VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

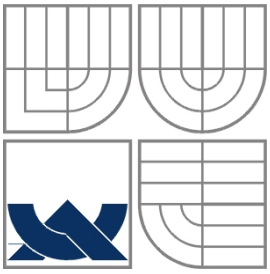
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# ZJEDNOZNAČŇOVÁNÍ SLOVNÍCH VÝZNAMŮ

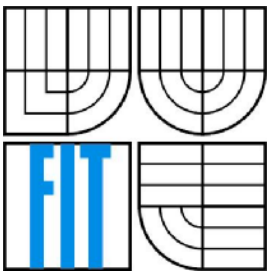
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. MICHAL KRAUS
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# ZJEDNOZNAČŇOVÁNÍ SLOVNÍCH VÝZNAMŮ
WORD SENSE DISAMBIGUATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. MICHAL KRAUS

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

BRNO 2008

# Zadání práce

## Zjednoznačňování slovních významů

## Word Sense Disambiguation

**Vedoucí:**

Smrž Pavel, doc. RNDr., Ph.D., UPGM FIT VUT

**Zadání:**

1. Seznamte se s metodami zjednoznačňování slovních významů.
2. Implementujte systém, který v textu vhodným způsobem označí nejpravděpodobnější význam slova v daném kontextu (např "jeřáb - stroj", "jeřáb - pták", nebo "jeřáb - strom").
3. Vytvořte testovací sadu pro vyhodnocení systému.
4. Vyhodnoťte vytvořený systém pomocí standardních metrik.

**Část požadovaná pro obhajobu SP:**

Bez požadavků.

**Kategorie:**

Umělá inteligence

# Licenční smlouva

Licenční smlouva je součástí vytištěné a svázané práce, která je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

## Abstrakt

Diplomová práce je zaměřena na rozpoznávání a zjednoznačňování českých slov. Nejprve se čtenář seznámí s historickým kontextem úkolu, poté jsou mu předvedeny použité algoritmy: naivní Bayesův klasifikátor, klasifikátor AdaBoost, metoda maximální entropie a rozhodovací strom. Použité metody jsou názorně předvedeny na příkladu. V dalších částech práce jsou popsány datové sady a parametry pro klasifikaci. V závěrečné části práce dojde na zhodnocení výsledků a nastínění možných úprav.



## Klíčová slova

zpracování přirozeného jazyka, český jazyk, zjednoznačňování slovních významů, klasifikační algoritmy, naivní Bayesův klasifikátor, klasifikátor AdaBoost, metoda maximální entropie, rozhodovací stromy



## Abstract

The master's thesis deals with sense disambiguation of Czech words. Reader is informed about task's history and used algorithms are introduced. There are naive Bayes classifier, AdaBoost classifier, maximum entrophy method and decision trees described in this thesis. Used methods are clearly demonstrated. In the next parts of this thesis are used data also described. Last part of the thesis describe reached results. There are some ideas to improve the system at the end of the thesis.



## Keywords

natural language processing, Czech language, word sense disambiguation, classifiers, naive Bayes classifier, AdaBoost classifier, maximum entrophy method, decision trees

## Citace

Kraus Michal: Zjednoznačňování slovních významů. Brno, 2008, diplomová práce, FIT VUT v Brně

# Zjednoznačňování slovních významů

## Prohlášení

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

Michal Kraus
19. května 2008

## Poděkování

Děkuji Doc. RNDr. Pavlu Smržovi za zadání diplomové práce, poskytnutí informací k vypracování a za data potřebná ke zdárnému úspěchu práce.

# Obsah

# Kapitola 1

# Úvod

Jazyk lidí, přirozený jazyk, je člověkem používán denně a rutině. Lidé spolu mluví, čímž si sdělují informace. Toto předávání informací je na rozdíl od jazyku počítačů, jako jsou např. programovací jazyky, značně neformální a jen velmi těžko se na něj dají aplikovat nástroje, které jsou použitelné pro jazyky umělé. Vágnost a neformálnost přirozeného jazyka vede k mnoha problémům, zvláště pokud se pokoušíme tímto jazykem komunikovat s počítačem. Navíc my, lidé, většinu problémů, které musí být vyřešeny (a ještě ani zdaleka nejsou), nevnímáme, protože je řešíme intuitivně. Úlohy spojené s komunikací člověk – počítač řeší hned několik dnešních oborů spjatých s informačními technologiemi. Jednak se jedná o umělou inteligenci, která poskytuje nástroje k řešení a na druhé straně stojí lingvistika, která dává řešení mantinely. Jedním z těchto problémů je zjednoznačňování slovních významů (word sense disambiguation, zkráceně WSD), který bude řešen v této diplomové práci.

Obor zpracování přirozeného jazyka má již za sebou nemalý vývoj. Hlavními úkoly tohoto oboru jsou např. strojový překlad mezi dvěma jazyky, získávání informací z textů, oprava gramatických chyb či “inteligentní konverzace” s počítačem. Tato část oboru pracuje ve většině případů s psaným textem (v libovolné formě). Dále se ke zpracování přirozeného jazyka může řadit i rozpoznávání mluvené řeči, tedy práce se signálem.

Abychom byli schopni efektivně tyto úlohy řešit, je nutné, aby počítač přirozenému jazyku rozuměl nejen po gramatické stránce (což se již víceméně povedlo), ale i po stránce obsahové. Způsob, jakým naučit počítač jazyku porozumět, může být např. báze znalostí (např. sémantická síť), které obsahuje závislostní vazby mezi jednotlivými slovy. Tyto sítě bývají v dnešní době stále plněny ručně, což je časově i zdrojově náročné. Právě tato náročnost vede ke snaze tyto znalosti co nejvíce automatizovat.

Pro svou práci jsem zvolil metodu několika úhlů pohledu. Snažil jsem se nalézt různé klasifikátory, které mohou řešit zvolený problém. Mezi ně patří např. naivní Bayesův klasifikátor, který je s úspěchem používán právě v oblasti zpracování přirozeného jazyka, dále klasifikátor AdaBoost, jehož hlavní doménou je sice počítačová grafika, ale díky jeho obecnosti je použitelný i pro WSD. Dalším klasifikátorem mohou být i různé pravděpodobnostní modely, mezi ně se řadí metoda největší entropie, která se uplatňuje např. ve fyzice. Neposlední možností je použití rozhodovacího stromu. Tyto klasifikátory budou v textu dále popsány a vyhodnoceny.

Práce je členěna následovně: Druhá kapitola popisuje úvod do problematiky WSD, zachycuje historické aspekty úkolu a nastiňuje možnosti řešení. Třetí kapitola je věnována popisu metod, které připadají do úvahy při řešení tohoto problému. Zvláštní

důraz je věnován metodám, které byly použity při řešení práce. Čtvrtá kapitola popisuje vlastní implementaci systému pro WSD a jsou zde řešeny související problémy, které vyvstaly při implementaci. Pátá kapitola obsahuje zhodnocení dosažených výsledků a metody, které byly použity při vyhodnocení. Nechybí ani návrhy na vylepšení systému. Závěrečná kapitola pak pouze stručně shrnuje výsledy práce.

# Kapitola 2

# Úvod do problematiky zjednoznačňování slovních významů

V této kapitole si vymezíme problém zjednoznačňování slovních významů. Také si zde uvedeme základní pojmy a historický kontext problému WSD.

Základním cílem je určení správného významu vícevýznamového slova (homonyma) v určitém kontextu přirozeného jazyka. Pro lidské myšlení je tento problém zpravidla triviální. Naopak je tomu u počítače. Aby počítač přirozenému jazyku porozuměl, je nutné najít vhodné vlastnosti, podle kterých by počítač významu porozuměl, a to s co nejlepší úspěšností. Pokud naučíme počítač význam interpretovat, dokážeme následně i rozlišit významy a tím i zjednoznačnit.

## 2.1 Vymezení pojmů

Pro porozumění dalšího textu je nezbytné, aby byly vysvětleny některé pojmy, které budou v dalších částech práce použity. Zaměříme se na pojmy týkající se významu slova, neboť tohoto se bude týkat celá práce. Informace v této části pocházejí z [2] a [3].

Pojem “význam slova či sousloví” (dále jen slovo) můžeme chápat jako informační obsah slova. Na tento obsah se dá nahlížet ze dvou pohledů:

- slovní (lexikální) význam - označuje samotnou skutečnost, tedy co slovo znamená
- mluvnický (gramatický) - tento význam zpravidla lze určit v závislosti na ostatních slovech ve větě či kontextu; jedná se o mluvnické kategorie (osoba, číslo, pád, slovní druh a další)

Ve své práci se budu zabývat výhradně slovním významem. Mluvnický význam je pro moji práci téměř nepoužitelný, jelikož často vychází z významu slovního, který se snažím určit. Budeme-li definovat slovní význam slova, použijeme zpravidla nějaký jazyk. A to buď:

- jazyk přirozený - snažíme se dané slovo nějakým vhodným způsobem popsat za pomocí již známých slov (Např. osobní auto bychom mohli popsat způsobem: “Má čtyři kola, je zhruba čtyři metry dlouhé, metr a půl vysoké…”). Tímto způsobem jsme schopni vysvětlit lexikální význam ostatním lidem, pro počítač je však tento způsob zpravidla nepoužitelný.
- formální jazyk - snažíme se dané slovo popsat vhodným matematickým nástrojem, jako jsou gramatiky, či různé kalkuly. Ačkoli toto je přesně způsob, který počítač dokáže snadno zpracovat, narazíme zde na druhé straně:

Matematický způsob vyjadřování je většině lidí cizí a stává se tak nepoužitelným pro vysvětlení lidem.

Další způsob vysvětlení významu je tzv. ostentativní. Je založen na ukázce (např. “toto je auto”,…). Tímto způsobem se učí významu slov zpravidla děti. Ve většině jazyků nastávají případy, kdy jedno slovo může mít významů více. V takovém případě mluvíme o slově víceznačném nebo vícevýznamovém slově. Význam u takových slov lze rozdělit na:

- základní (primární, původní)
- druhotný (sekundární, odvozený)

Příkladem může být třeba slovo ucho, kde primárním významem je sluchový orgán a sekundárním významem je např. ucho u hrnce, u kterého vzniklo takové pojmenování na základě podobnosti.

Každý přirozený jazyk má různý podíl mnohovýznamových slov. Pro češtinu je tento podíl poměrně malý a zpravidla nedochází k nedorozumění při jejich použití. Naopak k jazykům s velkým výskytem vícevýznamových slov se řadí angličtina, francouzština nebo čínština. S vícevýznamovostí souvisí i pojmy polysémie a homonymie. Pojem polysémie značí mnohoznačnost, neboli existence více významů pro jedno dané slovo, kde významy slova mají jistou podobnost (např. již zmiňované ucho). Homonymie je naopak existence více významů pro jedno slovo, ale mezi významy není žádná spojitost. Shoda hlásek je čistě náhodná (např. role - v divadelní hře, kus pole či svitek). Homonymii můžeme dále dělit na:

- lexikální - význam slova je závislý na kontextu. Mějme například tuto (hypotetickou) větu: "Mezi plesy letošní sezony nebyly vidět velké rozdíly v úrovni." Slovo „plesy“ umožňuje (teoreticky) dvojí interpretaci: „ples“ a „pleso“ (jezero).
- morfologickou - jedná se o homonymii vzniklou při skloňování či časování. Koncovky některých pádů mohou nést různé významy a připouštět různou interpretaci. Např. ze samotné věty “Autobus předjíždí tramvaj.” nepoznáme, zda podmět je slovo “autobus” či “tramvaj”, což samozřejmě ovlivní význam celé věty.
- slovně druhovou - význam slova je závislý na slovním druhu daného slova. Např. slovo večer může být jak podstatné jméno (označení části dne), tak příslovce (odpověď na otázku “Kdy přijdeš?”).

Morfologickou a slovně druhovou homonymií se ve své práci zabývat nebudu. Významy slov v těchto druzích homonymie mají k sobě velmi blízko.

Další možnosti víceznačnosti slov se týkají výslovnosti. Jedná se o homofonii (slova, jež se stejně vyslovují - časté např. v angličtině - two, to, too), kterou se v práci zabývat nebudu.

Pokud se budeme zajímat o lexikální homonymii, budeme chápat problém WSD jako výběr správného významu slova v daném kontextu.

## 2.2 Úvod do historie zjednoznačňování slovních významů

O automatické zjednoznačňování slovních významů se zajímali počítačoví odborníci hned od začátku "výpočetního věku", tedy přibližně od roku 1950. Jedná se o AI-úplný problém (tedy o problém, který je řešitelný algoritmy spadající do kategorie umělé inteligence). K řešení mohou existovat dva základní přístupy využívající odlišné techniky.

První možností je snaha o porozumění zpracovávaného textu. K tomuto se zpravidla používá nějaká báze znalostí, většinou tvořená ručně. Báze znalostí obsahuje tvary slov, jejich významy, možné kombinace, příklady použití, synonyma a vazby mezi významy.

Druhý způsob využívá obrovské množství dat, které analyzuje a vyhledává v nich souvislosti, jež se snaží nějak dále zpracovat. Zde se používají algoritmy strojového učení. Může se jednat o strojové učení s učitelem, bez něj, případně nějaká kombinace. Systém, založený na tomto způsobu, vytváří pravidla, která automaticky generuje ze vstupních dat - velkého kontextu. Použitelné metody jsou různé klasifikační a shlukovací algoritmy.

V následujícím textu jsou popsány významné kroky, které v historickém kontextu přispěly k řešení problému WSD. Přehled si nedělá nároky na kompletnost, podrobné informace o vývoji nalezne čtenář v [1].

## 2.3 Přehled historie WSD

V prvopočátcích se problém WSD spojoval se strojovým překladem. Koncem 40. let 20. století mnoho počítačových odborníků, zabývajících se strojovým překladem, tvrdilo, že k úspěšnému překladu je porozumění textu nezbytné. Tento fakt se týkal zvláště slov, která mají ve zdrojovém jazyce více významů.

