy se mají v textu vyhledat fráze značící aspekty, přičemž kategorie aspektu není sama o sobě důležitá.

4.4.1 Použití sekvenčního značkování

Úloha detekce cílů je nejčastěji řešena jako sekvenční značkování. Cílové fráze, které jsou tvořeny posloupnostmi slov, se zakódují – jednotlivým slovům jsou přiřazeny značky indikující jejich vztah k cílovým frázím. V [SV99] byla zkoumána účinnost několika strategií:

- Schéma `IOB1` – Slovo je označeno `O`, pokud není součástí žádné cílové fráze; `B`, pokud je prvním slovem víceslovné fráze; `I`, pokud je součástí fráze, ale ne prvním slovem.
- Schéma `IOB2` – stejné jako `IOB1`, ale `B` jsou označena i slova tvořící jednoslovné fráze.
- Schéma `IOE1` – Slovo je označeno `O`, pokud není součástí žádné cílové fráze; `E`, pokud je posledním slovem fráze, po které hned následuje další fráze; `I`, pokud je součástí fráze, ale není `E`.
- Schéma `IOE2` – stejné jako `IOE1`, ale `E` jsou označena i slova, po nichž nenásleduje žádná další fráze.
- Schéma `[+]` – použity jsou dva značkovače. První značí `[`, pokud slovem začíná fráze, `.` v opačném případě. Druhý značkovač značí `]` a `.` v přesně opačném významu. Sekvence slov je detekována jako cílová fráze, pokud první slovo bylo označeno `[`, poslední `]` a všechna případná vnitřní slova byla oběma značkovači označena `..`
- Schéma `[+IO` – použity jsou dva značkovače, přičemž první značkuje stejně jako u schématu `[+]`. Druhý značkuje `I`, pokud je slovo součástí fráze, `O` v opačném případě. Slova s oběma značkami `[` a `I` jsou přeznačena na `B` a výsledek se interpretuje jako `IOB2`.

- Schéma `[+IO` – podobné schámatu `[+IO`, ale první značkovač značkuje `]` a `..` Slova s oběma značkami `]` a `I` jsou přeznačena na `E` a výsledek se interpretuje jako `IOE2`.

4.4.2 Detekce frekventovaných frází

Tato metoda spočívá ve vytvoření seznamu frekventovaných slov a sousloví z trénovací sady. Obvykle se slova nejdříve označkují slovními druhy a v úvahu poté vezmou jen podstatná jména. Vzniklý seznam je následně použit pro predikci cílů v neoznačkovaných větách.

4.5 Predikce polarity

Určení polarity sentimentu komplikuje mnoho faktorů přirozeného jazyka. Často může směr polarity ovlivnit přidání jediného slova do věty (*jsem s tím spokojena* → *jsem s tím ztěží spokojena*). Speciálním případem jsou negační slova, která někdy obrací polaritu sentimentu – *ne*, *nikdy* a jiné. Problém představují i způsobová slovesa (*vzhled je lepší, než jsem čekal* vs. *vzhled mohl být lepší*). Detekovat slova mající speciální gramatický význam sice není těžké, ale jejich interpretace pro konkrétní případ už ano. Komplikace způsobují i fráze typu *nedá se říct, že jsem s výrobkem spokojen*, kde část věty obrací směr polarity sentimentu v jiné části věty. Detekce polarity je také problematická ve srovnávacích frázích (*lepší mobil nenajdete* vs. *lepší mobil prodává konkurence*). Jednou z největších výzev je pak odhalování sarkasmu.

Úloha přiřazení polarity jednotlivým, předem detekovaným, sentimentům je v praxi řešena dvěma základními přístupy [Liu12]: klasifikace použitím strojového učení s učitelem nebo vyhledáváním slov v lexikonu polarit.

4.5.1 Použití strojového učení

Myšlenkou prvního přístupu je ke každému sentimentu detekovat jeho kontext (část věty), přetransformovat ho do vektoru příznaků (nejčastěji jako *multimnožinu slov*) a pro predikci polarity použít klasifikační algoritmus. Úspěšnost tohoto postupu je závislá na kvalitě detekce kontextu. Systém *UMDuluth* (viz kapitola 4.6.2) použil čárky a větné spojky pro rozdělení souvětí na jednoduché věty. Každá věta pak představovala jeden kontext. Další možností je použití rozkladu vět – v [JYZ+11] byl použit závislostní rozklad. V [BM09] bylo místo detekce kontextu provedeno vážení příznaků každého slova na základě vzdálenosti od cílového termínu. Nevýhodou řešení úlohy pomocí strojového učení je jeho neobecnost – model, který byl natrénován na určitou doménu, obvykle nedosahuje uspokojivých výsledků v jiných doménách.

4.5.2 Použití lexikonů polarit

Další přístup spočívá ve zjištění polarity jednotlivých slov věty pomocí lexikonu polarit a následně je učitou strategií určena polarita celková. Je možné

provést vážený součet polarit všech slov věty, který bere v úvahu vzdálenost jednotlivých slov od cílového termínu (použito v [DLY08]) nebo provést rozklad věty a v úvahu vzít pouze slova závislá na cílovém slovu. Polarizovaná slova jsou nejčastěji přídavná jména. I tento přístup ale s sebou nese určité problémy. Polarita některých slov se například mění v závislosti na kontextu, ve kterém se slovo nachází. Dalším problémem jsou slova, která někdy převrací směr polarizovaných slov. S výhodou se však dají použít slova, která někdy polaritu implikují. Je-li například na hranici dvou kontextů slovo *ale*, je vpravděpodobné, že odpovídající sentimenty budou mít opačné polarity.

Tento přístup je možné modelovat jako systém pravidel podobně jako ve formálních gramatikách. Pravidla však musí být pružnější co do pořadí slov (*donesli mi* ***výborné tofu*** vs. *jejich* ***tofu*** *bylo* ***výborné***). Pokud je systém dostatečně komplexní, lze s ním úspěšně řešit různé gramatické obraty. Nevýhodou je však pracnost takového řešení a poměrně silná fixace na konkrétní jazyk datových sad. Pokud je navíc systém modelován podle nevhodné nebo příliš malé sady, existuje riziko, že systém nebude mít uspokojivé výsledky v jiných sadách.

4.6 Existující řešení v SemEval 2015

Aspektově orientovaná analýza sentimentu se od roku 2014 stala jedním z úkolů mezinárodní soutěže *SemEval*. Pro účely této práce byly čerpány informace z publikací psaných účastníky *SemEval 2015* [PGP+15]. Jediným podporovaným jazykem datových sad byla angličtina. Zúčastněné týmy měly příležitost testovat své systémy na dvou doménách: recenze *restaurací* a *laptopů*. V kapitole 2.2 byly uvedeny jednotlivé úkoly, kterými se tato diplomová práce zabývá. Doména *restaurace* pokrývá úkoly `CAT`, `TAR`, `CAT+TAR` a `POL2`, zatímco doména *laptopy* pokrývá jen úkoly `CAT` a `POL1`.

V následujících podkapitolách budou ve stručnosti představeny systémy nejúspěšnějších týmů. Popsán bude jejich přístup jen k těm úlohám, jež se zúčastnily a v nichž dosáhly uspokojivé přesnosti.

4.6.1 Systém NLANGP

Tým *NLANGP*[TS15] dosáhl ve vyhodnocení prvního místa v úlohách `CAT`, `CAT+TAR` a druhého místa v úloze `TAR`. Datové vzorky tým reprezentoval *bigramovou multimnožinou*, kde pro každé slovo byly použity následující příznaky:

1. slovo jako takové,
2. slovníkový tvar,
3. výskyt slova na předem sestaveném seznamu,
4. příslušnost slova shlukům.

Seznam slov zmíněný v bodě 3 byl použit jen v doméně *restaurace*, kde byl extrahován z trénovací sady.

Tým experimentoval se dvěma různými shlukovacími modely, z nichž pvním bylo *Brownovo shlukování* a druhým shluky vytvořené z *word2vec* modelu aplikováním algoritmu *k*-means na vektorový prostor.

Detekci aspektových kategorií řešil tým metodou *one-vs.-all*, kde třídy odpovídaly jednotlivým aspektovým kategoriím. Jako klasifikační algoritmus byla zvolena *neuronová síť* s jednou skrytou vrstvou a čtyřmi skrytými jednotkami. Pro každou vstupní větu byla množina výstupních kategorií sestavena z těch kategorií, jejichž klasifikátory vrátily pravděpodobnost vyšší než předem stanovený práh.

Detekce cílů byla řešena jako úloha sekvenčního značkování se schématem `IOB2` (viz kapitola 4.4.1) a modelována jako *podmíněné náhodné pole*. V případě úlohy `TAR` se u cílů aspektová kategorie nerozlišovala. Pro získání predikcí v úloze `CAT+TAR` však ano. Tým poté jednoduše zkombinoval výsledky `CAT` a `TAR` úloh.

4.6.2 Systém UMDuluth

Systém týmu *UMDuluth* [KPRM15] nejprve natrénoval klasifikátor rozpoznávat, zda jednotlivé vstupní věty obsahovaly vůbec nějaký sentiment. Věty testovací sady, jež daný klasifikátor označil za beznázorové, se dalších fází modelu nezúčastnily. Na základě pozorování byly podle určitých pravidel věty rozděleny na více částí takovým způsobem, že každá část obsahovala právě jeden sentiment.

Pro získání aspektových kategorií byly všechny části vět podrobeny oddělené detekci entit a atributů použitím klasifikačního algoritmu *podpůrných vektorů*. Jednoduchými pravidly byly poté entity spojeny s atributy, čímž vznikly odpovídající aspektové kategorie.

4.6.3 Systém SIEL

Tým *SIEL* [GJV15] použil pro získání aspektových kategorií model *multimnožiny slov* a metodu *one-vs.-all* pro rozklad úlohy na binární klasifikaci, která byla následně řešena algoritmem *náhodný les*. Pro reprezentaci každé věty bylo použito následujících příznaků:

- všechna slova věty jako *multimnožina slov*,
- výskyt čísla ve větě (indikace *PRICES* atributu),
- výskyt slov nacházejících se na předpřipravených seznamech jídel a nápojů (indikace *FOOD* a *DRINKS* entit),
- všechna slova věty nahrazena za reprezentativní synonyma použitím tezaurů databáze *WordNet* [Mil95]. Takto upravená věta byla opět reprezentovaná *multimnožinou slov*.

4.6.4 Systém Sentiue

Systém týmu *Sentiue* [Sai15] dosáhl prvního místa v úlohách `POL1` a `POL2`. V úloze `CAT` dosáhl uspokojivého výsledku pouze v doméně restaurace.

V části předzpracování textu byla použita lematizace a POS značkování. Stejně jako v případě týmu *UMDuluth* byly entity a atributy v textu detekovány odděleně a poté jednoduchou metodou spojeny do aspektových kategorií. Klasifikace byla provedena algoritmem *maximální entropie*.

Také v úlohách detekce polarity sentimentu byl využit algoritmus *maximální entropie*. Každou instanci trénovacího příkladu tvořila

- celá předzpracována věta s použitím multimnožiny slov,
- doména trénovací sady,
- všechny vstupní anotace (entita, atribut, cíl),
- výskyt záporek ve větě,
- přítomnost otazníku nebo vykřičníku,
- výskyt polarizovaných slov na základě tří lexikonů polarit,
- přítomnost polarizovaného slova těsně před otazníkem nebo vykřičníkem,
- dvojslovo po každém slovese, záporce nebo polarizovaném slově,
- inverze polarity pokud je záporka před polarizovaným slovem,
- přítomnost polarizovaných slov v pěti posledních slovech věty.

4.6.5 Systém EliXa

Tým *EliXa* [nSVSA15] dosáhl prvního místa v úloze `TAR`. Úloha byla řešena jako *sekvenční značkování* s použitím algoritmu *perceptron*. Každé slovo bylo reprezentováno

- slovem samotným a jeho atributy (počáteční velké písmeno, číslice, interpunkce),
- předchozí predikce,
- první slovo věty,
- čtyři znaky předpony a přípony slova,
- příslušnost slova k *Brownovým*, *Clarkovým*[Cla03] a *word2vec* shlukům.

Kapitola 5
Návrh

Cíle této práce byly popsány v kapitole 2.2. Vzhledem k povaze zadání, kdy jsou vždy k dispozici trénovací data, je systém navržen s využitím strojového učení s učitelem. Návrh je založen na znalostech popsaných v kapitole 4.

5.1 Reprezentace vět

Textové příspěvky všech použitých sad budou nejprve rozděleny na jednotlivá slova a ke každému slovu (mimo interpunkci) bude zjištěn *slovníkový tvar* a *POS značka*. Text příspěvku bude rovněž předán rozpoznávači pojmenovaných entit a ke každému slovu bude případně připojena informace o tom, jakou entitu reprezentuje. Pomocí slovníkového tvaru každého slova se také zjistí polarita slova vyhledáním v lexikonu polarit.

Interně se tedy příspěvky budou reprezentovat posloupností objektů představujících jednotlivá slova a každý z nich bude obsahovat pět hodnot:

- text slova převedený na malá písmena,
- slovníkový tvar,
- POS značku,
- pojmenovanou entitu,
- polaritu slova.

Tato reprezentace bude mít obecné využití a teprve na základě typu úlohy budou ke každému slovu vybrány relevantní informace, které se použijí k řešení úlohy.

5.2 Shlukování slov

Reprezentace Brownových shluků pomocí binárního stromu byla vysvětlena v kapitole 3.2.2. Vytvoření této reprezentace bude pro konkrétní jazyk provedena pouze jednou a poté opakovaně využívána. Systém umožní nastavit

hodnotu n udávající počet úvodních znaků cesty, jež se použijí pro identifikaci shluků.

Princip vnoření slov, tedy jejich převedení na vektory byl popsán v kapitole 4.2.2. Na rozdíl od předchozích řešení nebude v této práci použito shlukování aplikováním metody k-means na vektorový prostor. Efektivita tohoto přístupu je totiž silně závislá na zvolení vhodné konstanty k a opakování algoritmu k-means pro mnoho různých hodnot by bylo nepraktické. Místo toho bude aplikován mírně odlišný postup. V přípravné fázi bude pro každou dvojici unikátních slov z trénovací sady zjištěna jejich podobnost a ta se posléze využije k vlastnímu způsobu shlukování.

Na začátku shlukovacího procesu se každé slovo z trénovací sady přiřadí svému vlastnímu shluku. Poté budou iterativně slučovány nejpodobnější shluky. Nechť $s(a, b)$ značí kosinovou podobnost vektorů slov a a b. Podobnost shluků $A = \{a_1, \ldots, a_n\}$ a $B = \{b_1, \ldots, b_m\}$ bude dána funkcí S, která vrací geometrický průměr podobnosti všech dvojic slov $(a, b) \in A \times B$:

$$S(A, B) = \sqrt[|A|\cdot|B|]{\prod_{a \in A} \prod_{b \in B} s(a, b)}.$$

Asymptotická časová složitost algoritmu je však kvůli hledání nejpodobnějších shluků $O(N^2)$, kde N je velikost slovníku. Pokud by v každé iteraci došlo pouze k jednomu sloučení, pro slovníky větší než cca 5000 slov by úloha nebyla řešitelná v rozumném čase. Z tohoto důvodu se v každé iteraci nevyhledá jen jedna dvojice nejpodobnějších shluků, ale obecně n dvojic. Některé z nich budou poté případně vyloučeny, aby se mezi zbylými dvojicemi nevyskytoval žádný shluk vícekrát. Nakonec se sloučí všechny výsledné dvojice shluků. Přestože řád složitosti se tím nesníží, doba výpočtu se zkrátí n-krát. Konstanta n bude tedy volena s ohledem na velikost slovníku. Důležitou vlastností tohoto postupu je, že třebaže se s rostoucí velikostí slovníku také zvyšuje hodnota n, kvalita vytvořených shluků by neměla být o moc horší. Čím větší totiž slovník je, tím více se v něm také vyskytují navzájem podobnější slova.

Algoritmus si u každého sloučení bude pamatovat podobnost daných shluků. Iterovat bude tak dlouho, než podobnost slučovaných shluků klesne pod určitou mez. Celý proces shlukování bude proveden pouze jednou pro všechny použité datové sady a jeho výsledek uložen pro pozdější použití.

Místo konstanty k značící počet shluků bude systém pracovat s nastavitelným parametrem udávajícím minimální podobnost shluků potřebnou k jejich sloučení. S ohledem na jeho hodnotu bude z výsledků přípravné slučovací fáze načtena odpovídající funkce mapující slova na shluky.

5.3 Parametry modelu

V kapitole 4.1 byly popsány techniky a nástroje, které při správném použití mohou vést ke zvýšení přesnosti modelu. V závislosti na doméně, úloze a dokonce i dílčí částí úlohy se však přínosnost jednotlivých technik může lišit. V této části kapitoly bude popsáno, jaké možnosti budou k dispozici

u jednotlivých technik, čímž bude definována množina parametrů modelu. Výběrem vhodných hodnot parametrů pro jednotlivé případy použití se bude věnovat kapitola 5.8.

5.3.1 Učící algoritmy

Systém bude u jednotlivých úkolů a jejich podčástí umožňovat výběr učícího algoritmu. Jejich volba bude dána typem úlohy – klasifikace nebo sekvenční značkování.

Pro získání přesnějších predikcí bude možné nastavit ztrátovou funkci a predikční práh. Systém také umožní přizpůsobení multimnožiny slov, konkrétněji specifikaci, jak velké n-gramy a s kolika vynecháváním se budou generovat.

5.3.2 Pojmenované entity

V kapitole 4.1.4 byl vysvětlen pozitivní přínos použití pojmenovaných entit. V závislosti na úkolu analýzy, doméně a jazyku je příslušnost slova k určité kategorii obecně různě relevantní. Pro jednotlivé kategorie bude systém nabízet tři možnosti, jak zacházet se slovy patřící k dané kategorii:

- ponechat slova v jejich původním tvaru,
- nahrazovat slova jménem kategorie (povede k nerozlišování mezi jednotlivými slovy),
- odstranit slova z vět.

5.3.3 Využití POS značek

POS značky budou využity podobným způsobem jako pojmenované entity. Důležitost slov z hlediska slovního druhu se obecně liší podle úkolu analýzy a domény. Slova stejné značky budou ponechána v jejich tvaru, odstraněna z věty nebo nahrazena jménem značky.

5.3.4 Shlukování slov

Pomocí parametru modelu bude možné určit, zda se jednotlivá slova vět mají ponechat v jejich původním tvaru nebo nahradit za reprezentanty Brownových nebo word2vec shluků.

5.3.5 Stop slova

V kapitole 4.1.6 byla vysvětlena technika odstraňování stop slov. Systém bude podporovat využití několika různých stop seznamů třetích stran, přičemž bude možné nastavit, které z nich se mají použít. Dále bude možné specifikovat vlastní stop seznam s nastavitelným obsahem.