Během následujícího desetiletí se přišlo na fakt, že je-li k překladu poskytnut pouze omezený kontext (čtyři slova), není úspěšnost významně odlišná od překladu, kterému je poskytnut celý text. Této vlastnosti se využívá dodnes.

Kolem roku 1965 byl ke zjednoznačnění textu poprvé použit Bayesův teorém. Snaha byla odhadnout četnosti jednotlivých významů v určitém kontextu a z těchto četností poté vyvodit závěr pro zkoumaný text. Přesnost se pohybovala na hranici 90%. Bohužel v této době nebylo dostatek zdrojů k trénování takových klasifikátorů, a tedy výsledek to byl ojedinělý a neúplný.

Začátkem 60. let 20. století se do povědomí dostává umělá inteligence. Její růst pokračoval i v 70. letech. Snaha o uplatnění tohoto oboru měla za důsledek vytvoření metod pro WSD založené na umělé inteligenci. Metody se snažily porozumět textu a modelovat lingvistické teorie za pomocí sémantické sítě, většinou ručně tvořené. Použitelnost těchto metod byla však výrazně omezena. To bylo dáno faktem, že vytvořené znalostní báze pokrývaly stále jen část přirozeného jazyka.

Pozdější dobou byly stále dostupnější významové slovníky jako báze znalostí. Stále byly tvořeny ručně, ačkoliv byla snaha o jejich automatické vytváření.

Významným krokem bylo vytvoření slovníku anglického jazyka v roce 1990, nazvaného WordNet. Tento slovník patří k výčtovým lexikonům, tj. obsahuje veškeré tvary slova. Je tvořen stromy, kde kořen je tvar slova a listy jsou jednotlivé významy. Vývoj WordNetu neustále probíhá, v současnosti je dostupná verze 3.0. Čeština je jeden z osmi jazyků, která má svůj vlastní WordNet. Jeho velikost je zhruba 30 000 významových sémantických řad.

Opakem výčtového lexikonu je generativní lexikon, který obsahuje množinu pravidel, podle kterých jsou jednotlivé významy generovány.

Rozvojem počítačových technologií se do popředí dostává odlišný přístup k WSD, založený na obrovských, v drtivé většině ručně anotovaných, zdrojích dat - korpusech. Díky obsáhlosti dat je možné tyto korpusy použít pro statistické vyhodnocování přirozeného jazyka. Korpusem je myšlen ucelený soubor textů daného jazyka v elektronické podobě. Mohou obsahovat pouze holý text, nebo případně i další metadata, která obsahují informaci o korpusu se zřetelem na jeho použití. Tato metadata jsou přidávána stále ručně, ačkoliv se projevila snaha o automatické anotování korpusu. Jedná se o velice nákladnou práci, vzhledem k rozsáhlosti korpusů. První korpus byl vytvořen v roce 1964 pro angličtinu. Pomohl odhalit statistické charakteristiky slov (četnosti výskytů, slovní druhy,…) v angličtině. V současné době má již mnoho států vytvořený vlastní národní korpus, včetně češtiny. Mezi české korpusy patří především skupina Českého národního korpusu, která v současnosti poskytuje osm korpusů pocházejících z psaných textů a tři korpusy přepisů mluveného textu. Jako další současné korpusy lze zmínit korpus Dialog či Pražský závislostní koprus.

Největším problémem s korpusy, který se projevuje u WSD v mnohem větší míře než u ostatních problémů, je nerovnoměrnost dat. Nikdo nám totiž nemůže zaručit, že korpus obsahuje všechny významy daného vícevýznamového slova.

V poslední době je snaha o kombinovaný přístup k řešení WSD, tedy využití hybridních systémů založených jak na korpusech, tak na bázích znalostí. Zástupce takových hybridních algoritmů je algoritmus bootstrapping, který využívá pouze malou část báze dat, kterou rozšiřuje o nalezené vzory z korpusu a iterativně si takto svou bázi dat rozšiřuje. Dále existují různé předpoklady, heuristické metody, které je možné použít při zjednoznačňování. Patří sem např. pravidlo neměnnosti významu v jedné promluvě nebo pravidlo neměnnosti významu v závislosti na slovním spojení.

## 2.4 Konference WSD

V současnosti se téměř každoročně pořádají mezinárodní konference, na kterých se předvádějí systémy pro zjednoznačňovaní slovních významů. Jednou z nich je konference Senseval (Semeval). Její poslední ročník proběhl v roce 2007 v Praze. Konference je otevřená pro všechny a v současnosti již není zaměřena pouze na WSD, ale i na další úkoly vztahující se k sémantice slov. Tato konference (resp. její část zaměřená na WSD) poskytuje srovnání pro různé WSD systémy. Poslední ročník byl

zaměřen na zjednoznačňování převážně anglického jazyka, mezi další jazyky vyhodnocované na konferenci patřila např. španělština a katalánština. Český jazyk zastoupen nebyl.

Informace o těchto konferencích lze nalézt na internetových stránkách v [10].

# Kapitola 3

# Algoritmy pro zjednoznačňování slovních významů

Pro problém WDS se nabízí řada použitelných algoritmů. Jelikož se jedná o klasickou klasifikační úlohu, kdy je potřeba zadaný obraz přiřadit do jedné z několika tříd, nabízí se možnost využití metod strojového učení. Tyto metody se principiálně dají rozdělit do dvou skupin.

Učení s učitelem, skupina, která vyžaduje trénovací data, na kterých se algoritmus nejprve naučí klasifikovat do tříd. Pro WSD jsou jednotlivé třídy chápány jako jednotlivé významy. Každá třída během trénování získá různá ohodnocení, tzv. příznakový vektor, který je následně využíván v klasifikaci. Algoritmů pro klasifikaci je mnoho, většina z nich poskytuje velmi dobré výsledky ve WDS. Jedná se např. o algoritmy podpůrných vektorů (Support vector machines), rozhodovací stromy, naivní Bayesův klasifikátor, klasifikátor AdaBoost či metoda největší entropie.

Druhá skupina, učení bez učitele, žádná trénovací data nemá. Přiřazení do patřičné třídy je závislé pouze na vstupních datech. Zde se používají algoritmy shlukování.

Stanovení příznaků pro klasifikaci je jedním z klíčových bodů všech algoritmů. Poslední poznatky ukazují, že čím více typů (skupin) příznaků, tím lepší jsou výsledky. Dodnes však stále není jasné, jak vybrat optimálně příznaky, aby reprezentovaly co nejvíce skupin a zároveň byly ještě algoritmicky zpracovatelné v adekvátní době.

V následujících částech budou popsány jednotlivé metody. Na úvod je však nutné si něco povědět o klasifikaci obecně. Informace o obecné klasifikaci byly čerpány z [4].

## 3.1 Obecný popis klasifikátorů

Uvažujme situaci, kdy na jednotlivých objektech měříme p znaků, které mohou být jak spojité, tak diskrétní. Výsledky každého měření lze zapsat pomocí p-rozměrného vektoru :

$$x = (x_1, x_2, ..., x_p) \in c = c_1 \otimes c_2 \otimes ... \otimes c_p$$

Složka $x_i$, i = 1,2,…,p, je prvkem prostoru $\chi_i \subseteq R$. Prostor $\chi$ se nazývá stavový prostor. Mějme K tříd, označme si je čísly 1, 2, …, K a zaveďme množinu tříd

$$G = \{1,2,...,K\}$$

Předpokládejme dále, že každý objekt patří právě do jedné z uvažovaných tříd, a tudíž je zcela charakterizován dvojicí

$$(k,x), k \in G, x \in c$$

V reálných situacích se často stává, že informace o třídě objektu je neznámá nebo obtížně získatelná. Cílem klasifikace je sestrojit rozhodovací (klasifikační) pravidlo, které systematickou cestou umožní předpovídat (predikovat), do které třídy klasifikovaný objekt patří. Matematicky je klasifikační pravidlo definováno jako zobrazení d ze stavového prostoru χ do množiny tříd G:

$$d : c \rightarrow G, x \ \mathbf{a} \ k$$

Zobrazení d je námi hledaný klasifikátor.

Matematický model klasifikace, který byl zaveden výše, však nepostačuje přirozenému požadavku porovnávat skutečnou třídu objektu s předpovězenou třídou. Pro libovolný objekt je možné pouze konstatovat shodnost či neshodnost třídy skutečné a předpovězené. Pro principiálně jiný pohled na klasifikaci objektů lze matematický model rozšířit na model pravděpodobnostní. Rozšíření modelu spočívá v zavedení následujících náhodných veličin a vektorů:

Zaveďme náhodnou veličinu g popisující výskyt jednotlivých tříd k ∈ G: Rozdělení L(g) veličiny g je diskrétní a je charakterizováno pravděpodobnostmi

$$p_k \equiv p_g(k) := P[g = k], k \in G$$

Od pravděpodobností $\pi_k$ požadujeme, aby splňovaly:

$$\forall k \in G(p_k > 0)$$

$$\sum_{k=1}^{K} p_k = 1$$

Na stavovém prostoru χ zaveďme p-rozměrný náhodný vektor X = ($x_1$, $x_2$,…, $x_p$). Předpokládejme, že rozdělení vektoru X je charakterizováno hustotou $p_x(x)$.

V dalším textu budeme potřebovat i sdružené rozdělení vektoru (G;X) resp. podmíněné rozdělení G při daném x a podmíněné rozdělení x při daném G. Označme si hustoty jednotlivých náhodných veličin (vektorů) (G,X ), (G|X) a (X|G) po řadě $p_G$,X (k,x), $p_{G|X}$ (k|x) a $p_{X|G}$ (x|k). Pro podmíněné hustoty dále platí, že

$$p_{G|X}(k|x) = \frac{p_{G,X}(k,x)}{p_X(x)}$$ pokud $p_X(x)$ je různé od 0, jinak 0

Po zavedení pravděpodobnostního modelu lze pomocí klasifikátoru d definovat náhodnou veličinu d(X). Pravděpodobnostní model sám od sebe nabízí výběr nejlepšího klasifikátoru: Cílem je nalézt takový klasifikátor, aby pravděpodobnost P[d(X) = G] byla co nejblíže k jedné.

Aby bylo možné co nejlépe hodnotit kvalitu klasifikátoru, zavedeme si tzv. ztrátovou funkci L(k,l):

$$L : G \times G \to \Re_0^+$$
$$(k,l) \ \mathbf{a} \ L(k,l)$$

Hodnota L(k,l) odpovídá váze ztráty, která je utrpěna, je-li prvek třídy k chybně klasifikován do třídy l. Ztrátová funkce se často zadává penalizační maticí typu K × K. Tato matice se vyznačuje nulovými prvky na diagonále, reprezentující správné zařazení. Mimodiagonální prvky jsou nezáporná čísla, jejich hodnota vyjadřuje velikost penalizace. Matice nemusí být obecně symetrická. V obecné klasifikaci se většinou volí ztrátová funkce následovně:
L(k,l) = 0 pokud k = l, jinak 1

Poté co jsme si obecně popsali klasifikátor, můžeme přistoupit ke konkrétním klasifikátorům použitých v práci.

## 3.2 Naivní Bayesův klasifikátor

Naivní Bayesův klasifikátor je jednoduchý klasifikátor založený na Bayesově teroému. Pro jeho korektnost je vyžadováno, aby příznakový vektor byl mezi sebou nezávislý. Počet příznaků, se kterými klasifikátor pracuje, není omezen, efektivně lze zpracovat klidně i tisícovky příznaků. Jeho využití je široké, mimo WSD slouží např. ke klasifikaci textů (kontrola spamu). Obrovskou výhodou tohoto klasifkátoru je fakt, že k natrénování vyžaduje relativně malé množství trénovacích dat.

Použití klasifikátoru je intuitivní. Nejprve se vzor zkouší klasifikovat do všech tříd. Vítězná třída je ta, která dosáhne nejvyšší pravděpodobnosti.

### 3.2.1 Pravděpodobnostní model

Model pro Bayesův klasifikátor je podmíněný:

$$p(C|F1, F2, \ldots, Fn)$$

Výsledná pravděpodobnost závisí tedy na třídě C a F1, F2, …, Fn jsou příznakové proměnné pro danou třídu. Aplikujeme-li na tento model Bayesův teorém, dostaneme rovnici

$$p(C|F_1, F_2, ..., F_n) = \frac{p(C) * p(F_1, F_2, ..., F_n|C)}{p(F_1, F_2, ..., F_n)}$$

Vzhledem k faktu, že nás při klasifikaci nezajímá hodnota pravděpodobnosti, ale její poměr vůči ostatním pravděpodobnostem, můžeme výpočet zjednodušit odstraněním normalizačního jmenovatele ve zlomku. Podle definice podmíněné pravděpodobnosti můžeme vzorec dále upravit na tvar:

$$p(C| F1, F2, \ldots, Fn) = p(C, F1, F2, \ldots, Fn) = p(C) * p(F1, F2, \ldots, Fn|C)$$

Použijeme-li tuto definici opakovaně, pak za předpokladu nezávislých příznaků můžeme přepsat vzorec na tvar:

$$p(C, F_1, F_2, ..., F_n) = p(C) * p(F_1|C) * p(F_2|C) * ... * p(F_n|C)$$

Výsledné rozhodnutí klasifikátoru lze zapsat jako

$$cy(f_1, f_2, ..., f_n) = \arg\max_c P(C = c) * \prod_{i=1}^{n} p(F_i = f_i | C = c)$$

Budeme-li používat tento klasifikátor ke zjednoznačňování slovních významů, použijeme patrně jako příznakový vektor pravděpodobnosti výskytů klíčových slov v kontextu odhadovaného slova za předpokladu námi určovaného významu. Všechny podmíněné pravděpodobnosti lze snadno určit z trénovacích dat. V některých případech se může stát, že vzorek textu neobsahuje některá klíčová slova. Potom je však jejich pravděpodobnost nulová, čímž i výsledná celková pravděpodobnost je nulová, což je nežádoucí. Tento problém se většinou řeší přidáním malé konstanty všem pravděpodobnostem v příznakovém vektoru. Tato operace se nazývá Laplaceovo vyhlazování.