5.3.6 Polarita slov

Lexikon polarit by měl být výhodný především v úlohách `POL1` a `POL2`, nicméně experimentováno bude s jeho využitím i v dalších úlohách. Použit bude lexikon obsahující číselné vyjádření polarity p, kde $p < 0$ představuje negativní a $p > 0$ pozitivní polaritu. Absolutní hodnota p indikuje hloubku polarity (*dobrý* vs. *excelentní*). K dispozici bude několik možností, jak přistupovat ke slovům, jejichž slovníkové tvary budou v lexikonu nalezeny:

- Všechna kladná slova se nahradí společným termínem `positive` a všechna záporná slova se nahradí společným termínem `negative`.
- Všechna polarizovaná slova se nahradí termínem reflektující číselné vyjádření jejich polarity – `positive_`p nebo `negative_`p.
- Všechna polarizovaná slova se nahradí společným termínem `polarized` – tento přístup odstraní polaritu, ale zachová informaci o tom, že slovo bylo polarizované. To by mohlo být prospěšné v úlohách `CAT`, `TAR` a `CAT+TAR`, kde polarizované slovo indikuje možný sentiment, ale vlastní polarita není důležitá.
- Odstraní se všechna polarizovaná slova.

Nezávisle na výběru možnosti bude také možné specifikovat, od jaké hodnoty $|p|$ budou slova pokládána za polarizovaná. Slova s menší hodnotou budou považována za neutrální.

5.3.7 Doménově specifické lexikony

Pojmenované entity mají nevýhodu ve své obecnosti. Pro konkrétní domény by bylo výhodné mít vlastní doménově specifické lexikony. Ty však nelze vytvářet zcela automatizovaně, a tak je jejich tvorba časově náročná. Součástí této práce bude vytvoření tématických lexikonů pro dvě domény (*restaurace* a *laptopy*) v anglickém jazyce a systém bude podporovat použití externích lexikonů. Slova spadající do stejného lexikonu budou buď nahrazována svým reprezentantem nebo odstraňována z vět, čímž vznikle možnost tvorby doménově specifických stop seznamů.

Počet použitých lexikonů nebude nijak omezen, a tak bude možné pro každou doménu vytvořit větší množství malých lexikonů pro přesnější pochopení sémantiky vstupních vět. Pro jednotlivé případy použití bude možné nejenom specifikovat, zda se konkrétní lexikon má použít, ale bude také možné některá slova lexikonu zastínit (nepoužívat pro konkrétní účel) a zjemnit tak případné negativní dopady výskytu nevhodných slov v daném lexikonu.

5.3.8 Ostatní jednoduché techniky

Systém bude schopný nad textem provádět také několik jednoduchých úprav:

- Slova obsahující pouze číslice volitelně nahradí slovem *number*. Přesná čísla obvykle nejsou sémanticky významná a jejich nahrazení za společné slovo by mělo zvýšit přesnost modelu.

- Pokud číslo tvoří dvojslovo se znakem některé měny, bude možné toto dvojslovo nahradit jedním slovem *price*. Tato technika by měla být prospěšná především v doménách, kde je cena součástí aspektových kategorií.

- Slova obsahující číslice i písmena budou volitelně nahrazena slovem *model*. Tato úprava by měla být prospěšná v doménách obsahující texty např. o výrobcích značené různými kódy (*i7*, *G73JH-x3*, *d620* a jiné).

- Dále bude nastavena minimální délka slov. Všechna slova obsahující menší počet znaků budou odstraněna.

- Sekvence dvou nebo více po sobě jdoucích stejných znaků budou volitelně v každém slově zkráceny na jeden znak. To by mohlo částečně řešit problém pravopisných chyb v uživatelských komentářích. Fráze typu *it was waaay to expensive* obsahují buď přebytečnou sekvenci (*waaay*) nebo naopak sekvence chybí (*to*). Tato technika však může sjednocovat i sémanticky odlišná slova (*to* vs. *too*), což v konečném důsledku může přesnost modelu také snížit.

5.4 Detekce aspektových kategorií

Úloha predikce aspektových kategorií bude modelována jako n-gramová multimnožina slov řešená klasifikačním algoritmem v kombinaci se sekvenčním značkováním. Třídám budou odpovídat jednotlivé aspektové kategorie. Úloha se pomocí *binární relevance* transformuje na binární klasifikaci. Každá aspektová kategorie bude z hlediska parametrů tvořit samostatný podproblém.

Každá věta bude transformována na klasifikační příklad aplikováním technik nastavených příslušnými parametry (definovaných v kapitole 5.3), jejichž hodnoty se budou obecně lišit podle toho, pro kterou aspektovou kategorii bude příklad vytvářen. Výsledkem každé tranformace bude řetězec, ve kterém mohou být jednotlivá slova původní věty odstraněna, nahrazena nebo pozměněna.

V trénovací fázi bude nejprve vytvořen samostatný klasifikátor pro každou aspektovou kategorii nalezenou v trénovacích datech. Poté se provede vytvoření klasifikačního příkladu každé trénovací věty zvlášť pro každou aspektovou kategorii a ty se předají odpovídajícím klasifikátorům. Příklad bude označen jako pozitivní právě tehdy, pokud daná věta obsahuje alespoň jeden sentiment s danou kategorií. V opačném případě bude označen jako negativní.

I v predikční fázi bude každá věta transformována zvlášť pro každou aspektovou kategorii. Všem klasifikátorům se předá odpovídající podoba věty a jako predikované budou vráceny všechny aspektové kategorie, jejichž

klasifikátory pro daný příklad vrátí pravděpodobnost vyšší než predikční práh dané kategorie.

Volitelnou částí predikční fáze bude dodatečné přidání některých aspektových kategorií do výsledné množiny. Pro každou větu a každou entitu se v dané větě zkusmo vyhledají jednoslovné cíle reprezentující danou entitu (popsáno v kapitole 5.5.1) a pokud se najde alespoň jeden takový cíl a žádná z predikovaných aspektových kategorií danou entitu neobsahuje, bude entita spojena s jejím nejčastějším atributem a vzniklá aspektová kategorie se přidá jako predikovaná. Tato technika však nebude aktivní při optimalizaci systému (kapitola 5.8), protože by svým opravným charakterem vedla k degradaci strojového učení.

5.5 Detekce cílů

Problém predikce cílů bude modelován jako sekvenční značkování. Na rozdíl od predikce aspektových kategorií se v této úloze s výhodou využije možnost specifikovat více příznaků k jednotlivým slovům věty:

- tvar slova převedený na malá písmena,
- pravděpodnostní hodnota reflektující použití počátečního velkého písmena,
- slovníkový tvar slova,
- POS značka,
- příslušnost k Brownovým a word2vec shlukům,
- případná pojmenovaná entita,
- polarita slova.

V případě word2vec shluků bude možné použít více příznaků reflektujících příslušnost slova ke shluku v závislosti na různých minimálních podobnostech shluků (popsáno v kapitole 5.2). Systém umožní nastavit následující parametry:

- jaká podmnožina výše uvedených příznaků je nejlepší,
- který algoritmus sekvenčního značkování bude použit a s jakými parametry,
- velikost kontextového okna – kolik slov před a kolik slov za aktuálním slovem brát v úvahu při učení.

Značka každého slova bude

- `S`, pokud slovo reprezentuje jednoslovnou cílovou frázi;

- `B`, pokud je prvním slovem cílové fráze;
- `E`, pokud je posledním slovem cílové fráze;
- `I`, pokud je slovo součástí cílové fráze, ale ne prvním ani posledním;
- `O`, pokud není slovo součástí žádné cílové fráze.

5.5.1 Vyhledávání jednoslovných cílů

Speciálním případem bude vyhledávání jednoslovných cílů. Tato úloha není součástí zadání práce, ale využije se jako podpůrná technika v úlohách `CAT` a `CAT+TAR`. Úkolem bude v testovací sadě najít pro každou větu jednoslovné cíle reprezentující určitou entitu. Vyhledávání cílů skládajících se pouze z jednoho slova je jednodušší než vyhledávání sekvencí, proto je možné očekávat vyšší přesnost.

Množina příznaků každého slova a množina parametrů učení budou obě shodné s těmi, které byly popsány v předchozí části kapitoly. V první fázi se trénovací sada použije zároveň také jako testovací a cílem bude najít takové parametry učení, pro které dává značkování nejlepší výsledky. Kritériem pro hodnocení kvality vyhledaných cílů bude funkce

$$Q(P) = \begin{cases} T_P & \text{pokud } \mathit{precision} \geq k \\ 0 & \text{jinak} \end{cases},$$

kde P představuje vektor parametrů a T_P a *precision* odpovídají definicím z kapitoly 3.4.2. Konstanta k bude minimální požadovaná správnost predikovaných slov a její hodnota by neměla být příliš nízká. Motivací je vyhledávat jen taková slova, u nichž je vysoká pravděpodobnost, že označují danou entitu. Zároveň by však konstanta k neměla být příliš vysoká, jinak bude opomíjena příliš velká část entit. Zoptimalizovaný vektor parametrů se poté použije na neoznačkovanou testovací sadu a pro každou větu tak vznikne seznam cílových slov.

5.6 Detekce aspektových kategorií a cílů

Úloha detekce dvojic (aspektová kategorie, cíl) bude, stejně jako v [TS15], řešena oddělenou detekcí obou podčástí a jejich následným spojením pomocí jednoduchých pravidel. Cíle aspektů však nebudou hledány odděleně pro všechny možné aspektové kategorie, ale jen pro jednotlivé entity. Tím se sníží množství značek a sekvenční značkování bude mít vyšší přesnost. Zároveň se ale ztratí informace o tom, jaký atribut byl s cílem spjat.