### 3.2.2 Demonstrace použití naivního Bayesova klasifikátoru

V této kapitole si názorně ukážeme práci naivního Bayesova klasifikátoru při určování významu slov. Je zde podrobně zpracován jednoduchý příklad vycházející z testovaných dat. Trénink klasifikátoru proběhl na trénovacích datech, jak je popsáno dále.

V příkladu se snažíme určit význam slova *zájem* v tomto kontextu:

*… cenového rozpětí mezi nejlevnějšími a nejdražšími akciemi nelze vyloučit větší zájem drobných investorů právě o tyto podniky. …*

Z trénovacích dat jsem zjistil, že slovo *zájem* (nebo jeho jiný tvar) se vyskytlo celkem 125 krát. Dle významového slovníku může mít slovo zájem následující významy, na které byl klasifikátor naučen:

- *prospěch, blaho, užitek (např. hmotný zájem; osobní, obecný, veřejný zájem; zájem jednotlivce podřídit zájmu celku; hájit, poškozovat něčí zájmy; jednat ve veřejném zájmu)*
- *pozornost soustředěná na něco, někoho, účast věnovaná něčemu, někomu (např. dělat práci se zájmem, bez zájmu; sledovat představení, poslouchat přednášku s velkým zájmem; vyvolat zájem o něco, někoho; mít, ztratit zájem na něčem, někom; výstava se těší značnému zájmu veřejnosti; dostat se, vstoupit do středu, do popředí zájmu; proč má na tom, na něm takový zájem?; zájem pracujících na výsledcích podniku)*
- *náklonnost, záliba, obliba (např. projevit zájem o dívku; zájem o cestování; klukovský zájem pro dobrodružství; mít s někým společné zájmy)*

Dále, jednotlivé zjištěné četnosti a pravděpodobnosti, potřebné pro výpočet, jsou uvedeny v následující tabulce 3.1

| Slovo (třída) | počet výskytů | pravděpodobnost |
|---|---|---|
| zájem (prospěch) | 46 | 36,8% |
| zájem (pozornost) | 36 | 28,8% |
| zájem (záliba) | 43 | 34,4% |
| celkem | 125 | 100,0% |

*Tabulka 3.1: Četnost a pravděpodobnost výskytu slova zájem v natrénovaných datech*

Proměnné modelu jsou lemmata jednotlivých slov z kontextu, tedy slova *cena, rozpětí, mezi, levný, drahý, akcie, lze, vyloučit, velký, drobný, investor, tento, podnik*. Výskyty a pravděpodobnosti těchto slov v trénovacích datech byly následující (tabulka 3.2 a 3.3). Parametry, které se nevyskytly v žádné třídě, jsou ignorovány.

| slovo | třída významu | | |
|---|---|---|---|
| | zájem(prospěch) | zájem (pozornost) | zájem (záliba) |
| cena | 0 | 2 | 0 |
| mezi | 4 | 1 | 0 |
| akcie | 1 | 4 | 0 |
| lze | 1 | 1 | 1 |
| velký | 3 | 7 | 2 |
| investor | 1 | 2 | 0 |
| tento | 1 | 0 | 1 |
| podnik | 1 | 3 | 2 |

*Tabulka 3.2: Četnosti výskytů klíčových slov v trénovacích datech pro slovo zájem pro jednotlivé významy*

| slovo | třída významu | | |
|---|---|---|---|
| | **zájem(prospěch)** | **zájem (pozornost)** | **zájem (záliba)** |
| cena | 0,00% | 100,00% | 0,00% |
| mezi | 80,00% | 20,00% | 0,00% |
| akcie | 20,00% | 80,00% | 0,00% |
| lze | 33,33% | 33,33% | 33,33% |
| velký | 25,00% | 58,33% | 16,67% |
| investor | 33,33% | 66,67% | 0,00% |
| tento | 50,00% | 0,00% | 50,00% |
| podnik | 16,67% | 50,00% | 33,33% |

*Tabulka 3.3: Pravděpodobnosti výskytů klíčových slov v trénovacích datech pro slovo zájem pro jednotlivé významy*

Pohledem do tabulky zjistíme, že některé pravděpodobnosti jsou nulové. To je způsobeno faktem, že slova, která mají nulovou pravděpodobnost, se v trénovacích datech v daném významu nevyskytla. Tato skutečnost bude muset být při výpočtu ošetřena (násobení jakéhokoli čísla nulou způsobí, že výsledek bude nulový) přičtením vhodné konstanty ke všem hodnotám pravděpodobnosti, se kterými se ve výpočtu pracuje. Tato konstanta byla zvolena jako 0,1 %. Více o nastavení této konstanty bude popsáno v kapitole 4.6.

V tomto okamžiku máme všechny potřebné údaje pro klasifikaci k dispozici a můžeme zahájit klasifikaci již zmíněného textu

*… cenového rozpětí mezi nejlevnějšími a nejdražšími akciemi nelze vyloučit větší zájem drobných investorů právě o tyto podniky. …*

Základní vzorec pro klasifikátor, jak již bylo ukázáno, je následující:

$$p(C| F1, F2, \ldots, Fn) = p(C, F1, F2, \ldots, Fn) = p(C) * p(F1, F2, \ldots, Fn|C)$$

Pro náš případ bude konkrétně vypadat vzorec klasifikátoru následovně:

$$p(C|cena, mezi, akcie, lze, velký, investor, tento, podnik) = p(C) * p(cena|C) * p(mezi|C) * p(akcie|C) * p(lze|C) * p(velký|C) * p(investor|C) * p(tento|C) * p(podnik|C)$$

Je nanejvýš vhodné tento vzorec zlogaritmovat (násobení čísel menších než jedna přejde na sčítání záporných čísel) na tento tvar:

*ln(p(C|cena, mezi, akcie, lze, velký, investor, tento, podnik)) =*

*ln( p(C)) +*
*ln(p(cena|C)) +*
*ln(p(mezi|C)) +*
*ln(p(akcie|C)) +*
*ln(p(lze|C)) +*
*ln(p(velký|C)) +*
*ln(p(investor|C)) +*
*ln(p(tento|C)) +*
*ln(p(podnik|C))*

Nyní přistoupíme k samotnému výpočtu. Pro každou třídu provedeme evaluaci výše zmíněné rovnice. Výslednou třídou bude ta, jejíž pravděpodobnost bude největší.

*ln(p(zájem(prospěch)|cena, mezi, akcie, lze, velký, investor, tento, podnik)) =*

*– 0,9970*
*– 6,9078*
*– 0,2219*
*– 1,6044*
*– 1,0966*
*– 1,3823*
*– 1,0966*
*– 0,6911*
*– 1,7856*

*= – 15,7566*

*ln(p(zájem(pozornost)|cena, mezi, akcie, lze, velký, investor, tento, podnik)) =*

*– 1,2413*
*+ 0,0095*
*– 1,6044*
*– 0,2219*
*– 1,0966*
*– 0,5373*
*– 0,4039*
*– 6,9078*
*– 0,6911*

*= – 12,6948*

*ln(p(zájem(záliba)|cena, mezi, akcie, lze, velký, investor, tento, podnik)) =*

*– 0,8416*
*– 6,9078*
*– 6,9078*
*– 6,9078*
*– 1,0966*
*– 1,7856*
*– 6,9078*
*– 0,6911*
*– 1,0966*

*= – 33,1427*

Výsledná klasifikace je tedy do třídy *zájem(pozornost)*, zcela podle očekávání.

V tuto chvíli si všimněme ještě výpočtu výsledku u třídy *zájem(pozornost)*. Na druhém řádku výpočtu je zjevné použití přičtení konstanty 0,1 % vypočtené pravděpodobnosti 100 %. U pravděpodobnosti je všeobecně očekáváno, že výsledná pravděpodobnost je menší nebo rovna 100 % (a tedy logaritmus z ní je záporný nebo nulový), což v tomto případě nebylo splněno. Z hlediska našeho výpočtu však tento fakt nemá příliš velký význam a může být zanedbán. Další fakt, na který jsme v tomto případě narazili, souvisí s typem trénovacích dat a tento problém bude popsán v kapitole 4.2.

## 3.3 Klasifikátor AdaBoost

Tento klasifikátor využívá ke své klasifikaci rozhodnutí jednodušších klasifikátorů (tzv. slabých klasifikátorů). Během trénování posílí některé z nich, čímž potenciálně může dosáhnout lepších výsledků, než použití samotných slabých klasifikátorů.

### 3.3.1 Popis algoritmu

Vstupem algoritmu je trénovací množina a výstupem je klasifikátor do dvou tříd. Výsledný klasifikátor H(x) je pak lineární kombinací slabých klasifikátorů ht(x).

$$H(x) = sign\left(\sum_{t=1}^{T} a_t h_t(x)\right)$$

V každém kroku učení je do této lineární kombinace přidán jeden slabý klasifikátor z množiny klasifikátorů H. Výběr v každém kroku učení je realizován hladovým způsobem tak, aby byl minimalizován horní odhad chyby klasifikátoru.

Trénovací množina se skládá z dvojic ($x_i$, $y_i$), kde $x_i$ je klasifikovaný objekt a $y_i$ je skutečná třída, do které objekt $x_i$ patří. Označme si je množinou {−1, 1}. AdaBoost využívá při učení ovážení trénovací množiny váhami $D_t$. Ty jsou na začátku inicializovány rovnoměrně.

Vlastní algoritmus učení probíhá ve smyčce. V každém cyklu smyčky je třeba provést následující kroky:

1. nalézt nejlepší slabý klasifikátor při daném ovážení $D_t$ trénovacích dat,
2. ověřit, že chyba tohoto klasifikátoru nepřekročila 0,5,
3. spočítat koeficient slabého klasifikátoru v lineární kombinaci H(x),
4. aktualizovat váhy $D_t$.

Algoritmus učení je následující:

- *Vstup: $(x_1, y_1),\ldots, (x_m, y_m)$; $x_i$ Î X, $y_i$ Î {−1, 1}*
- *Inicializuj váhy D1(i) = 1/m*
- *For t = 1 to T:*
    1. *Find*

$$h_t = \arg\min_{h_j \in H} e_j; e_j = \sum_{i=1}^{m} D_t(i) I\left[y_i \neq h_j(x_i)\right]$$

    2. *If et ³ ½ then stop*
    3. $$a_t = \frac{1}{2}\log\left(\frac{1-e_t}{e_t}\right)$$
    4. $$D_{t+1}(i) = \frac{D_t(i)\exp(-a_t y_i h_t(x_i))}{\sum_{i=1}^{m} D_t(i)\exp(-a_t y_i h_t(x_i))}$$

Výsledný klasifikátor :

$$H(x) = sign\left(\sum_{t=1}^{T} a_t h_t(x)\right)$$

*Algoritmus 3.1: Učení klasifikátoru AdaBoost*

Vybraný slabý klasifikátor (krok 1 v cyklu) je takový, jehož vážená chyba na trénovací množině je nejmenší. Výraz I[výrok] vrací 1 pokud je výrok pravdivý a 0 pokud je nepravdivý. Podmínka v kroku 2 zajišťuje, aby byl nalezený slabý klasifikátor lepší než klasifikátor vracející náhodnou třídu. To je potřeba k zajištění konvergence algoritmu. Výběr příznaku podle minima vážené chyby společně s výpočtem koeficientu pro lineární kombinaci αt (krok 3) jsou navrženy tak, aby byl v každém kroku učení minimalizován horní odhad chyby výsledného klasifikátoru:

$$e_{tr}(H) \leq \prod_{t=1}^{T}\left(\sum_{i=1}^{m} D_t(i)\exp(-a_t y_i h_t(x_i))\right) = \frac{1}{2^T}\prod_{t=1}^{T}\sqrt{e_t(1-e_t)}$$

Aktualizace vah v kroku 4 způsobí to, že váha špatně klasifikovaných měření se zvětší a váha dobře klasifikovaných se zmenší. V následujícím kroku bude tedy hledán slabý klasifikátor, který bude muset lépe klasifikovat doposud špatně klasifikovaná měření.

Použijeme-li tento algoritmus pro řešení problému WSD, množina X trénovacích dat bude obsahovat jednotlivá slova ke zjednoznačnění, slabé klasifikátory budou představovat slova z kontextu. Klasifikace bude probíhat pro každý význam slova zvlášť a stanoví nám, zda klasifikované slovo patří do dané třídy, či ne.

### 3.3.2 Příklad určení významu

Zde si názorně ukážeme, jak lze použít navržený algoritmus AdaBoostu pro klasifikaci příkladu z kapitoly 3.2.2. Připomeňme si, že část textu použitá pro klasifikaci zněla

*… cenového rozpětí mezi nejlevnějšími a nejdražšími akciemi nelze vyloučit větší zájem drobných investorů právě o tyto podniky. …*

a cílem bylo klasifikovat slovo zájem do tříd *zájem(prospěch)*, *zájem(pozornost)* a *zájem(záliba)*. V této klasifikační metodě se zajímáme o jednotlivá slova z kontextu. Tato slova jsou pak slabými klasifikátory v algoritmu. Stanovili jsme si, že jednotlivý slabý klasifikátor bude klasifikovat do takové třídy, do které připadá nejvíce jeho výskytů. Algoritmus pro učení nám stanovuje, že každý slabý klasifikátor zařazuje buď do třídy 1 nebo -1. Prvním krokem bude sestavení tabulky slabých klasifikátorů h a jejich výsledných klasifikací h(x) v závislosti na datech x. To provedeme podle testovacích dat.

Připomeňme si tabulku pravděpodobností padnutí klasifikovaného slova *zájem* do jednotlivých tříd při výskytu klíčového slova z kap. 3.2.2.

| **slovo** | **třída významu** | | |
|---|---|---|---|
| | **zájem(prospěch)** | **zájem (pozornost)** | **zájem (záliba)** |
| cena | 0,00% | 100,00% | 0,00% |
| mezi | 80,00% | 20,00% | 0,00% |
| akcie | 20,00% | 80,00% | 0,00% |
| lze | 33,33% | 33,33% | 33,33% |
| velký | 25,00% | 58,33% | 16,67% |
| investor | 33,33% | 66,67% | 0,00% |
| tento | 50,00% | 0,00% | 50,00% |
| podnik | 16,67% | 50,00% | 33,33% |

*Tabulka 3.4: Četnosti výskytů klíčových slov v trénovacích datech pro slovo zájem pro jednotlivé významy*

Pro náš případ bude tabulka slabých klasifikátorů a jejich klasifikujících tříd vypadat následovně:

| klasifikátor | třída významu | | |
|---|---|---|---|
| | zájem(prospěch) | zájem (pozornost) | zájem (záliba) |
| cena | -1 | 1 | -1 |
| mezi | 1 | -1 | -1 |
| akcie | -1 | 1 | -1 |
| Lze | 1 | 1 | 1 |
| velký | -1 | 1 | -1 |
| investor | -1 | 1 | -1 |
| tento | 1 | -1 | 1 |
| podnik | -1 | 1 | -1 |

*Tabulka 3.5: Přiřazení jednotlivých tříd ke slabým klasifikátorům pro slovo zájem*

Pro větší názornost je uveden zápis prvního řádku tabulky i v rovnicovém zápisu.

$$h1(x1) = cena(zájem(prospěch)) = -1$$

$$h1(x2) = cena(zájem(pozornost)) = 1$$

$$h1(x3) = cena(zájem(záliba)) = -1$$

Zbývá nám určit váhy $\alpha_i$ jednotlivých slabých klasifikátorů. Tyto váhy se iterativně vypočítají z chyb slabých klasifikátorů pomocí výše uvedeného algoritmu. Chyba je stanovena jako počet chybně určených klasifikací za použití pouze tohoto jednoho slabého klasifikátoru.

Z natrénovaných dat jsme odhadli následující chyby pro použité slabé klasifikátory.

| klasifikátor | chybná klasifikace | | |
|---|---|---|---|
| | zájem(prospěch) | zájem (pozornost) | zájem (záliba) |
| Cena | 0,00% | 0,00% | 0,00% |
| mezi | 20,00% | 20,00% | 0,00% |
| akcie | 20,00% | 20,00% | 0,00% |
| Lze | 33,33% | 33,33% | 33,33% |
| velký | 25,00% | 41,67% | 16,67% |
| investor | 33,33% | 33,33% | 0,00% |
| tento | 50,00% | 0,00% | 50,00% |
| podnik | 16,67% | 50,00% | 33,33% |

*Tabulka 3.6: Pravděpodobnost chybné klasifikace slabých klasifikátorů pro slovo zájem a jednotlivé třídy*

Za použití algoritmu pro výpočet vah jsme získali následující váhy (tabulka 3.7)

| slovo | váha |
|---|---|
| cena | 1,50 |
| mezi | 0,48 |
| akcie | 0,49 |
| lze | 0,15 |
| velký | 0,26 |
| investor | 0,49 |
| tento | 0,50 |
| podnik | 0,25 |

*Tabulka 3.7: Váhy slabých klasifikátorů pro jednotlivé slabé klasifikátory*

Nyní můžeme přistoupit k vlastní klasifikaci.

Obecně takto natrénovaný klasifikátor rozhoduje následovně:

$$H(C) = \alpha_{cena} * h_{cena}(C) + \alpha_{mezi} * h_{mezi}(C) + \alpha_{akcie} * h_{akcie}(C) + \alpha_{lze} * h_{lze}(C) + \alpha_{velký} * h_{velký}(C) + \alpha_{investor} * h_{investor}(C) + \alpha_{tento} * h_{tento}(C) + \alpha_{podnik} * h_{podnik}(C)$$

Pro námi klasifikované slovo je to tedy

$$H(prospěch) = \alpha_{cena} * h_{cena}(prospěch) + \alpha_{mezi} * h_{mezi}(prospěch) + \alpha_{akcie} * h_{akcie}(prospěch) + \alpha_{lze} * h_{lze}(prospěch) + \alpha_{velký} * h_{velký}(prospěch) + \alpha_{investor} * h_{investor}(prospěch) + \alpha_{tento} * h_{tento}(prospěch) + \alpha_{podnik} * h_{podnik}(prospěch) = -1,5 + 0,48 - 0,49 + 0,15 - 0,26 - 0,49 + 0,50 - 0,25 = -1,86$$

$H(pozornost) =$

$\alpha_{cena} * h_{cena}(pozornost) +$
$\alpha_{mezi} * h_{mezi}(pozornost) +$
$\alpha_{akcie} * h_{akcie}(pozornost) +$
$\alpha_{lze} * h_{lze}(pozornost) +$
$\alpha_{velký} * h_{velký}(pozornost) +$
$\alpha_{investor} * h_{investor}(pozornost) +$
$\alpha_{tento} * h_{tento}(pozornost) +$
$\alpha_{podnik} * h_{podnik}(pozornost) =$
$1,5 – 0,48 + 0,49 + 0,15 + 0,26 + 0,49 – 0,50 + 0,25 =$
$2,16$

$H(záliba) =$

$\alpha_{cena} * h_{cena}(záliba) +$
$\alpha_{mezi} * h_{mezi}(záliba) +$
$\alpha_{akcie} * h_{akcie}(záliba) +$
$\alpha_{lze} * h_{lze}(záliba) +$
$\alpha_{velký} * h_{velký}(záliba) +$
$\alpha_{investor} * h_{investor}(záliba) +$
$\alpha_{tento} * h_{tento}(záliba) +$
$\alpha_{podnik} * h_{podnik}(záliba) =$
$– 1,5 – 0,48 – 0,49 + 0,15 – 0,26 – 0,49 + 0,50 – 0,25 =$
$– 2,86$

Klasifikace tedy dopadla dle očekávání, slovo zájem bylo zařazeno do třídy zájem(pozornost), ostatní třídy byly zamítnuty.

### 3.3.3 Odlišnosti implementovaného algoritmu

Implementovaný algoritmus AdaBoostu, jak je popsán v kapitole 3.3.1, předpokládá, že hodnota všech slabých klasifikátorů je v okamžiku klasifikace vyčíslitelná. Tento předpoklad je, zvláště pokud bereme jako slabé klasifikátory všechna slova z kontextu, v některých případech nesplnitelný. Jedná se zejména o případy, kdy trénovací data obsahují pouze několik příkladů pro klasifikaci do jedné tříd a dojde k tomu, že klíčové slovo klasifikuje ve všech případech pouze do jedné třídy. Z hlediska algoritmu dojde k tomu, že takový slabý klasifikátor rázem dostane velmi silné ohodnocení a potlačí ostatní slabší klasifikátory. Pokud poté dojde k tomu, že je stejný klasifikátor použit na jiný kontext, může dojít k tomu, že mnoho takto silně ohodnocených slabých klasifikátorů není možné vyčíslit, protože se v rozpoznávaném kontextu nenacházejí (Jestliže by byl např. použit ke klasifikaci stejně natrénovaný klasifikátor, jak v příkladu z kapitoly 3.3.2, ovšem v kontextu by se nevyskytlo slovo cena, výsledek klasifikátoru by nebyl už tak markantní a pokud by takových případů nastalo více, bylo by možné,

že by se výsledek klasifikace výrazně změnil). Řešením tohoto problému bylo zvolení dynamického kontextu (viz kap. 4.3).

Další možností, která byla vyzkoušena, byla změna kriteria pro volbu slabého klasifikátoru, který má být aktuálně použit. Úprava spočívala v prvotním seřazení klasifikátorů podle úspěšnosti. Výpočet vah poté probíhal v pořadí od nejúspěšnějšího slabého klasifikátoru směrem k nejméně úspěšnému. Výpočet ustal v okamžiku, kdy chyba klasifikátoru byla větší než 0,5 a jako výsledné slabé klasifikátory byly brány pouze ty, které měly chybu menší. Tento způsob se však příliš neosvědčil, protože vylučoval velké množství klasifikátorů, které ačkoli nebyly pro jednotlivé rozhodování příliš přesné, dokázaly výsledek korigovat.

## 3.4 Klasifikace pomocí maximální entropie

Dalším možným způsobem, jak určit význam slova, je použití metody maximální entropie. Princip maximální entropie je ve zpracování přirozeného jazyka obecně široce použitelný. Aplikace tohoto principu je použitelná např. pro strojový překlad nebo pro značkování textu.

### 3.4.1 Princip maximální entropie

Princip maximální entropie tvrdí, že nevíme-li o daném statistickém rozložení všechny informace, ale jen některé, jako je např. střední hodnota, střední kvadratická odchylka, poměr pravděpodobností padnutí některých stavů, potom nejpravděpodobnější tvar rozdělení je takový, který splňuje požadavky, které o rozdělení máme, a má nejvyšší možnou entropii. Tato entropie se spočítá podle vzorce

$$S = -\sum_{i=1}^{N} \left( p_i \log p_i \right)$$

Kde N je celkový počet stavů a $p_i$ je pravděpodobnost padnutí daného stavu. Míra entropie S je taktéž interpretovatelná jako množství informace.

Motivace k užití principu maximální entropie je taktéž vlastnost, že efektivně modeluje normální rozložení pravděpodobnosti.

V metodě maximální entropie se používají, stejně jako ve většině klasifikačních metod s učitelem, trénovací data. Tato data se použijí jako omezení rozložení, které musí být splněno při odhadu. Pravděpodobnost $p_i$ získáme jako funkci pravděpodobnosti padnutí daného slova do určité třídy významu: $p_i(d,i)$, kterou jsme získali na testovacích datech. Symbol d vyjadřuje slovo z kontextu a i význam klasifikovaného slova.

Vlastní klasifikace probíhá následovně: Pro všechny třídy se z trénovacích dat vypočte pravděpodobnost $p_i(d,i)$.

$$p_i(d, j) = p(d|j) \text{ pokud j = i, jinak } 1 - p(d|j)$$

Následně se vypočte entropie pro všechny možné případy podle výše uvedeného vzorce. Jako výsledná třída se poté vybere ta, která má maximální entropii.

### 3.4.2 Příklad klasifikace pomocí maximální entropie

Vezměme si opět část textu

*… cenového rozpětí mezi nejlevnějšími a nejdražšími akciemi nelze vyloučit větší zájem drobných investorů právě o tyto podniky. …*

a opět se pokusíme určit význam slova *zájem*.

Prvním krokem bude zjištění pravděpodobností $p_i(d,j)$. Tato pravděpodobnost je odhadnutelná z trénovacích dat a je zobrazena v tabulkách pro každou třídu. Stanovme si, že pravděpodobnost pro každou tabulku třídy i stanovíme jako

$$p_i(d, j) = p(d|j) \text{ pokud j = i, jinak } 1 - p(d|j)$$

kde proměnná j se bere přes všechny třídy. Takto vytvořené tabulky určují pravděpodobnost, že klasifikátor zařadí podle daného kontextu slovo do třídy i a nezařadí do žádné jiné.

Znovu si připomeňme tabulku pravděpodobností padnutí klasifikovaného slova *zájem* do jednotlivých tříd při výskytu klíčového slova z kap. 3.2.2

| **slovo** | **třída významu** | | |
|---|---|---|---|
| | **zájem(prospěch)** | **zájem (pozornost)** | **zájem (záliba)** |
| cena | 0,00% | 100,00% | 0,00% |
| mezi | 80,00% | 20,00% | 0,00% |
| akcie | 20,00% | 80,00% | 0,00% |
| lze | 33,33% | 33,33% | 33,33% |
| velký | 25,00% | 58,33% | 16,67% |
| investor | 33,33% | 66,67% | 0,00% |
| tento | 50,00% | 0,00% | 50,00% |
| podnik | 16,67% | 50,00% | 33,33% |

*Tabulka 3.8: Pravděpodobnost výskytů klíčových slov v trénovacích datech pro slovo zájem pro jednotlivé významy*

Tabulky pro jednotlivé třídy jsou podle vzorce následující (tabulky 3.9, 3.10 a 3.11):

| slovo | třída významu *zájem(prospěch)* | | |
|---|---|---|---|
| | **zájem(prospěch)** | **zájem (pozornost)** | **zájem (záliba)** |
| cena | 0,00% | 0,00% | 100,00% |
| mezi | 80,00% | 80,00% | 100,00% |
| akcie | 20,00% | 20,00% | 100,00% |
| lze | 33,33% | 67,67% | 66,67% |
| velký | 25,00% | 41,67% | 83,33% |
| investor | 33,33% | 33,33% | 100,00% |
| tento | 50,00% | 100,00% | 50,00% |
| podnik | 16,67% | 50,00% | 67,66% |

*Tabulka 3.9: Pravděpodobnosti přiřazení slova zájem do třídy prospěch a nepřiřazení do ostatních tříd*

| slovo | třída významu *zájem(pozornost)* | | |
|---|---|---|---|
| | **zájem(prospěch)** | **zájem (pozornost)** | **zájem (záliba)** |
| cena | 100,00% | 100,00% | 100,00% |
| mezi | 20,00% | 20,00% | 100,00% |
| akcie | 80,00% | 80,00% | 100,00% |
| lze | 66,67% | 33,33% | 66,67% |
| velký | 75,00% | 58,33% | 83,33% |
| investor | 66,67% | 66,67% | 100,00% |
| tento | 50,00% | 0,00% | 50,00% |
| podnik | 83,33% | 50,00% | 66,67% |

*Tabulka 3.10: Pravděpodobnosti přiřazení slova zájem do třídy pozornost a nepřiřazení do ostatních tříd*

| slovo | třída významu *zájem(záliba)* | | |
|---|---|---|---|
| | **zájem(prospěch)** | **zájem (pozornost)** | **zájem (záliba)** |
| cena | 100,00% | 0,00% | 0,00% |
| mezi | 20,00% | 80,00% | 0,00% |
| akcie | 80,00% | 20,00% | 0,00% |
| lze | 66,67% | 66,67% | 33,33% |
| velký | 75,00% | 41,67% | 16,67% |
| investor | 66,67% | 33,33% | 0,00% |
| tento | 50,00% | 100,00% | 50,00% |
| podnik | 83,33% | 50,00% | 33,33% |

*Tabulka 3.11: Pravděpodobnosti přiřazení slova zájem do třídy záliba a nepřiřazení do ostatních tříd*

Tyto tabulky jsou jediný vstup algoritmu. Můžeme tedy přejít ke klasifikaci. Vyjádříme si entropii každé takto vytvořené tabulky a podle toho rozhodneme, do jaké třídy klasifikované slovo patří.