V rámci věty bude entita každého nalezeného cíle porovnávána s nalezenými aspektovými kategoriemi. Pokud se mezi nimi budou vyskytovat kategorie obsahující danou entitu, jedna z nich bude vybrána. Priorita výběru bude v první řadě dána tím, zda kategorie byla již přiřazena některému cíly (nepřiřazené budou mít přednost) a v druhé řadě frekvencí kategorie

v trénovací sadě (frekventovanější budou mít přednost). Nebude-li se mezi aspektovými kategoriemi vyskytovat žádná obsahující danou entitu, bude kategorie vytvořena spojením entity s jejím nejčastějším atributem.

Pokud po výše uvedeném postupu zůstane nepoužitá některá aspektová kategorie, pro její entitu se ve větě zkusí vyhledat všechny jednoslovné cíle podle postupu uvedeném v kapitole 5.5.1. Bude-li mezi nimi takový cíl, který není součástí žádného cíle nalezeného při klasické víceslovné detekci cílů, pak bude spojen s danou aspektovou kategorií. V opačném případě se pravděpobnost kategorie vrácená klasifikátorem z úlohy `CAT` porovná s nastavitelným prahem. Bude-li vyšší než tento práh, pak se aspektová kategorie spojí s prázdným cílem. V případě nižší pravděpobnosti se kategorie zahodí.

5.7 Predikce polarit

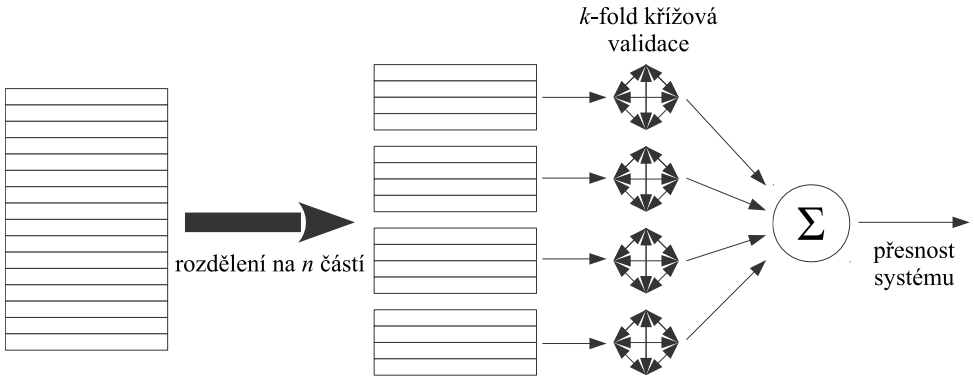
Predikce polarity sentimentu bude řešena jako klasifikační úloha. Na rozdíl od úlohy `CAT` však bude použit jediný klasifikátor. Mezi parametry systému budou přidány volby příznaků pro reprezentaci každého sentimentu. K dispozici bude

- entita, atribut, a odpovídající aspektová kategorie,
- sečtení normalizovaných polarit všech slov věty (pouze pokud není znám cíl),
- přítomnost otazníku nebo vykřičníku ve větě,
- normalizovaná polarita slova před vykřičníkem nebo otazníkem,
- součet normalizovaných polarit x posledních slov věty.

Pokud bude znám cíl, pak budou k dispozici také tyto příznaky:

- n nejbližších slov předcházejících cíli a m nejbližších slov následujících po cíli jako *multimnožina slov*,
- unigramy nacházející se těsně před a těsně za cílem,
- pro $\forall\, k \in N$: součet normalizovaných polarit slov nacházejících se přesně ve vzdálenosti k od cíle (dvojice slov – jedno před a jedno za cílem).

Parametry n, m, x a k bude možné nastavit. Normalizovaná polarita slova w je definována jako $\mathrm{neg}(w) \cdot \mathrm{sgn}(p) \cdot |p|^{\alpha}$, kde sgn je *funkce signum*, p je polarita slova w (dle definice z kapitoly 5.3.6), $\alpha \geq 1$ je nastavitelný parametr a $\mathrm{neg}(w) = -1$, pokud se před slovem w do vzálenosti k nachází některé ze slov obracejících polaritu, $\mathrm{neg}(w) = +1$ v opačném případě. Parametr k bude také nastavitelný. Množinu negačních slov bude možné specifikovat pro každý jazyk zvlášť.



Obrázek 5.1: Schéma měření přesnosti systému pomocí rozdělení trénovací sady na nezávislé části

5.8 Optimalizace modelu

Sada parametrů definovaná v kapitole 5.3 bude dávat uživateli systému široké možnosti ovlivnění výsledků analýzy. Díky tomu bude možné experimentovat s přesností systému při použití různých technik předzpracování textu a vlastnostmi strojového učení. Předpokládá se však, že ve většině případů se od systému bude chtít co nejvyšší přesnost analýzy dat bez nutnosti nastavování jednotlivých parametrů.

Optimalizace bude představovat nepovinnou fázi, ve které se model sám pokusí co nejvíce adaptovat na danou doménu a jazyk měřením svojí přesnosti s různými hodnotami parametrů. Tato část kapitoly popisuje návrh optimalizačního algoritmu, který bude v systému použit.

Algoritmus bude možné spustit pro ladění modelu určeného pro specifické domény, jazyky, datové sady a úlohy, které mají být řešeny. Úloha `CAT` bude optimalizována nezávisle pro každou aspektovou kategorii.

Trénovací sada se rozdělí na k stejně velkých částí, přičemž v každé z nich bude stejný poměr pozitivních a negativních vzorků (platí pouze pro úlohu `CAT`). Tyto části budou používány pro k-fold křížovou validaci (viz kapitola 7.1.1) s geometrickým průměrem, který penalizuje výkyvy mezi přesnostmi jednotlivých testů.

Opakování testů pouze nad touto jedinou datovou sadou by však vedlo k přeučení. Aby se tomu zabránilo, bude přesnost systému pro konkrétní hodnoty parametrů dodatečně měřena použitím dalších datových sad. Kopie trénovací sady se rozdělí na n stejně velkých částí, přičemž v každé z nich bude opět stejný poměr pozitivních a negativních vzorků, a k-fold křížová validace se bude provádět odděleně pro každou část, jak ilustruje obrázek 5.1. Kvůli nízkému počtu vzorků v jednotlivých částech však v tomto případě bude použit průměr aritmetický.

Parametry systému budou na počátku ladící fáze ve výchozím stavu. Optimalizační algoritmus bude postupně měnit jejich hodnoty a po každé změně změří přesnost systému. Aby mohla být nová konfigurace C prohlášena za lepší než stávající konfigurace C^*, musí platit všechny následující podmínky:

- k-fold křížová validace nad celou sadou není u konfigurace C nižší než u konfigurace C^*,
- součet výsledků k-fold křížových validací nad jednotlivými částmi sady není u konfigurace C nižší než u konfigurace C^*,
- alespoň jedna z předchozích podmínek musí platit i pro relaci *je větší než*.

Optimalizační rutina bude tvořena cyklem, ve kterém se bude postupně optimalizovat každý parametr související s optimalizovanou úlohou. Výjimku bude tvořit parametr udávající učící algoritmus. Ten je pokládán za tak zásadní, že ostatní parametry se budou optimalizovat zvlášť pro každou jeho hodnotu. Na konci procesu optimalizace bude jako hodnota nastaven ten učící algoritmus, pro nějž se nalezla nejlepší konfigurace ostatních parametrů. Každý parametr bude mít předem určenou množinu hodnot, jež může nabývat. Existovat bude pět typů parametrů:

- **celé číslo/výčet** – Definována bude spodní a horní hranice a algoritmus postupně vyzkouší všechny hodnoty v tomto rozmezí.
- **desetinné číslo** – Definována bude spodní a horní hranice a algoritmus vyzkouší k od sebe stejně vzdálených hodnot. Tento typ bude použit jen na hodnoty z krátkého intervalu. Hodnoty, jež mohou nabývat různých řádů, budou převedeny na exponenciální tvar a parametr řešen jako celé číslo.
- **pravdivostní hodnota** – Vyzkoušeny budou vždy obě hodnoty.
- **dvojice celých čísel** – Tento typ parametru bude použit v situacích, kdy dané hodnoty jsou na sobě závislé (např. jaké n-gramy a s kolika vynecháváním se budou generovat). Pro každou hodnotu bude definována spodní a horní hranice a algoritmus vyzkouší všechny kombinace hodnot v daných intervalech.
- **podmnožina řetězců** – Pro danou množinu řetězců bude množina všech jejích podmnožin tvořit obor možných hodnot. Protože se však nepředpokládá, že by řetězce byly na sobě závislé, algoritmus pouze zkusí do množiny jednotlivé řetězce přidat, případně je z ní odebrat.

Každá nalezená konfigurace, která bude lepší než stávající, se uloží do souboru. Pro každou podúlohu bude algoritmus iterovat tak dlouho, dokud budou nacházeny lepší konfigurace.

Kapitola 6
Implementace

V kapitolách 3 a 4 byl popsán teoretický základ pro aspektově orientovanou analýzu sentimentu a v kapitole 5 byl představen návrh systému schopného analýzu provádět. V této kapitole bude stručně popsána jeho implementace. S ohledem na dostupnost podpůrných knihoven byl k implementaci zvolen jazyk *Java*. Nemalé množství nástrojů je také dostupných pro jazyk *Python*.

6.1 Strojové učení

Vzhledem k dostupnosti velkého množství volně přístupných implementací algoritmů strojového učení nebyla v rámci této práce vytvořena žádná vlastní implementace. Využity byly knihovny *Vowpal Wabbit* a *CRFsuite*.