Obecný tvar vzorce pro výpočet entropie na takto natrénovaných datech bude

*S(C) =*

- *Σ(p(cena|C) * log(p(cena|C)))*
- *Σ (p(mezi|C) * log(p(mezi|C)))*
- *Σ (p(akcie|C) * log(p(akcie|C)))*
- *Σ (p(lze|C) * log(p(lze|C)))*
- *Σ (p(velký|C) * log(p(velký|C)))*
- *Σ (p(investor|C) * log(p(investor|C)))*
- *Σ (p(tento|C) * log(p(tento|C)))*
- *Σ (p(podnik|C) * log(p(podnik|C)))*

Přistupme nyní k výpočtu pro všechny třídy C. Při dosazování hodnot je opět nutné přičíst malou konstantu, která zajistí, aby se logaritmus nepočítal z nulové hodnoty

*S(zájem(prospěch)) =*

*- Σ (p(cena|zájem(prospěch)) * log(p(cena|zájem(prospěch))))*
*- Σ (p(mezi|zájem(prospěch)) * log(p(mezi|zájem(prospěch))))*
*- Σ (p(akcie|zájem(prospěch)) * log(p(akcie|zájem(prospěch))))*
*- Σ (p(lze|zájem(prospěch)) * log(p(lze|zájem(prospěch))))*
*- Σ (p(velký|zájem(prospěch)) * log(p(velký|zájem(prospěch))))*
*- Σ (p(investor|zájem(prospěch)) * log(p(investor|zájem(prospěch))))*
*- Σ (p(tento|zájem(prospěch)) * log(p(tento|zájem(prospěch))))*
*- Σ (p(podnik|zájem(prospěch)) * log(p(podnik|zájem(prospěch))))*

*=*

*– 0,001 * 3,000 – 0,001 * 3,000 + 1,001 * 0,000*
*– 0,801 * 0,096 – 0,801 * 0,096 + 1,001 * 0,000*
*– 0,201 * 0,697 – 0,201 * 0,697 + 1,001 * 0,000*
*– 0,334 * 0,476 – 0,667 * 0,175 – 0,667 * 0,175*
*– 0,251 * 0,600 – 0,417 * 0,379 – 0,834 * 0,078*
*– 0,334 * 0,476 – 0,334 * 0,476 + 1,001 * 0,000*
*– 0,501 * 0,300 + 1,001 * 0,000 – 0,501 * 0,300*
*– 0,167 * 0,777 – 0,501 * 0,300 – 0,667 * 0,169*

*= – 2,220*

*S(zájem(pozornost)) =*

*- Σ (p(cena|zájem(pozornost)) * log(p(cena|zájem(pozornost))))*
*- Σ (p(mezi|zájem(pozornost)) * log(p(mezi|zájem(pozornost))))*
*- Σ (p(akcie|zájem(pozornost)) * log(p(akcie|zájem(pozornost))))*
*- Σ (p(lze|zájem(pozornost)) * log(p(lze|zájem(pozornost))))*
*- Σ (p(velký|zájem(pozornost)) * log(p(velký|zájem(pozornost))))*
*- Σ (p(investor|zájem(pozornost)) * log(p(investor|zájem(pozornost))))*
*- Σ (p(tento|zájem(pozornost)) * log(p(tento|zájem(pozornost))))*
*- Σ (p(podnik|zájem(pozornost)) * log(p(podnik|zájem(pozornost))))*

*=*

*+ 1,001 * 0,000 + 1,001 * 0,000 + 1,001 * 0,000*
*– 0,201 * 0,697 – 0,201 * 0,697 + 1,001 * 0,000*
*– 0,801 * 0,096 – 0,801 * 0,096 + 1,001 * 0,000*
*– 0,667 * 0,175 – 0,334 * 0,476 – 0,667 * 0,175*
*– 0,751 * 0,124 – 0,584 * 0,233 – 0,834 * 0,078*
*– 0,667 * 0,175 – 0,667 * 0,175 + 1,001 * 0,000*
*– 0,501 * 0,300 + 0,001 * 3,000 – 0,501 * 0,300*
*– 0,834 * 0,078 – 0,501 * 0,300 – 0,667 * 0,169*

*= – 1,989*

*S(zájem(záliba)) =*

*- Σ (p(cena|zájem(záliba)) * log(p(cena|zájem(záliba))))*
*- Σ (p(mezi|zájem(záliba)) * log(p(mezi|zájem(záliba))))*
*- Σ (p(akcie|zájem(záliba)) * log(p(akcie|zájem(záliba))))*
*- Σ (p(lze|zájem(záliba)) * log(p(lze|zájem(záliba))))*
*- Σ (p(velký|zájem(záliba)) * log(p(velký|zájem(záliba))))*
*- Σ (p(investor|zájem(záliba)) * log(p(investor|zájem(záliba))))*
*- Σ (p(tento|zájem(záliba)) * log(p(tento|zájem(záliba))))*
*- Σ (p(podnik|zájem(záliba)) * log(p(podnik|zájem(záliba))))*

=

*+ 1,001 * 0,000 – 0,001 * 3,000 – 0,001 * 3,000*
*– 0,201 * 0,697 – 0,801 * 0,096 – 0,001 * 3,000*
*– 0,801 * 0,096 – 0,201 * 0,697 – 0,001 * 3,000*
*– 0,667 * 0,175 – 0,667 * 0,175 – 0,334 * 0,476*
*– 0,751 * 0,124 – 0,417 * 0,379 – 0,167 * 0,775*
*– 0,667 * 0,175 – 0,334 * 0,476 – 0,001 * 3,000*
*– 0,501 * 0,300 + 1,001 * 0,000 – 0,501 * 0,300*
*– 0,834 * 0,078 – 0,501 * 0,300 – 0,334 * 0,476*

*= – 2,175*

I v tomto konkrétním případě metoda maximální entropie byla úspěšná, také označila jako nejpravděpodobnější výsledek třídu *zájem(pozornost)*. Obecně je však tato metoda nejslabší z implementovaných, podrobné výsledky jsou zmíněny v kap. 5.3.

## 3.5 Rozhodovací stromy

Rozhodovací stromy jsou jednou z nejčastěji používaných technik ke klasifikaci a to z několika důvodů: Jsou přehledné a snadno interpretovatelné. Jejich cílem je, stejně jako u všech klasifikačních algoritmů, identifikovat objekty podle atributů a zařadit je do tříd. Rozhodovací strom může být reprezentován tabulkou, kde jednotlivé sloupce jsou atributy, jako např. sousední slovo, velikost písmen, … Algoritmus klasifikace je velmi rychlý, protože se jedná o procházení stromu. Samotný klasifikační strom se musí nejprve natrénovat na datech. Každý uzel takového stromu reprezentuje jednu vlastnost klasifikovaného objektu. Z takového uzlu pak vede konečný počet hran, které představují rozhodovací možnosti. Tyto vlastnosti je v některých případech nutné nejprve nějak rozdělit do konečného počtu intervalů, a to tehdy, když je vlastnost spojitá. Nejtěžší úkol je vytvoření takového stromu. Musí dojít k vybrání takových vlastností, které objekty rozlišují nejvyšší možnou měrou. Jako kořenový uzel se pak vybere taková vlastnost, která dané třídy rozlišuje nejvíce. S výhodou se tady používá entropie klasifikované vlastnosti (viz kap. 3.4). Pro tvorbu rozhodovacího stromu existuje mnoho algoritmů, mezi něž patří např. CART, C4.5 nebo ID3, který byl použit v této práci.

### 3.5.1 Algoritmus ID3

Algoritmus ID3 lze popsat následovně:

1. Strom začíná jako samostatný uzel reprezentující tréninkové záznamy.
2. Jestliže jsou všechny záznamy stejné třídy, uzel se stane listem a je označen výslednou třídou.
3. Pokud nepatří všechny záznamy do dané třídy, provede se výpočet entropie, jež vybere nejlepší atribut, který rozdělí vzorky do tříd. Tento atribut se stane testovacím pro daný uzel.
4. Pro každou možnou hodnotu atributu je vytvořena vlastní větev a vzorky jsou přiřazeny jednotlivým větvím.

Tímto způsobem se rekurzivně vytvoří celý strom. Jakmile se použije nějaký atribut jako testovací, nesmí se již dále použít v dané větvi stromu.

Je evidentní, že tvorba stromu končí ve chvíli, kdy všechny vzorky patří buď do stejné třídy, nebo není možné nalézt žádný atribut, který provede dělení. V takovém případě proběhne tzv. „většinová volba“ – výsledný uzel se převede na list a označí se třídou převažujících vzorků.

V této práci je rozhodovací strom brán pouze jako podpůrný nástroj. V přirozeném jazyce existuje mnoho výjimek, či právě naopak ustálených vazeb, které mají svůj stálý význam. Právě díky těmto často přeneseným významům klasické klasifikátory selhávají. Cílem tvorby stromu bylo zaměřit se na tyto případy a vytvořit tak nástroj pro klasifikaci těchto výjimek. Jednotlivé atributy byly při tvorbě stromu vybírány s ohledem na tento fakt, a proto výsledný strom není klasifikátorem v pravém slova smyslu.

# Kapitola 4

# Implementace systému pro zjednoznačňování slovních významů

V předchozí kapitole jsme si popsali metody, které byly použity ve výsledné implementaci. V této kapitole se zaměříme na vlastní implementaci systému. Popíšeme si použitá data, vlastní návrh, realizaci implementace a výstup programu. Nezbytnou součástí bude popis struktury a typu dat, které byly použity při učení klasifikátorů. V kapitole je také ukázán způsob, jak je pomocí jednotlivých metod ovlivněn celkový výsledek klasifikace.

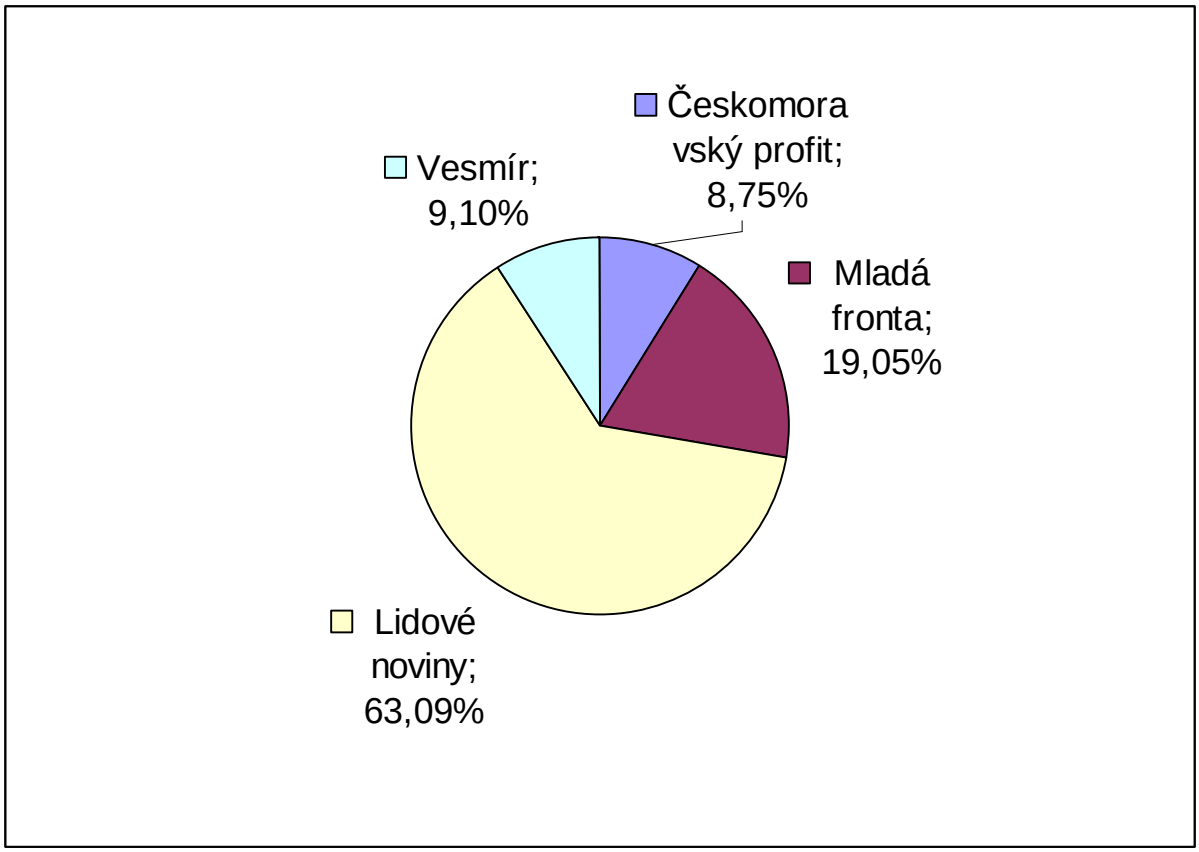
Nalézt vhodná trénovací data pro český jazyk nebylo snadné. Mnoho dat existuje pro anglický jazyk. Mým cílem však byl klasifikátor pro češtinu. Zde jsem měl v podstatě na výběr mezi dvěma korpusy: Pražským závislostním korpusem (Prague dependency treebank, PDT) a Českým národním korpusem (ČNK). Ačkoli Český národní korpus obsahuje více dat (cca 500 milionů slovních jednotek), nakonec jsem se spokojil s Pražským závislostním korpusem, který byl dostupnější. Z této volby však posléze vyplynuly některé problémy, týkající se především dostatečného množství dat pro trénování. Problémy a nastínění řešení budou popsány v dalších částech práce, konkrétně v páté kapitole, zabývající se výsledky.