6.1.1 Vowpal Wabbit

Jednou z nejpoužívanějších knihoven na řešení problémů učení s učitelem je *Vowpal Wabbit* [LLS07] – velice rychlá open-source implementace *inkrementálního učení* šířená pod BSD licencí. Dostupné je vícero algoritmů, především

- lineární regrese,
- logistická regrese,
- podpůrné vektory,
- neuronové sítě s jednou skrytou vrstvou.

Pro klasifikační úlohy (`CAT`, `POL1` a `POL2`) umožňuje systém výběr všech výše uvedených algoritmů s výjimkou lineární regrese (není klasifikačním algoritmem).

V optimalizační fázi jednotlivých algoritmů používá Vowpal Wabbit vlastní modifikaci *stochastického gradientního sestupu*.

Metoda podpůrných vektorů byla ve Vowpal Wabbit implementována podle [BEWB05]. Je možné volit mezi třemi typy jader: *lineární*, *polynomiální* a *RBF*. Volba jádra byla přidána mezi parametry mého systému. V případě *polynomiálního* je také možné vybrat si stupeň 2 nebo 3. Když je ke klasifikaci

použita *neuronová síť*, systém umožňuje volbu mezi jednou skrytou jednotkou a deseti skrytými jednotkami.

S výjimkou *křížové entropie* jsou podporovány všechny ztrátové funkce uvedené v tabulce 3.1. Knihovna také umožňuje nastavení celé řady dalších vlastností, z nichž byly mezi parametry mého systému přidány následující:

- *learning rate* (souvisí s optimalizační fází strojového učení) a míra její degradace mezi jednotlivými průchody učení,
- λ_1 a λ_2 regularizace,
- použití *online stimulace* (*boosting*) [BKL15],
- *FTRL proximační optimalizace*[MHS+13].

Vowpal Wabbit je implementován v jazyce C++, ale pro několik programovacích jazyků (včetně Javy) existují nádstavby pro jeho použití.

6.1.2 CRFsuite

CRFsuite [Oka07] je velice rychlou implementací modelu podmíněného náhodného pole šířenou pod modifikovanou licencí BSD. K dispozici jsou následující algoritmy:

- L-BFGS (paměťově odlehčená verze *Broyden–Fletcher–Goldfarb–Shanno algoritmu* [Sha85]),
- stochastický gradientní sestup,
- průměrný perceptron [FS99],
- pasivní agrese,
- adaptivní regularizace váhových vektorů [CKD09].

V úloze `TAR` umožňuje systém volbu mezi všemi výše uvedenými algoritmy a také nastavení některých parametrů, jež jsou specifické konkrétním algoritmům.

6.2 Zpracování přirozeného jazyka

V této části kapitoly bude popsáno, jaké podpůrné nástroje a data pro zpracování přirozeného jazyka byly v systému využity.

6.2.1 Stanford CoreNLP

Stanford CoreNLP [MSB+14] je široce používanou sadou nástrojů pro zpracování přirozeného jazyka. Knihovna je šířena pod licencí GNU GPL a obsahuje velké množsví funkcí, z nichž byly v této práci využity následující:

- tokenizace textu,
- POS značkování,
- lematizace,
- rozpoznávání pojmenovaných entit.

Všechny tyto operace jsou značně časově náročné, proto jsou nad každou větou vykonány pouze jednou a jejich výsledek je pro pozdější použití uložen do souboru. Knihovna u jednotlivých operací podporuje jen některé jazyky. Pro podporu ostatních jazyků byla implementována vlastní tokenizace textu a jednoduchá stematizace. Využití stematizace je však podmíněno poskytnutím jazykového korpusu. Ten je totiž použit pro sestavení seznamu nejfrekventovanějších koncovek slov.

6.2.2 Word2vec

Word2vec[1] je populárním modelem vnoření slov. Pro anglický jazyk byla v této práci použita předpřipravená databáze 300-dimenzionálních vektorů, kde byla jako korpus použita část článků z Google News čítající asi 100 miliard slov[2]. Databáze čítá přibližně 3 miliony unikátních slov, kde se rozlišuje mezi jednotlivými tvary slov a to včetně počátečních velkých písmen.

Pro ostatní jazyky byla využita sada korpusů *MultiUN* [EC10]. Jednotlivé korpusy byly vytvořeny z dat stažených z webových stránek *Organizace spojených národů* a jsou dostupné jako volný open-source. Tabulka 6.1 zobrazuje velikosti korpusů, jež byly použity v této práci. Počty slov platí pro text extrahovaný ze sady XML souborů s odstraněnými slovy se znaky cizích abeced. Z textu byla také odstraněna interpunkce a všechna písmena převedena na malá.

Jazyk	Počet slov
Ruština	286 636 959
Španělština	359 307 580
Arabština	242 730 773
Francouzština	443 743 813

Tabulka 6.1: Velikosti MultiUN korpusů

Podobnost slov definovaná v kapitole 5.2 byla zjištěna použitím implementace `distance.cpp` z repozitáře projektu *word2vec*. Ukázku vytvořených shluků obsahuje příloha B.

[1] Domovská stránka: http://code.google.com/archive/p/word2vec/

[2] Soubor `GoogleNews-vectors-negative300.bin.gz` ke stažení na domovské stránce

6.2.3 Brownovo shlukování

V této práci byla použita C++ implementace *Brownových shluků* od Percyho Lianga[3]. Jako vstupní korpus pro anglický jazyk byla použita *Yelp sada*[4] obsahující asi 20 milionů slov. U větších korpusů byl bohužel problém s časovou složitostí algoritmu. Vzhledem k tomu, že word2vec implementace umožňovala rychlejší zpracování větších korpusů a ve výsledku tak pro systém představovala vyšší přínos, nebylo v případě Brownových shluků experimentováno s dalšími jazyky.

6.2.4 Lexikon polarit

Po experimentování s různými lexikony polarit pro anglické texty bylo nakonec rozhodnuto v systému ponechat pouze *AFINN lexikon* [Nie11], který dosáhl dobrého hodnocení v [KA15]. Lexikon obsahuje bezmála 2500 slov s polaritami od -5 do $+5$ a je šířen pod licencí GNU GPL. Jiné lexikony byly sice obsáhlejší co do počtu slov, ale k vyšší přesnosti systému nepřispěly.

Pro jiné jazyky nejsou dostupné prakticky žádné lexikony polarit, které by bylo možné s výhodou použít pro aspektově orientovanou analýzu sentimentu. Existující lexikony bývají orientované na běžnou analýzu sentimentu s přístupem sečtení polarit všech slov ve větě.

[3] Repozitář dostupný na http://github.com/percyliang/brown-cluster
[4] Stáhnutelná z http://github.com/acbart/WebInACan/tree/master/Yelper

Kapitola 7
Výsledky

V této kapitole budou demonstrovány výsledky implementovaného systému. Nejdříve bude vysvětlena technika měření přesnosti systému v jednotlivých úlohách a následně budou prezentovány samotné výsledky.

7.1 Metody měření přesnosti

Přesnost systému byla v úlohách `CAT`, `TAR` a `CAT+TAR` pro konkrétní běh (trénování a testování) měřena použitím tzv. *F-míry* (anglicky *F-score*, *F-measure*). Ta je definována jako harmonický průměr *precision* a *recall*:

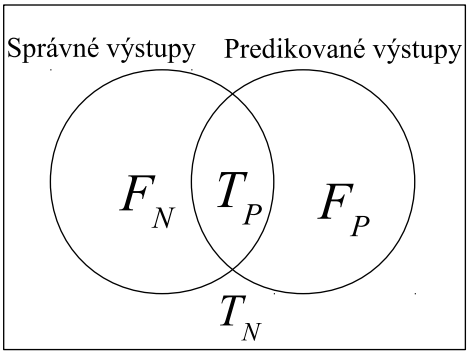
$$F = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}, \quad \text{kde} \quad P = \frac{T_P}{T_P + F_P} \quad \text{a} \quad R = \frac{T_P}{T_P + F_N}.$$

Systém pro každou větu vrátí množinu predikovaných výstupů (aspektových kategorií a/nebo cílů) a ta je následně porovnána s množinou očekávaných výstupů. Hodnoty T_P, F_P a F_N reprezentují celkové počty *správně pozitivních*, *falešně pozitivních* a *falešně negativních* výsledků přes všechny věty testovací sady. Vztah mezi množinami správných a predikovaných výstupů ilustruje obrázek 7.1. V úloze `TAR` jsou za správné považovány jen takové cíle, které se přesně shodují s očekávanými cíli. Pokud je například očekáván jeden dvojslovný cíl a místo toho systém predikuje obě slova jako samostatné cíle, při měření přesnosti se tato situace započítá jako $1 \times F_N$ a $2 \times F_P$.

V úlohách `POL1` a `POL2` vrací systém právě jednu polaritu pro každý sentiment testovací sady. Přesnost systému je pro konkrétní běh stanovena jako podíl počtu sentimentů, u kterých byla polarita správně predikována, ku celkovému počtu všech sentimentů.

7.1.1 Křížová validace

Výsledky jednoho konkrétního běhu však nejsou ze statistického pohledu příliš významné. V oblasti statistické analýzy je pro vyhodnocení přesnosti systémů obvykle používána jedna z forem *křížové validace* (*cross-validation*). Nejčastěji používaná varianta, tzv. *k-fold křížová validace*, spočívá v náhodném rozdělení všech dostupných označkovaných dat na k stejně velkých disjuktních podmnožin a vykonání k nezávislých testů, z nichž se poté vypočítá průměr. Každá



Obrázek 7.1: Vztah správných a predikovaných výstupů

podmnožina je právě jednou použita jako testovací sada, přičemž v každém testu jsou všechny zbývající podmnožiny spojeny do jedné trénovací sady. Tento postup je pro dosažení ještě přesnějších výsledků možné libovolněkrát opakovat a z jednotlivých výsledků spočítat průměr. Nejčastěji se používá průměr aritmetický, jehož výsledky velmi dobře reflektují průměrnou přesnost systému. Je-li ale např. důležitější zohlednit výkyvy přesnosti v závislosti na povaze testovaných vzorků, je možné použít průměr geometrický nebo harmonický.