## 4.1 Pražský závislostní korpus

Vlastní data pro trénování a následné testování algoritmů pro zjednoznačňování slovních významů, které byly implementovány, jsem čerpal, jak již bylo řečeno, z Pražského závislostního korpusu. Tento korpus obsahuje anotované nezkrácené články z těchto novin a časopisů:

- Lidové noviny (deník), ISSN 1213-1385, 1991, 1994, 1995
- Mladá fronta Dnes (deník), 1992
- Českomoravský Profit (ekonomický týdeník), 1994
- Vesmír (populárně vědecký měsíčník), ISSN 1214-4029, Vesmír, s.r.o., 1992, 1993

Rozdělení dat podle zdrojů je v následujícím grafu




*Graf 4. 1: Rozdělení zdrojů v použitém Pražském závislostním korpusu*

Korpus obsahuje celkem téměř 2 000 000 slovních jednotek (jednotlivých slov, čísel, interpunkčních znamének) a je anotován ve třech vrstvách, které se liší množstvím informací. Pro mou práci byla použita pouze vrstva nejnižší, která obsahuje pouze rozdělení textu na jednotlivé slovní jednotky.

Data v korpusu jsou rozdělena do tří skupin: Trénovací data (train), vývojová testovací data (dtest) a evaluační testovací data (etest). Trénovací data tvoří přibližně 80% celkového množství dat, vývojová 10% a evaluační rovněž 10%. Podrobné informace o rozdělení jednotlivých použitých dat je v následující tabulce 4.1:

| | train | dtest | etest | celkem |
|---|---|---|---|---|
| počet dokumentů | 1 422<br>79,9% | 179<br>10,1% | 179<br>10,1% | 1 780<br>100,0% |
| počet vět | 22 333<br>80,0% | 2 610<br>9,3% | 2 988<br>10,7% | 27 931<br>100,0% |
| počet slovních jednotek | 364 640<br>80,4% | 42 689<br>9,4% | 46 179<br>10,2% | 453 508<br>100,0% |

*Tabulka 4.1: Rozdělení dat v Pražském závislostním korpusu podle účelu*

### 4.1.1 Použití korpusu

Pražský závislostní korpus je korpus obecný, nemá nějak explicitně označeny vícevýznamová slova, která stála v popředí mého zájmu. Data v něm jsou uložena ve formátu, který vychází z XML, čímž jsou snadno zpracovatelná.

Datová struktura na vrstvě „w“, která byla použita, obsahuje pouze vlastní slovní jednotky. Pokud bychom chtěli použít další vrstvy pro rozpoznávání, nabízí se ručně anotovaná vrstva „a“, která dále obsahuje lemmata daných slov. Tato vrstva však nebyla v mé práci použita, a to z důvodu obecnosti. Více o tomto důvodu je napsáno v kapitole 4.2.

Dalším zdrojem pro stanovení hledaných slov a jejich významů byl významový slovník. Ze slovníku jsem čerpal slova, která jsem označil za víceznačná.

## 4.2 Vytvořené datové sady

Ke správné funkci klasifikačních algoritmů bylo nutné uchovat již naučené znalosti. Proto jsem se rozhodl vytvořit si vlastní strukturu (bázi) znalostí. Základní znalostní bází je soubor s vícevýznamovými slovy a jejich významy. Celkem je umožněno tímto souborem rozpoznat mým systémem význam 248 různých slov ve všech jejich tvarech. Dále je pro každé vícevýznamové slovo vytvořena další struktura obsahující údaje jako celkový počet výskytů, počet výskytů v jednotlivých významech, možné tvary slova a několik dalších údajů, které jsou popsány v příloze 1.

V pokročilejších fázích testování vznikla dále potřeba určovat pro všechna slova jejich lemma. Ačkoliv by bylo možné použít již výše zmíněné údaje z vrstvy „a“ PDT, rozhodl jsem se implementovat vlastní metodu, která se bude pokoušet lemmatizovat slova. Metoda je založena na koncovkách českých slov. Spočívá v detekování koncovky slova a následném odstranění. Detekce probíhá „hltavým“ způsobem, kdy se snaží nalézt co nejdelší možnou koncovku. Takto vytvořené „pseudo-lemma“ je pak použito k vytvoření údajů o kontextu rozpoznávaného slova. Je evidentní, že takto provedená lemmatizace nebude stoprocentně úspěšná, nicméně povede na stav, kdy každému různému slovu přiřadí jiné lemma a každému jinému tvaru stejného slova

přiřadí buď stejné, nebo podobné lemma, což je lepší, než stav bez lemmatizace a pro detekci klíčových slov postačující.

Navíc k úplné úspěšné lemmatizaci by bylo nutné znát druh slova, jelikož pro různý slovní druh může stejné slovo mít jiné lemma. V českém jazyce například slovo psát může mít lemma psaní (např. ve větě: „Učí se psát.“) nebo psát (např. věta „Chtěl psát dopis.“), podle toho, zda se jedná o podstatné jméno nebo sloveso. Implementovaný algoritmus je však schopen pracovat i bez toho, jelikož slovní druh zpravidla neovlivňuje význam slova.

Algoritmus lemmatizace je popsán v příloze 2, kde jsou uvedeny i jednotlivé koncovky.

Dále se během trénování ukázalo, že obecný korpus není ve všech případech schopen pokrýt veškeré významové případy jednotlivých slov. Aby byly algoritmy schopny rozpoznat alespoň některé další případy, bylo nutné je naučit i z jiných textů. Pro toto učení byla použita obyčejná data ve formátu TXT. Takto hrubá data neobsahovala implicitně žádné informace a proto by byly trénovací algoritmy, které by vyžadovaly i další informace (lemma, slovní druh) nepoužitelné. V zájmu této zpětné kompatibility byla použita z korpusu pouze vrstva „w“, obsahující hrubá data.

Pro srovnání algoritmů byl následně vytvořen jednoduchý algoritmus, který určoval poměrně naivním způsobem význam slova. Algoritmus vybere z kontextu rozpoznávaného slova ta slova, která se vyskytují v již naučeném kontextu a srovná jejich počet vynásobený pravděpodobností padnutí do dané třídy v závislosti na slovu kontextu. Ta třída, která má největší skóre, se označí jako vítězná. Pseudoalgoritmus takového naivního klasifikátoru je následující:

- *FOR <všechny významy rozpoznávaného slova> v*
  1. $H_v = 0$
  2. *FOR <všechny slova z kontextu rozpoznávaného slova> k*
     - *IF <k> IN <naučený kontext rozpoznávaného významu> THEN* $H_v = H_v$ *+ <počet výskytů <k> v naučeném kontextu ve významu <v>> / <počet výskytů <k> v naučeném kontextu>*
     - *IF <k> IN <naučený kontext jiného než rozpoznávaného významu> THEN* $H_v = H_v$ *– <počet výskytů <k> v naučeném kontextu ve významu <v>> / <počet výskytů <k> v naučeném kontextu>*
- *Result = <v> WHERE* $H_v$ *IS max*

*Algoritmus 4.1: Naivní klasifikátor*

## 4.3 Výběr příznaků

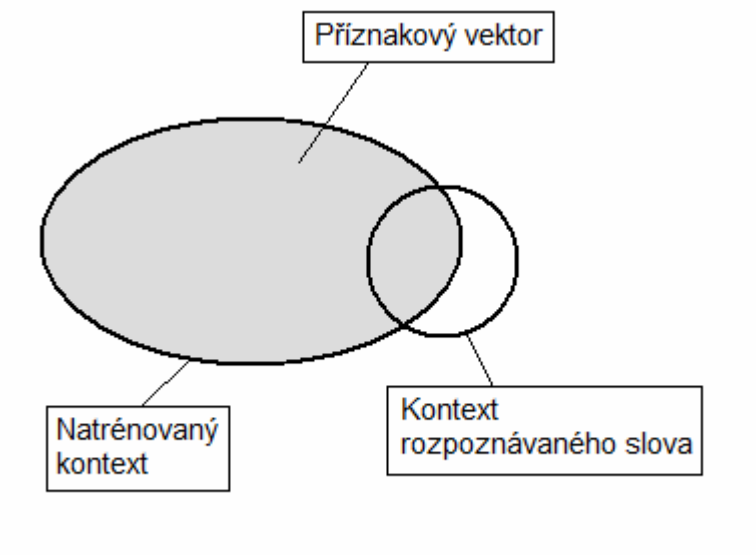
Výběr příznaků je jeden z nejdůležitějších kroků klasifikačních metod. Sebelepší algoritmus, pokud mu budou dodány nekvalitní zdroje, nebude produkovat dobré výsledky. S ohledem na tento fakt musí být výběr příznaků proveden co nejkvalitněji.

Způsobů, jak vybrat příznaky pro klasifikátory, je poměrně velké množství. Pro WSD se nabízí vytvoření příznakového vektoru z celého kontextu daného významu. Každé slovo v natrénovaném kontextu poté znamená jednu položku příznakového vektoru a jeho hodnota je nastavena buď na 1, pokud se slovo v kontextu rozpoznávaného slova vyskytuje, nebo na 0, pokud ne. Takový příznakový vektor je univerzální (globální), může být použit, kdykoli se vyskytne v rozpoznávaném textu stejné slovo, na které je příznakový vektor připraven. Takto připravený vektor příznaků zpravidla obsahuje mnoho položek a vyplnění hodnotami je závislé pouze na velikosti kontextu. Výhodou tohoto přístupu je stále stejná délka vektoru pro stejné slovo. Významnou nevýhodou však je fakt, že vektor zpravidla obsahuje velký počet nulových hodnot, což z informačního hlediska není příliš přínosné.

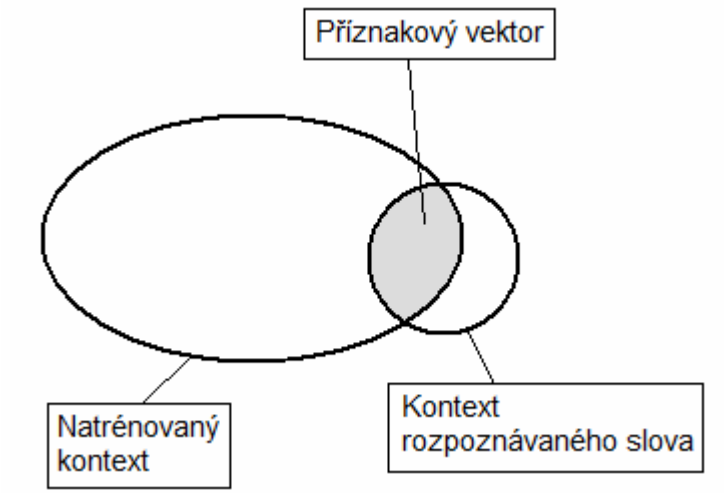
V mnou implementovaných algoritmech je tvorba příznakového vektoru dynamická pro každý jednotlivý výskyt rozpoznávaného slova. Takto dynamicky vytvořeným příznakovým vektorem dosáhneme mnohem nižšího počtu příznaků a zároveň vyšší zaplnění příznakového vektoru relevantními údaji, než je tomu v předchozím případě a tím i snížení výpočetní náročnosti. Slova -příznaky- jsou vždy vybírána tak, aby se vyskytovala jak v kontextu rozpoznávaného slova, tak i v kontextu natrénovaném pro dané slovo. Touto operací se příznakový vektor smrskne na cca 10 položek. Navíc je zaručeno, že příznakový vektor bude obsahovat vždy alespoň nějaké položky, pokud to bude možné. V případě, že bude příznakový vektor prázdný, z většiny implementovaných klasifikačních metod se stanou pouze slepé klasifikátory pracující na základě statistických dat ze zpracovaných trénovacích souborů (zpravidla dojde k označení nejčastěji se vyskytovaného významu). Výhodou tohoto přístupu je již zmíněný malý počet položek vektoru. Drobnou nevýhodou je dopředu neznámý a proměnný počet položek pro každý výskyt stejného slova. Tato nevýhoda však není příliš markantní, aby převážila výhody rychlejšího zpracování.

Pro srovnání rozpoznávání založené na dynamickém a globálním příznakovém vektoru byla v rané fázi implementace systému vyzkoušena práce metody AdaBoost s globálním příznakovým vektorem.

Srovnání těchto dvou přístupů tvorby příznakových vektorů je na obrázcích 4.1 a 4.2.




*Obrázek 4.1: Ilustrace obsahu příznakového vektoru při globálním použití*




*Obrázek 4.2: Ilustrace obsahu příznakového vektoru při dynamickém použití*

## 4.4 Práce programu

Výsledný program byl implementován v jazyce Object Pascal v prostředí Delphi5. Běh programu řídí hlavní formulář, který poskytuje přístup k několika možným funkcí programu. Jedná se zejména o trénování algoritmů, testování algoritmů a úpravu dat. Při spuštění programu dojde k naplnění struktury *homonyma*, která obsahuje veškeré doposud naučené informace. Tato operace je poněkud časově náročnější, neboť struktura je nosnou částí programu a obsahuje veškerá data o všech víceznačných

slovech. Informace čerpá ze souborů `homonyma.txt`, který obsahuje seznam všech rozpoznávaných víceznačných slov a jejich možné významy. Tento soubor má pevnou strukturu, kdy na každém řádku je jedno vícevýznamové slovo a tabelátorem (znak 9) jsou odděleny všechny možné významy. V případě potřeby je možné tento soubor editovat a doplňovat tak další vícevýznamová slova, případně jejich významy. Na ruční editaci je kladeno jediné omezení, a to aby bylo zachováno pořadí významů jednotlivých slov v rámci řádku (V programu je dále umožněna editace pomocí jedné z metod vlastní aplikace). Po načtení tohoto souboru dojde k načtení doplňujících informací o všech vícevýznamových slovech z dalších souborů. Tyto soubory jsou pojaty jako inicializační soubory s patřičnou strukturou a jsou pojmenovány `<vícevýznamové_slovo>.ini`, kde `<vícevýznamové_slovo>` je interní název rozpoznávaného slova (až na drobnou odlišnost reprezentující skutečné slovo), které je převzato jako první text z každého řádku souboru `homonyma.txt`. Při načítání těchto *.ini souborů je dále kontrolováno, jestli jsou u všech slov, která byla označena jako skloňovatelná (viz. příloha 1) dostupné alespoň nějaké tvary. Pokud tomu tak není, je uživatel vyzván k doplnění. Při startu programu se dále načítají informace pro lemmatizaci ze souboru `koncovky.txt`, pokud takový soubor existuje.