Jednotlivé datové vzorky mívají různou povahu a systém je obecně různě úspěšný v jejich klasifikaci. Pokud by se k měření přesnosti používala stále stejná trénovací a testovací sada, vzorky by nemusely být do těchto sad vhodně rozděleny. Pokud by například v testovací sadě převládaly snadno klasifikovatelné vzorky, testy by vykazovaly vyšší přesnost, než jakou by systém skutečně měl. V případě provádění opakovaných optimalizací nad těmito sadami by navíc docházelo k přeučení. Křížová validace tyto problémy do značné míry redukuje opakováním testů pro všechny části.

7.2 SemEval 2016

V kapitole 4.6 byly popsány přístupy týmů, které se v roce 2015 zúčastnily mezinárodní soutěže *SemEval*. Pro demonstraci výsledků této diplomové práce byl systém přihlášen k úkolu číslo 5 (*aspektově orientovaná analýza sentimentu*)[1] aktuálního běhu *SemEval 2016* [PGP+16]. V době odevzdávání výsledků (leden 2016) však uměl systém řešit jen detekci aspektových kategorií v anglickém jazyce, proto se účastnil jen tohoto úkolu. Systém mohl běžet ve dvou různých módech:

- *omezeném*, kdy nebylo možné použít žádná externí data (externí lexikony, dodatečné trénovací sady atd.),
- *neomezeném*, kdy použití externích dat nebylo bráněno.

[1] Zadání úkolu je popsáno na http://alt.qcri.org/semeval2016/task5/

	Restaurace		Laptopy	
	O	**N**	**O**	**N**
1.	**71.494**	73.031	47.891	51.937
2.	68.701	72.886	**47.527**	49.105
3.	67.817	**72.396**	46.728	49.076
4.	67.350	71.537	45.629	**48.396**
5.	65.563	**71.494**	43.754	47.891

Tabulka 7.1: Výsledky nejúspěšnějších týmů v úloze detekce aspektových kategorií. F-míra jednotlivých týmů je uvedena v procentech a výsledky mého systému tučným písmem. O je zkratkou pro omezený a N pro neomezený mód běhu.

Restaurace
angličtina
nizozemština
francouzština
ruština
španělština
turečtina

Hotely	**Laptopy**	**Mob. telefony**	**Fotoaparáty**	**Telecom**
arabština	angličtina	čínština nizozemština	čínština	turečtina

Tabulka 7.2: Datové sady uvolněné v *SemEval 2016*

Tabulka 7.1 ukazuje výsledky nejúspěšnějších týmů v této části úkolu. Kompletní žebříček zobrazují tabulky A.1 a A.2. Pro hodnocení byla použita *F-míra*, která byla definována v kapitole 7.1. Popis mého systému je možné nalézt v [Mac16].

Odevzdaný systém však obsahoval manuálně vytvořené doménově specifické lexikony (popsáno v kapitole 5.3.7), které výrazně zvyšovaly jeho přesnost. Kvůli své neuniverzálnosti nebyly tyto lexikony používány v dalším vývoji systému.

Vzhledem k dostupnosti relativně velkého množství datových sad poskytnutých v SemEval 2016 (tabulka 7.2) nebyly pro účely testování této práce vytvořeny žádné vlastní sady.

7.3 Přesnost systému v jednotlivých úlohách

Přestože se systém nezúčastnil dalších úloh aspektově orientované analýzy sentimentu v soutěži *SemEval 2016*, jeho výsledky je možné dodatečně porovnat s výsledky přihlášených systémů. V této části kapitoly bude u jednotlivých úloh zobrazen žebříček nejúspěšnějších systémů a na něm ukázáno, jak by se můj systém ve vyhodnocení umístil. Ve všech tabulkách bude formou F-míry od každého systému zahrnut jen lepší výsledek z omezeného a neomezeného módu. Tabulky budou rovněž obsahovat přesnost systému změřenou 4-fold kří-

Dom.	Restaurace				Laptopy	Hotely
Jazyk	**Anglič.**	**Španěl.**	**Ruština**	**Franc.**	**Anglič.**	**Arab.**
1.	73.031	70.588	**70.849**	**65.723**	51.937	**52.580**
2.	72.886	**68.512**	64.825	61.207	**49.592**	52.114
3.	**72.396***	63.551	62.802	57.875	49.105	47.302
4.	71.537	61.968	62.689	53.592	49.076	
5.	**71.133**	61.370	39.601	49.928	**48.396***	
6.	70.869	59.899			47.891	
7.	68.701	58.810			47.196	
8.	68.203				45.629	
9.	68.108				43.913	
10.	67.979				43.754	
...	...				...	
Průměr	61.843	62.698	57.479	55.651	41.677	49.708
4-fold	**71.670**	**73.388**	**73.821**	**66.876**	**50.321**	**53.713**

Tabulka 7.3: Výsledky predikce aspektových kategorií. Hvězdičkou jsou označeny výsledky mého systému odevzdaného v oficiálním vyhodnocení.

žovou validací nad jednotlivými trénovacími sadami. Výsledky mého systému budou zvýrazněny tučným písmem.

Počet unikátních slov v datových sadách se u jednotlivých jazyků velmi liší. Nejméně obsahují anglické a francouzské sady, přibližně 4 tisíce, následují španělské s 6 tisíci, ruské s 9 tisíci a nakonec arabské s 21 tisíci.

Před samotným měřením přesnosti prošel systém optimalizační fází dle kapitoly 5.8, ve které mu pro každou doménu, jazyk, úlohu a případnou podúlohu byly nastaveny vhodné hodnoty parametrů.

7.3.1 Detekce aspektových kategorií

V úloze predikce aspektových kategorií bylo dosaženo uspokojivých výsledků. Jak je patrné z tabulky 7.3, v anglickém jazyce se po vylepšení systému, především využitím modelu vnoření slov, podařilo i bez doménově specifických lexikonů dosáhnout podobné přesnosti jako v oficiálním vyhodnocení. Za povšimnutí také stojí, že v doméně *restaurace* se ve všech třech jazycích dosáhlo přibližně stejné přesnosti.

Zatímco v doméně *restaurace* bylo definováno jen 12 aspektových kategorií, v doméně *hotely* to bylo již 34 a v doméně *laptopy* dokonce více než 80, přičemž některé z nich ani nebyly obsaženy v trénovací sadě. Velký rozdíl v počtu kategorií je důvodem, proč je dosažená přesnost v doméně *laptopy* nižší než v doméně *restaurace* o více než 20%.

Relativně nízkou přesnost v doméně *hotely* pravděpodobně způsobil velký počet unikátních arabských slov v datových sadách ku poměrně malému korpusu, jenž byl použit k vytvoření vektorů vnoření slov.

Z klasifikačních algoritmů byla nejúspěšnější logistická regrese. U většiny aspektových kategorií bylo nejvyšší přesnosti dosaženo použitím kvadrátové ztrátové funkce, v některých případech však byla lepší závěsová. U větší části

Dom.	**Restaurace**				**Hotely**
Jazyk	**Angličtina**	**Španěl.**	**Ruština**	**Francouz.**	**Arabština**
1.	72.340	68.515	**61.766**	66.667	**59.324**
2.	70.441	**68.401**	33.472	**66.617**	
3.	**67.654**	64.338		65.316	
4.	67.089	55.764			
5.	66.553				
6.	66.545				
7.	64.882				
8.	63.495				
9.	61.980				
10.	57.067				
...	...				
Průměr	58.217	62.872	33.472	65.992	
4-fold	**70.314**	**62.676**	**62.746**	**69.147**	**59.173**

Tabulka 7.4: Výsledky predikce cílů

kategorií bylo přínosné v rámci multimnožiny slov generovat jen unigramy a bigramy bez přeskakování. Odstraňování stop slov z textu se neukázalo jako moc přínosná technika. Ani použití pojmenovaných entit přesnost systému nezvýšilo, protože bylo prakticky zastíněno daleko užitečnějším modelem vnořených slov.

Přesnost predikce se u jednotlivých aspektových kategorií liší v závislosti na počtu trénovacích vzorků a charakteru kategorie. Zatímco u kategorií `JÍDLO#KVALITA` nebo `OBSLUHA#OBECNĚ` přesahuje 80%, u vágní a méně frekventované kategorie `RESTAURACE#RŮZNÉ` nedosahuje ani 25%. Přesnost v predikci velmi málo frekventovaných kategorií (např. `NÁPOJE#CENA`) je prakticky nulová.

7.3.2 Detekce cílů

I v úloze detekce cílů dosáhl systém dobré přesnosti, tabulka 7.4 ukazuje konkrétní výsledky. Doména *laptopy* nebyla v této úloze dostupná.

V ruském a arabském jazyce je přesnost systému znatelně nižší. Kromě problému velkého počtu unikátních slov v těchto jazycích může být nízká přesnost způsobená také absencí POS značkovače a lematizátoru nahrazeného za jednoduchý stematizátor. Experimentální vypnutí těchto nástrojů při analýze anglických a španělských příspěvků vedlo sice jen k mírnému poklesu přesnosti, ale ruština a arabština jsou vyšší mírou flektivními jazyky. V úloze `CAT` však bylo v ruském jazyce dosaženo dobré přesnosti, a tak může být slabší výsledek v úloze `TAR` způsoben i výběrem obtížněji klasifikovatelných vzorků v datových sadách. Kvůli absenci výsledků jiných systémů v těchto jazycích však nelze tuto domněnku jednoduše ověřit nebo vyvrátit.

Přesnost systému v doméně *hotely* je také negativně ovlivněna skutečností, že v datových sadách jsou cíle označovány bez určitých členů. Ty se ale v arabštině spojují s následujícím slovem, čímž se porušuje atomicita slov –

Dom.	Restaurace				Hotely
Jazyk	**Angličtina**	**Španěl.**	**Ruština**	**Francouz.**	**Arabština**
1.	52.607	**54.025**	**52.569**	**48.584**	**36.475**
2.	**51.644**	41.219	22.591	47.721	
3.	48.891				
4.	43.081				
5.	41.113				
6.	41.108				
7.	39.796				
8.	35.608				
9.	34.536				
10.	30.667				
...	...				
Průměr	38.584	41.219	22.591	47.721	
4-fold	**51.661**	**50.937**	**52.554**	**48.569**	**36.864**

Tabulka 7.5: Výsledky predikce dvojic (aspektová kategorie, cíl)

situace se kterou systém nepočítá. Tento problém by bylo možné odstranit vyjmutím členů před samotnou interpretací textu, ale to už by byla jazykově specifická optimalizace.

Ve španělském jazyce je patrný rozdíl v přesnosti mezi oficiálním *SemEval* testem a 4-fold křížovou validací nad trénovací sadou. Experimentálně bylo ověřeno, že tato skutečnost je způsobena tím, že oficiální *SemEval* testovací sada obsahuje snadněji klasifikovatelné vzorky

Nejvyšší přesnost značkování byla naměřena při použití algoritmu L-BFGS s kontextovým oknem dvě slova před a jedno slova za aktuálním slovem. Pro jednotlivá slova se všechny příznaky uvedené v kapitole 5.5 ukázaly jako přínosné.

7.3.3 Detekce aspektových kategorií a cílů

V úloze detekce dvojic (aspektová kategorie, cíl) dosahuje systém velmi dobré přesnosti v porovnání se systémy, které se zúčastnily této úlohy v *SemEval 2016*. Přesnost v jednotlivých jazycích ukazuje tabulka 7.5. Doména *laptopy* opět nebyla dostupná.

Za vysokou přesností v této úloze pravděpobně stojí vyhledávání cílů zvlášť jen pro jednotlivé entity, nikoliv pro všechny kombinace entit a atributů, přičemž atribut je následně odhadnut. Přesnost také zvýšilo použití jednoslovných cílů podle kapitoly 5.5.1, nicméně vzhledem ke způsobu jejich získávání nebyla tato metoda zahrnuta v testech křížové validace nad trénovacími datovými sadami. Při jejich použití byla přesnost systému uměla zvýšena o přibližně 10%. Výsledky křížových validací jsou tedy v této úloze o něco nižší než by bylo možné očekávat.

Ve španělském jazyce je pozorovatelný rozdíl mezi křížovou validací nad trénovací sadou a oficiálním *SemEval* testem. Rozdíl je zcela určitě způsoben

odlišnou přesností v úloze TAR.

Experimentováním bylo zjištěno, že pokud se aspektové kategorie, pro něž není nalezen žádný cíl, nikdy nespojují s prázdnými cíli, ale jednoduše zahodí, F-míra systému se sníží jen napatrně. Výrazně však vzroste rozdíl mezi *precision* a *recall*.

7.3.4 Predikce polarit

V úlohách predikce polarity sentimentu dosahuje systém spíše průměrných výsledků. V tabulce 7.6 jsou ukázány výsledky pro obě úlohy POL1, kde jsou vstupem pouze aspektové kategorie, a POL2, kde jsou vstupem dvojice (aspektová kategorie, cíl).

Z výsledků je možné usoudit, že znalost cíle byla sice v úloze predikce polarity sentimentu přínosná, ale ne závratně. Za povšimnutí také stojí, že výsledky 4-fold křížové validace nad trénovacími sadami jsou v angličtině nižší, než přesnost změřená při běžném testu použitím oficiální trénovací a testovací sady. To je způsobené rozdílným rozložením polarit v těchto sadách. Jak je patrné z tabulky 7.7, dominantní je polarita pozitivní. Z tohoto důvodu měly systémy tendenci častěji predikovat pozitivní polaritu, což při použití oficiální testovací sady vedlo k lepším výsledkům. V ruském jazyce je situace přesně opačná. V doméně *hotely* byl opět problém s některými cíli sentimentů.

V doménách a jazycích s nevyrovnaným poměrem polarit je možné přesnost systému dodatečně zvýšit posunutím rozhodovacího predikčního prahu. Tento trik však vyžaduje znalost odhadovaného poměru polarit v testovací sadě, a tak není zahrnut do výsledků zobrazených v tabulce 7.6.

Za slabšími výsledky v této úloze pravděpobně stojí nepoužití závislostního rozkladu vět k detekci relevantních přídavných jmen. Ačkoliv bylo vynaloženo jisté úsilí vyřešit rozklady vět pomocí knihovny *Stanford CoreNLP*, z technických důvodů nebylo tohoto cíle nakonec dosaženo.

Dom.	Restaurace				Laptopy	Hotely
Jazyk	**Anglič.**	**Španěl.**	**Ruština**	**Franc.**	**Anglič.**	**Arab.**
1.	88.126	83.582	77.923	78.826	82.772	82.719
2.	86.729	82.090	75.077	75.262	78.402	81.720
3.	85.448	81.343	73.615	73.166	78.152	**75.384**
4.	83.935	79.571	73.308	72.222	77.903	
5.	83.586	**76.026**	**70.846**	**68.332**	77.403	
6.	83.236				76.904	
7.	82.072				75.905	
8.	81.839				**75.281**	
9.	81.141				74.282	
10.	81.024				73.783	
11.	80.908				72.160	
12.	80.326				71.286	
13.	**79.977**				70.287	
14.	79.977				67.541	
15.	78.114				67.291	
16.	78.114				59.925	
...	...					
Průměr	79.034	81.647	74.981	74.869	73.600	82.220
4-fold	**76.464**	**77.014**	**76.655**	**68.344**	**71.950**	**75.890**

Tabulka 7.6: Výsledky predikce polarit. Doména laptopy neobsahuje cíle sentimentu jako vstupní anotace (`POL1`), ostatní domény ano (`POL2`).

Dom.	Restaurace								Laptopy		Hotely	
Jazyk	**Angl.**		**Špan.**		**Ruš.**		**Franc.**		**Angl.**		**Arab.**	
	tr	**te**	**tr**	**te**	**tr**	**te**	**tr**	**te**	**tr**	**te**	**tr**	**te**
pozitivní	66	71	71	70	76	67	46	46	56	60	59	58
negativní	30	24	25	26	17	25	48	46	37	34	35	36
neutrální	4	5	4	4	7	8	6	8	7	6	6	6

Tabulka 7.7: Rozložení polarit v datových sadách

Kapitola 8

Závěr

Výsledkem této práce je systém, jenž je možné použít pro aspektově orientovanou analýzu sentimentu krátkých textových příspěvků. Bylo demonstrováno, že jeho výsledky jsou poměrně dobré v porovnání s nejlepšími existujícími systémy. Slabšího výsledku bylo dosaženo pouze v predikci polarit sentimentů.

V návrhu systému se vycházelo ze znalostí z oblasti strojového učení a zpracování přirozeného jazyka a také z publikací, které byly zaměřeny na aspektově orientovanou analýzu sentimentu či příbuzná témata.

Velkou výhodou vzniklého systému je jeho schopnost pracovat v libovolných doménách a jazycích bez dodatečné manuální práce. Vzhledem k povaze analyzovaného jazyku však může systém dosahovat různé přesnosti. Bylo ukázáno, že v angličtině, ruštině i španělštině jsou jeho výsledky lepší než v arabštině. Pro zvýšení přesnosti analýzy příspěvků je užitečné systému dodat dostatečně velký jazykový korpus.

V dalším vývoji by bylo vhodné zaměřit se na vyzkoušení technik, jejichž implementace se v rámci diplomové práce nepodařila stihnout. Mezi ty patří použití závislostního nebo strukturálního rozkladu věty. Přesnost systému bude dále vylepšena použitím větších korpusů pro tvorbu vektorů modelu vnoření slov. Pro anglický jazyk se v této práci použil přibližně 300× větší korpus než u ostatních jazyků a rozdíl v kvalitě odpovídajících vektorů byl patrný. Vylepšený systém bude přihlášen do dalšího ročníku soutěže SemEval.

Literatura

[BdM+92] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai, *Class-based n-gram models of natural language*, Comput. Linguist. **18** (1992), no. 4, 467–479.

[BDVJ03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, *A neural probabilistic language model. 3: 1137–1155*, 2003.

[BEWB05] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou, *Fast kernel classifiers with online and active learning*, The Journal of Machine Learning Research **6** (2005), 1579–1619.

[BKL15] Alina Beygelzimer, Satyen Kale, and Haipeng Luo, *Optimal and adaptive algorithms for online boosting*, arXiv preprint arXiv:1502.02651 (2015).

[BKS14] Tomáš Brychcín, Michal Konkol, and Josef Steinberger, *UWB: machine learning approach to aspect-based sentiment analysis*, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 817–822.

[BM09] Erik Boiy and Marie-Francine Moens, *A machine learning approach to sentiment analysis in multilingual web texts*, Information retrieval **12** (2009), no. 5, 526–558.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan, *Latent dirichlet allocation*, the Journal of machine Learning research **3** (2003), 993–1022.

[CBHK02] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer, *Smote: synthetic minority over-sampling technique*, Journal of artificial intelligence research (2002), 321–357.