Soubor `homonyma.txt` je ke běhu programu nezbytný, v *.ini souborech jsou zapsány výsledky trénování a nejsou pro běh programu nezbytné (bez nich však natrénované algoritmy nebudou mít žádné znalosti). Soubor s koncovkami není pro správný běh programu nutný.

Po této inicializaci je program schopen pokračovat v trénování, nebo zahájit rozpoznávání.

Pokud zvolíme trénování algoritmů, dojde k vyzvání uživatele, aby vybral trénovací soubory. Po vybrání jsou testovací soubory otevřeny a je zahájeno trénování. Program postupně prochází trénovací data a pokud narazí na známé vícevýznamové slovo, dotáže se uživatele na význam slova. Ke snazšímu rozhodnutí mu zobrazí kontext okolo daného slova. Poté dojde k upravení informací ve struktuře *homonyma*. Takto program prochází iterativně všechna slova a všechny testovací soubory.

Nechceme-li dále trénovat nebo se nám zdá, že už je program dostatečně natrénován, můžeme přejít k rozpoznávání. Při rozpoznávání dojde opět k dotazu na uživatele, aby zvolil soubor k rozpoznání, a opět dojde k vyhledávání vícevýznamových slov. Pokud je nalezeno, je předán všem implementovaným metodám kontext daného slova a rozpoznávané slovo. Implementované metody poté zajistí výpočet a program zapíše výsledek jako prostý text do výstupního souboru.

Poslední částí programu je soubor nastavení. Zde je možnost nastavit volitelné parametry klasifikátorů.

Jelikož program pracuje s rozsáhlými daty, je časově i prostorově dosti náročný. Při implementaci byla snaha tuto náročnost co nejvíce omezit, přesto však zůstává několik míst, kdy výpočty či jiné operace trvají poměrně dlouhou dobu. Jedná se zejména o start programu a následně o fázi učení klasifikátorů. Optimalizací bylo

dosaženo, že tato operace probíhá vždy pouze před tím, než je daný klasifikátor použit a během jednoho spuštění programu pouze jednou. Přesto však paměťová náročnost programu při použitých datech je cca 100 MB a čas spuštění programu na počítači s procesorem 2,1 GHz je cca 45 s.

## 4.5 Zpracování výsledků jednotlivých metod

Celkový klasifikátor se, jak již bylo popsáno, skládá z pěti nezávislých částí. Každá část poskytuje samostatný výstup v podobě klasifikace slovního významu. Dále je součástí výstupu každé klasifikační metody ještě číslo, vyjadřující skóre daného klasifikátoru. Toto číslo v jeho hrubé podobě, jak jej poskytuje daná metoda, nevyjadřuje obecně pravděpodobnost či míru důvěry v rozhodnutí daného klasifikátoru na společné stupnici všech metod. Interpretace tohoto čísla je dána použitou metodou. Provést srovnání kvality či odvození celkového výsledku pouze pomocí výsledných čísel těchto metod je značně problematické. V následujících odstavcích budou popsány výsledné hodnoty jednotlivých metod a jejich interpretace.

Naivní Bayesův klasifikátor má tento číselný výstup roven logaritmické hodnotě pravděpodobnosti. Hodnota pravděpodobnosti u této metody vychází zpravidla velmi malá (v řádech $10^{-10}$ i méně) a teoreticky může nabývat hodnot 0 až 1 (pro logaritmické měřítko tedy $-\infty$ až 0). Čím větší je číslo, tím je jistější výsledek klasifikátoru.

Klasifikační algoritmus AdaBoost vrací číselnou hodnotu nejlépe klasifikovaného výsledku. Výsledná hodnota vyjadřuje naklonění klasifikátoru pro určení dané třídy. Tento rozsah je tedy (teoreticky) $-\infty$ až $\infty$, nejčastěji však se pohybuje kolem nuly. Čím více je tato hodnota kladná, tím jistější by měl být výsledek tohoto klasifikátoru.

Metoda maximální entropie vrací obecně hodnoty v intervalu $\langle -\infty, 0 \rangle$. Hodnota zcela závisí na počtu slov v kontextu, proto se nedá určit častý rozsah hodnot. Opět i zde platí, že čím vyšší entropie, tím důvěryhodnější je výsledek.

Metoda klasifikačního stromu je, jak již bylo napsáno, specifická. Tato metoda poskytuje výsledek, pouze pokud si je velmi jistá (v kontextu bylo nalezeno slovní spojení či slovo typické pro daný význam). Tato metoda jako jediná nemá za výsledek žádný mezistav. Buď je význam potvrzen (výsledkem je daný význam), nebo nepotvrzen (výsledkem je odpověď „nevím“).

Poslední implementovaná metoda – naivní – poskytuje výsledek, který je principielně podobný metodě AdaBoost. Dalo by se říci, že tato metoda je odlišná od AdaBoostu pouze v několika drobnostech (chybějící váhy, naivní výpočet). Poskytuje tedy stejný interval výsledných hodnot a stejnou interpretaci výsledku.

Jakým způsobem se pracuje s výsledky jednotlivých metod, je popsáno v následující kapitole.

## 4.6 Nastavení parametrů systému

Možností, jak ovlivnit výsledky systému, je nastavení některých parametrů systému. Jedná se zejména o koeficient pro Laplaceovo vyhlazování, nastavení významu (vah)

jednotlivých implementovaných metod a koeficienty pro práci s kontextem. Pro dosažení co nejlepších výsledů bylo nutné tyto koeficienty experimentálně určit. První stanovení hodnot těchto koeficientů bylo víceméně náhodné, nicméně s jistým ohledem na skutečný význam jednotlivých koeficientů. Tento odhad byl poté v testech použit jako základní sada parametrů, ze které se při jednotlivých krocích testování měnil vždy jen jeden parametr. Nejlepší výsledek daného parametru byl poté použit v celkovém systému. Testování probíhalo vždy na stejných datech z d-testovacích dat.

Testovaly se tyto parametry: Konstanta pro Laplaceovo vyhlazování $\mu$, délka kontextu rozpoznávaného slova $d$, počet výskytů slov v natrénovaném kontextu, nutných pro zahrnutí do výpočtu n a nakonec rozložení vah jednotlivých metod $w$ ($w_B$ pro Naivní Bayesův klasifikátor, $w_A$ pro AdaBoost, $w_E$ pro metodu maximální entropie a $w_N$ pro Naivní klasifikátor). Počáteční nastavení těchto parametrů je v tabulce 4.2.

| **parametr** | **hodnota** |
|---|---|
| μ | 0,001 |
| d | 20 |
| n | 1 |
| $w_B$ | 0,25 |
| $w_A$ | 0,25 |
| $w_E$ | 0,25 |
| $w_N$ | 0,25 |

*Tabulka 4.2: Počáteční nastavení parametrů pro klasifikaci významů*

Konstanta $\mu$ má pro výpočty významný vliv. Určuje chování klasifikátoru v případech, kdy nalezne slovo, které v natrénovaném kontextu nemá žádný výskyt. Pokud by byla tato hodnota příliš malá, klasifikátor by podle tohoto slova mohl zamítnout okamžitě třeba i správný význam, protože by rozhodnutí podle tohoto slova silně posílilo záporný vliv dané třídy. Naopak vysoká hodnota způsobí, že takto vybraná slova budou mít velký význam, i přesto, že o nich klasifikátor v podstatě nic neví. Výsledky testu jsou v tabulce 4.3.

| μ | Naivní Bayesův klasifikátor | Klasifikátor AdaBoost | Metoda maximální entropie | Naivní klasifikátor | Celkový výsledek |
|---|---|---|---|---|---|
| 0,001 | 83,33% | 84,91% | 60,42% | 84,38% | 84,38% |
| 0,01 | 84,38% | 86,54% | 59,24% | 85,42% | 84,13% |
| 0,1 | 62,21% | 66,00% | 53,18% | 63,37% | 62,22% |

*Tabulka 4.3: Výsledky jednotlivých klasifikátorů pro proměnný parametr konstanty pro Laplaceovo vyhlazování*

Délka kontextu *d* nemá na výsledky zas tak významný vliv, tedy pokud nejsou nastaveny extrémně nízké nebo příliš vysoké hodnoty. U nízkých hodnot je problém, zvláště v českém jazyce, že slovosled není přesně určen a tedy slova, určující význam slova se můžou nacházet na opačném konci věty, než je námi rozpoznávané slovo. Naopak velký rozsah kontextu již obsahuje slova, která se k rozpoznávanému slovu již nevztahují a negativně tak ovlivňují výsledek. Pro hodnoty parametru *d* v intervalu od 10 do 30 jsou rozdíly ve výsledcích testu zanedbatelné, jak je vidět v tabulce 4.4.

| d | Naivní Bayesův klasifikátor | Klasifikátor AdaBoost | Metoda maximální entropie | Naivní klasifikátor | Celkový výsledek |
|---|---|---|---|---|---|
| 10 | 83,33% | 84,91% | 60,42% | 84,38% | 84,38% |
| 20 | 84,38% | 86,54% | 60,83% | 85,42% | 84,38% |
| 30 | 85,12% | 88,22% | 60,03% | 83,92% | 84,73% |
| 40 | 85,12% | 87,34% | 60,42% | 84,38% | 83,97% |
| 50 | 84,38% | 88,22% | 60,42% | 84,38% | 83,97% |
| 100 | 82,88% | 84,36% | 60,03% | 83,23% | 83,56% |

*Tabulka 4.4: Výsledky jednotlivých klasifikátorů pro proměnný parametr délky kontextu*

Počet výskytů natrénovaných slov v kontextu *N* je další parametr k odhadu, jehož nastavení mohlo vylepšit systém. Jeho cílem bylo při hodnotách větších než jedna odfiltrovat z kontextu taková slova, která se ke klasifikovanému slovu dostala náhodou a je nepravděpodobné, že se vztahují k danému slovu. Pokud je však nastaven na příliš vysokou hodnotu, klasifikátor dostane málo informací o kontextu a není schopen zpravidla správně určit výsledek, protože se jeho rozhodování omezuje na nejčastěji vyskytované slovo, podle kterého rozhoduje. Při odhadu tohoto parametru byla

limitující celková velikost trénovacích dat. Díky jejich relativně malé velikosti byly výsledky s parametrem N větší než 3 značně nepoužitelné, jak napovídá tabulka 4.5.

| N | Naivní Bayesův klasifikátor | Klasifikátor AdaBoost | Metoda maximální entropie | Naivní klasifikátor | Celkový výsledek |
|---|---|---|---|---|---|
| 1 | 83,33% | 84,91% | 60,42% | 84,38% | 84,38% |
| 2 | 84,38% | 84,22% | 56,43% | 70,52% | 76,67% |
| 3 | 83,33% | 81,22% | 53,62% | 67,35% | 73,16% |

*Tabulka 4.5: Výsledky jednotlivých klasifikátorů pro proměnný parametr minimální počet výskytů slov v natrénovaném kontextu*

Pro nastavení vah *w* byla použita jiná metodika testu. Prvním krokem bylo určit, jak přesné jednotlivé metody poskytují výsledky a zda vůbec jsou mezi nimi rozdíly natolik výrazné, aby nestačilo stanovit celkový výsledek klasifikace jako výsledek většinového rozhodnutí použitých klasifikátorů – tedy použití stejných vah. K tomuto účelu byla vytvořena sada testů, která měla pomoci rozhodnout, zda rozdíl mezi jednotlivými metodami je statisticky významný. Každá sada testů obsahovala 20 dokumentů s vícevýznamovými slovy. Počet vícevýznamových slov v dokumentech byl různý (jednalo se o reálné texty). Na každé takové sadě se provedlo rozpoznání významů pomocí jednotlivých metod. Výsledek testu je zobrazen v tabulce 4.6.

| měření | Naivní Bayesův klasifikátor | AdaBoost | Metoda maximální entropie | Naivní metoda |
|---|---|---|---|---|
| 1 | 92,31% | 88,89% | 80,77% | 96,15% |
| 2 | 85,71% | 70,00% | 76,19% | 95,24% |
| 3 | 80,77% | 77,78% | 73,08% | 76,92% |
| 4 | 76,92% | 90,00% | 73,08% | 84,62% |
| 5 | 92,31% | 100,00% | 80,77% | 84,62% |
| 6 | 76,92% | 90,00% | 73,08% | 76,92% |
| 7 | 96,15% | 100,00% | 76,92% | 92,31% |
| 8 | 84,62% | 95,45% | 76,92% | 84,62% |
| 9 | 76,92% | 93,75% | 80,77% | 73,08% |

*Tabulka 4.6: Úspěšnost jednotlivých metod při testovacích měřeních*

Takto získaná data z klasifikace byla pak podrobena statistickému testu. Je nutné zjistit spolehlivost jednotlivých metod. Pro odhad úspěšnosti byl použit intervalový odhad spolehlivosti metod na hladině spolehlivosti 95%. Interval spolehlivosti je dán vzorcem:

$$\left( \bar{x} - t_{1-\frac{a}{2}} \frac{s}{\sqrt{n}}, \bar{x} + t_{1-\frac{a}{2}} \frac{s}{\sqrt{n}} \right)$$

Kde x jsou naměřené hodnoty, t je kvantil studentova rozdělení na hladině spolehlivosti α o n-1 stupních volnosti, σ je směrodatná odchylka a n je počet hodnot. Pro konkrétní případ byla brána hodnota t = 2,306.