[CKD09] Koby Crammer, Alex Kulesza, and Mark Dredze, *Adaptive regularization of weight vectors*, Advances in neural information processing systems, 2009, pp. 414–422.

[Cla03] Alexander Clark, *Combining distributional and morphological information for part of speech induction*, Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1, Association for Computational Linguistics, 2003, pp. 59–66.

[Cyb89] George Cybenko, *Approximation by superpositions of a sigmoidal function*, Mathematics of control, signals and systems **2** (1989), no. 4, 303–314.

[DLY08] Xiaowen Ding, Bing Liu, and Philip S Yu, *A holistic lexicon-based approach to opinion mining*, Proceedings of the 2008 International Conference on Web Search and Data Mining, ACM, 2008, pp. 231–240.

[EC10] Andreas Eisele and Yu Chen, *Multiun: A multilingual corpus from united nation documents*, Proceedings of the Seventh conference on International Language Resources and Evaluation (Daniel Tapias, Mike Rosner, Stelios Piperidis, Jan Odjik, Joseph Mariani, Bente Maegaard, Khalid Choukri, and Nicoletta Calzolari (Conference Chair), eds.), European Language Resources Association (ELRA), 5 2010, pp. 2868–2872.

[FS99] Yoav Freund and Robert E Schapire, *Large margin classification using the perceptron algorithm*, Machine learning **37** (1999), no. 3, 277–296.

[GJV15] Satarupa Guha, Aditya Joshi, and Vasudeva Varma, *SIEL: Aspect Based Sentiment Analysis in Reviews*, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (Denver, Colorado), 2015.

[Hof99] Thomas Hofmann, *Probabilistic latent semantic indexing*, Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1999, pp. 50–57.

[JYZ+11] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao, *Target-dependent twitter sentiment classification*, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 151–160.

[KA15] Fajri Koto and Mirna Adriani, *A comparative study on twitter sentiment analysis: Which features are good?*, Natural Language Processing and Information Systems, Springer, 2015, pp. 453–457.

[KPRM15] Akshay Reddy Koppula, Ranga Reddy Pallelra, Ravikanth Repaka, and Venkata Subhash Movva, *UMDuluth-CS8761-12: A Novel Machine Learning Approach for Aspect Based Sentiment*

Analysis, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (Denver, Colorado), 2015.

[Liu12] Bing Liu, *Sentiment analysis and opinion mining*, Synthesis lectures on human language technologies **5** (2012), no. 1, 58–89.

[LLS07] John Langford, Lihong Li, and Alex Strehl, *Vowpal wabbit online learning project*, 2007.

[LMP01] John Lafferty, Andrew McCallum, and Fernando CN Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*.

[Mac16] Jakub Macháček, *BUTknot at SemEval-2016 Task 5: Supervised Machine Learning with Term Substitution Approach in Aspect Category Detection*, Proceedings of the 10th International Workshop on Semantic Evaluation (San Diego, California), SemEval '16, Association for Computational Linguistics, June 2016.

[MCCD13] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).

[MGZ04] Scott Miller, Jethran Guinness, and Alex Zamanian, *Name tagging with word clusters and discriminative training.*, HLT-NAACL, vol. 4, 2004, pp. 337–342.

[MHS+13] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al., *Ad click prediction: a view from the trenches*, Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013, pp. 1222–1230.

[Mil95] George A Miller, *Wordnet: a lexical database for english*, Communications of the ACM **38** (1995), no. 11, 39–41.

[MSB+14] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky, *The Stanford CoreNLP natural language processing toolkit*, Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60.

[MSC+13] Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, *Distributed representations of words and phrases and their compositionality*, Advances in neural information processing systems, 2013, pp. 3111–3119.

[MYZ13] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig, *Linguistic regularities in continuous space word representations.*, HLT-NAACL, 2013, pp. 746–751.

[Nie11] Finn Årup Nielsen, *A new ANEW: Evaluation of a word list for sentiment analysis in microblogs*, arXiv preprint arXiv:1103.2903 (2011).

[nSVSA15] Iñaki San Vicente, Xabier Saralegi, and Rodrigo Agerri, *EliXa: A modular and flexible ABSA platform*, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (Denver, Colorado), 2015.

[Oka07] Naoaki Okazaki, *CRFsuite: a fast implementation of Conditional Random Fields (CRFs)*, 2007.

[PGP+15] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos, *Semeval-2015 task 12: Aspect based sentiment analysis*, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 486–495.

[PGP+16] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit, *SemEval-2016 task 5: Aspect based sentiment analysis*, Proceedings of the 10th International Workshop on Semantic Evaluation (San Diego, California), SemEval '16, Association for Computational Linguistics, June 2016.

[Sai15] José Saias, *Sentiue: Target and Aspect based Sentiment Analysis in SemEval-2015 Task 12*, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (Denver, Colorado), 2015.

[Sch01] Bernhard Schölkopf, *The kernel trick for distances*, Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference, vol. 13, MIT Press, 2001, p. 301.

[Sha85] David F Shanno, *On broyden-fletcher-goldfarb-shanno method*, Journal of Optimization Theory and Applications **46** (1985), no. 1, 87–94.

[SV99] Erik F Sang and Jorn Veenstra, *Representing text chunks*, Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 1999, pp. 173–179.

[TS15] Zhiqiang Toh and Jian Su, *NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction*, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (Denver, Colorado), 2015.

[ZWX15] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao, *Representation learning for aspect category detection in online reviews.*, AAAI, 2015, pp. 417–424.

Příloha A

Podrobnější výsledky SemEval 2016

#	Jméno	Mód	F-míra
1	NLANGP	N	73.031
2	NileT.	N	72.886
3	**BUTknot**	**N**	**72.396**
4	AUEB-.	N	71.537
5	**BUTknot**	**O**	**71.494**
6	SYSU	N	70.869
7	XRCE	O	68.701
8	UWB	N	68.203
9	INSIG.	N	68.108
10	ESI	N	67.979
11	UWB	O	67.817
12	GTI	N	67.714
13	AUEB-.	O	67.350
14	NLANGP	O	65.563
15	LeeHu.	O	65.455
16	TGB	O	63.919
17	IIT-T.	N	63.051
18	DMIS	N	62.583
19	DMIS	O	61.754
20	IIT-T.	O	61.227
21	bunji	N	60.145
22	UFAL	N	59.300
23	INSIG.	O	58.303
24	IHS-R.	N	55.034
25	IHS-R.	N	53.149
26	SeemGo	N	50.737
27	UWate.	N	49.730
28	CENNL.	O	40.578
29	BUAP	N	37.290

Tabulka A.1: Úplné výsledky detekce aspektových kategorií v doméně *restaurace* a anglickém jazyku

#	Jméno	Mód	F-míra
1	NLANGP	N	51.937
2	AUEB-.	N	49.105
3	SYSU	N	49.076
4	**BUTknot**	**N**	**48.396**
5	UWB	O	47.891
6	**BUTknot**	**O**	**47.527**
7	UWB	N	47.258
8	NileT.	N	47.196
9	NLANGP	O	46.728
10	INSIG.	N	45.863
11	AUEB-.	O	45.629
12	IIT-T.	N	43.913
13	LeeHu.	O	43.754
14	IIT-T.	O	42.609
15	SeemGo	N	41.499
16	INSIG.	O	41.458
17	bunji	N	39.586
18	IHS-R.	N	39.024
19	UFAL	N	26.984
20	CENNL.	O	26.908
21	BUAP	N	26.787

Tabulka A.2: Úplné výsledky detekce aspektových kategorií v doméně *laptopy* a anglickém jazyku

Příloha B

Výsledky slučování podobných slov

Tato příloha obsahuje výčet některých zajímavých shluků slov vytvořených s pomocí word2vec modelu pro doménu restaurace a anglický jazyk. Minimální podobnost všech slov je v rámci stejného shluku 0.6 .

- fajitas, cheeseburgers, fries, sandwich, sandwiches, burgers, burritos
- seabass, yellowtail, halibut, mackerel
- tomatoes, peppers, onions, onion, artichoke, asparagus, tomato, chard, lettuce, eggplant
- chewy, creamy, flavorful, crispy, savory, tomatoey, crunchy, buttery, spicy, lemony
- delectable, yummy, delicous, delish, tasty, delicious
- fiance, mother, girlfriend, boyfriend, wife, husband
- son, sons, father, brother, daughter, sister
- awesome, incredible, fantastic, wonderful
- red, orange, pink, blue, yellow, purple
- disappointed, disapointed, dissappointed
- tuesday, friday, monday
- pricey, expensive
- nicely, beautifully
- eh, hmmmm, hmmm

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Jakub Macháček**

Studijní program: Otevřená informatika
Obor: Softwarové inženýrství

Název tématu: **Aspektově orientovaná analýza sentimentu**

Pokyny pro vypracování

1) Prostudujte existující metody aspektově orientované analýzy sentimentu
2) Seznamte se s dosavadními výsledky v oblasti automatického zpracování přirozeného jazyka, které je možné použít k získání relevantních příznaků pro aspektově orientovanou analýzu sentimentu.
3) Shromážděte data potřebná pro průběžné testování jednotlivých fází řešení problému
4) Navrhněte a implementujte systém, který dokáže provádět aspektově orientovanou analýzu sentimentu.
5) Vyhodnoťte výsledky systému na reprezentativním vzorku dat.

Seznam odborné literatury

1. Bing Liu: Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012
2. Projekt MixedEmotions, http://mixedemotions-project.eu/about/mixedemotions

Vedoucí: Ing. Božena Mannová, Ph.D.

Platnost zadání: do konce zimního semestru 2017/2018

L.S.

prof. Dr. Michal Pěchouček, MSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 3. 3. 2016