Výsledky tohoto odhadu jsou v tabulce 4.7.

| **klasifikátor** | **dolní hladina intervalu** | **horní hladina intervalu** |
|---|---|---|
| Naivní Bayesův klasifikátor | 79,33% | 90,14% |
| AdaBoost | 82,33% | 96,75% |
| Metoda maximální entropie | 74,42% | 79,26% |
| Naivní metoda | 78,90% | 90,98% |

*Tabulka 4.7: Intervalový odhad spolehlivosti jednotlivých metod*

Z tabulky 4.7 je zřejmé, že úspěšnost metod (kromě metody maximální entropie) je víceméně totožná. Váhy jednotlivých metod tedy budou nastaveny rovnoměrně, pouze metoda maximální entropie bude mírně znevýhodněna.

Z výsledků testu je patrné, že parametry programu lze docílit různých výsledků klasifikace. V případě mého programu má na systém největší vliv parametr N, což bylo způsobeno faktem, že trénovací data nemají potřebnou velikost a tudíž neposkytují mnoho výskytů víceznačných slov. Ruku v ruce jde tato skutečnost i s faktorem, že český jazyk neobsahuje víceznačných slov tolik, jako např. anglický jazyk.

# Kapitola 5

# Vyhodnocení výsledků

V této kapitole si popíšeme výsledky systému a jejich vyhodnocení. Budou zde předvedeny metriky sloužící k vyhodnocení systémů pro zjednoznačňování slovních významů. Ukážeme si také příklad chybného určení významu, na kterém si předvedeme slabé stránky systému a navrhneme některá možná vylepšení. Nejprve si popíšeme způsoby, jakými lze vyhodnotit systémy pro zjednoznačňování významů.

## 5.1 Metodiky vyhodnocení

Obecně je pro vyhodnocení jakýchkoli systémů vhodné použít alespoň několik metod. Pro obor zpracování přirozeného jazyka jsou pro některé obory doporučeny různé metody vyhodnocení. Bohužel u zjednoznačňování slovních významů tomu tak není. Pro vyhodnocení úspěšnosti může použít každý řešitel tohoto problému libovolný způsob, který mu přijde vhodný.

### 5.1.1 Vyhodnocení založené na počtu správně určených významů

Nejjednodušší způsob, jak určit úspěšnost systému, je vyhodnocení počtu správně určených významů v závislosti na všech určovaných významech. Úspěšnost systému je dána vzorcem:

*úspěšnost = počet správně určených významů / počet všech určovaných významů*

Tato velice jednoduchá metoda vyhodnocení dokáže vytvořit základní obrázek o úspěšnosti vyhodnocovaného systému. Tím, že je tato metoda silně intuitivní, je zpravidla používána jako první. Má však jistá úskalí. Ukážeme si je na následujícím případě.

Představme si, že máme určit význam slova *w* ve větě. Víme, že slovo *w* může mít jeden ze tří významů: $s_1$,$s_2$ a $s_3$. K dispozici máme celkem 4 systémy (označme si je A, B, C a D). Předpokládejme, že správný význam je $s_1$. Systémy poskytly následující pravděpodobnosti jednotlivých významů (tabulka 5.1):

| **systém** / **význam** | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| $s_1$ | 41% | 32% | 10% | 0% |
| $s_2$ | 44% | 36% | 80% | 100% |
| $s_3$ | 15% | 32% | 10% | 0% |

*Tabulka 5.1: Pravděpodobnosti určení významu u čtyř teoretických systémů*

Z tabulky je patrné, že za předpokladu správného významu $s_1$ ani jeden systém neurčil správný význam. Podle výše uvedeného kriteria by tedy všechny systémy měly úspěšnost 0%. Pokud však budeme blíže zkoumat jednotlivé výsledky, zjistíme, že systému A nechybělo málo, aby správný význam určil a naopak systém D správný význam stoprocentně zamítl. Systém C taktéž silně preferoval odlišný význam a systém B poskytuje značně podobné výsledky pro všechny významy. Nabízí se tedy možnost vyhodnocovat systémy podle toho, jakou důvěru mají ve správný výsledek. Tato důvěra nemusí být přímo pravděpodobnost správného určení významu, pro vzájemné srovnání metod je však vhodné, aby jejich výsledky byly mezi sebou srovnatelné.

Pokud bychom chtěli vybudovaný systém použít jako vstup pro další systémy, je dokonce velmi vhodné, aby výsledek určení významu obsahoval nějaké číslo – míru důvěry, které je nějakou matematickou funkcí převoditelné na pravděpodobnost.

Výše předvedený příklad ukázal, že určení úspěšnosti pomocí počtu správně určených významů není příliš vhodný k vyhodnocení systému. Přesto tímto způsobem budou vyhodnoceny některé části implementovaného systému z důvodu porovnání výsledků s dalšími metodami vyhodnocení.

### 5.1.2 Vyhodnocení založené na křížové entropii

Tato metoda vyhodnocení řeší problémy předchozí metody s mírami důvěry v jednotlivé výsledky. Vyhodnocení je provedeno podle následujícího vzorce

$$acc = -\frac{1}{N}\sum_{i=1}^{N}\log_2 P_s(cs)$$

kde *acc* je úspěšnost systému, *N* je počet testovaných instancí, $P_s$ je pravděpodobnost přiřazená správnému významu slova *cs*. Tento způsob ohodnocení zvýhodní ty systémy, které poskytnou vyšší důvěru ve správné výsledky. Vyhodnocení pomocí této metody je v tabulce 5.2.

| systém | acc |
|---|---|
| A | 1,29 |
| B | 1,64 |
| C | 3,32 |
| D | ∞ |

*Tabulka 5.1: Entropie pro správný význam u čtyř teoretických systémů*

Je nutné si uvědomit, že nejlepší ohodnocení je takové, které má v tabulce 5.2 nejmenší hodnotu. Přesně podle očekávání získal systém A nejlepší ohodnocení, naopak systém D, který správný význam zamítl se stoprocentní pravděpodobností, nejhorší.

Jeden z problémů znemožňující použití této metody vyhodnocení je fakt, že ne všechny systémy vrací společně s určeným významem i míru důvěry.

## 5.2 Rozlišení sémanticky blízkých významů při vyhodnocení

V přirozeném jazyce je množství vícevýznamových slov, u nichž jsou jednotlivé významy sémanticky blízké. Tato vícevýznamová slova je tedy vhodné kvůli vyhodnocení uspořádat do hierarchické struktury podle sémantiky. Ukázkou takové struktury může být tabulka 5.4 pro slovo *třída*.

| **slovo** | **skupina významu** | **význam** | |
|---|---|---|---|
| třída | I | I 1 | společenská vrstva |
| | | I 2 | skupina se společnými znaky |
| | | I 3 | vyučovací stupeň |
| | | I 4 | vyučovací místnost |
| | | I 5 | stupeň, kategorie |
| | II | II 1 | ulice |

*Tabulka 5.4: Hierarchická struktura sémanticky blízkých významů pro slovo třída*

Tato struktura může pomoci při vyhodnocení chybných určení systému tím, že sníží váhu chybného zařazení do stejné skupiny významu a posílí váhu chybného zařazení do jiné skupiny. Váhu chyby pro každý konkrétní případ je možné nastavit penalizační maticí (viz kapitola 3.1). Matice pro slovo třída může vypadat takto (tabulka 5.5):

| **určený význam** / **správný význam** | **I 1** | **I 2** | **I 3** | **I 4** | **I 5** | **II 1** |
|---|---|---|---|---|---|---|
| I 1 | 0 | 1 | 1 | 1 | 1 | 2 |
| I 2 | 1 | 0 | 1 | 1 | 1 | 2 |
| I 3 | 1 | 1 | 0 | 1 | 1 | 2 |
| I 4 | 1 | 1 | 1 | 0 | 1 | 2 |
| I 5 | 1 | 1 | 1 | 1 | 0 | 2 |
| II 1 | 2 | 2 | 2 | 2 | 2 | 0 |

*Tabulka 5.5: Penalizační matice pro slovo třída*

Systémy pro zjednoznačnovaní by obecně měly rozlišovat slova až na úroveň významů (v tzv. jemné granularitě výsledků). Jak je však patrné z tabulky 5.4, některé významy (např. konkrétně I 3 a I 5) jsou si velmi podobné a dochází k častému zaměňování výsledků, neboť i slova v kontextech bývají podobná. Tento fakt pak samozřejmě vnáší další chybu do systému a snižuje jeho úspěšnost. Často se tento problém řeší tím, že se provede přetvoření skupin významů tím, že se některé blízké

významy sloučí do jedné skupiny (tento způsob se nazývá hrubá granularita výsledků). Nikde však není exaktně uvedeno, jaké významy a do kolika skupin se mají sloučit, zde záleží hodně na citu toho, kdo systém navrhoval.

## 5.3 Výsledky implementovaného systému

V této části budou popsány výsledky mnou navrženého a implementovaného systému. Čtenář se dozví celkové úspěšnosti jednotlivých metod, výsledek celku a možné porovnání s ostatními systémy pro WSD.

Systém byl implementován tak, aby poskytoval výsledky pouze v hrubé granularitě. To je dáno především použitými daty, které nepodporují jemnou granularitu vyhodnocení.

Systém byl testován na dvou sadách. První použitá sada byla data z PDT 2.0 ze skupiny dat dtest. Pro tato data dosáhl systém 86,75% úspěšnosti. K vyhodnocení byla použita metoda založená na počtu správně určených významů, neboť systém jako celek neumí vyčíslit důvěru v řešení. Toto dokážou pouze jednotlivé použité metody, a to ještě ne všechny. Výsledky pro jednotlivé metody jsou uvedeny v tabulce 5.6.

| | |
|---|---|
| naivní Bayesův klasifikátor | 87,25% |
| AdaBoost | 88,93% |
| metoda maximální entropie | 83,50% |
| naivní klasifikátor | 83,50% |
| **CELKEM** | **86,75%** |

*Tabulka 5.6: Celkové výsledky jednotlivých metod*

Druhá použitá sada vycházela z náhodně posbíraných dokumentů. Myšlenka pro využití odlišných testovacích dat byla motivace vyzkoušet systém, jak si poradí s daty, na které nebyl implicitně naučen. Při prohlížení dat z PDT 2.0 bylo zjištěno, že mnoho dokumentů se týká podobných témat, tudíž určování významů by mohlo být ovlivněné touto datovou korelací. S takovými daty si systém poradil s velmi podobnou úspěšností 85,33%. I v tomto případě bylo použito vyhodnocení založené na počtu správně určených významů. Druhá testová sada obsahovala jen 15 dokumentů (namísto 179 v předchozím případě), takže síla tohoto testu je přirozeně nižší, než u první datové sady. Ale i přesto lze z porovnání těchto dvou výsledků tvrdit, že systém byl natrénován poměrně nezávisle na datech.

Výsledky pro metodu rozhodovacího stromu byly příliš malé, aby se z nich dal udělat nějaký rozumný závěr. Z této malé skupiny však lze říci, že funkčnost implementovaného rozhodovacího stromu je velmi dobrá, když pokrývá pouze velice úzkou skupinu dat.

### 5.3.1 Srovnání výsledků

Ekvivalentní srovnání výsledků s jinými systémy pro zjednoznačnování slovních významů bude v případě této práce mnohem obtížnější a hůře interpretovatelné, než ve většině doposud prezentovaných prací. Kámen úrazu je v použitých datech. Tento systém byl již od začátku vytvářen pro český jazyk, na rozdíl od většiny systémů pracujících s jazykem anglickým. Proto také většina srovnání existuje pro systémy jazyka anglického. Výsledky libovolného systému pro český jazyk se mi nepodařilo nikde nalézt. Není mi dokonce ani známo, že by se někdo problémem zjednoznačňování českých slov zabýval.

Pokusíme-li se abstrahovat od jazyka a zkusíme porovnávat pouze výsledné hodnoty (ačkoliv to bude mít malou vypovídající hodnotu), pak podle výsledků konference Semeval z roku 2007 by se systém s 86,75 procentními body umístil na 3. místě ze 13. Znovu je však nutné připomenout, že takto provedené srovnání nemá příliš velký význam.

### 5.3.2 Zamyšlení nad výsledky

Při podrobném prozkoumání výsledků z tabulky 5.5 je zřejmé, že celkový výsledek systému je horší, než výsledek nejlepších použitých metod. To je dáno několika faktory.

Prvním z nich je výběr metod. V práci byly použity dvě relativně slabé metody (Metoda maximální entropie a Naivní Metoda), které mohly způsobit zhoršení výsledků. Jejich použití bylo vhodné spíše pro srovnání s jinými metodami. Pokud tyto dvě metody nebudou do celkového systému zahrnuty a výsledek bude stanoven pouze pomocí Naivního Bayesova klasifikátoru a klasifikátoru AdaBoost, dosáhne systém o půl procentního bodu lepší skóre, tedy 87,25%.

Dalším faktorem byl klasifikátor AdaBoost. Ačkoli dle tabulky 5.5 dosahoval nejlepších výsledků, samotný klasifikátor v některých případech nedokázal určit žádný význam jako pravděpodobný. To nastalo hlavně buď v případě, kdy ostatní klasifikátory rozhodovaly pouze na základě pravděpodobnosti výskytu daného slova (tedy bez příznakového vektoru), nebo v případě klasifikace pouze pomocí několika málo klasifikátorů, poskytujících velmi vyrovnané výsledky. Případů, kdy klasifikátor AdaBoost nebyl schopen určit výsledek, bylo v testovacích datech poměrně velké procento, celkem 36,75%.

Jádro výkonnějšího systému by tedy mohl tvořit Naivní Bayesův klasifikátor, který by byl podpořen klasifikátorem AdaBoost. Metoda maximální entropie by mohla plnit pouze podpůrnou roli, která by buď povyšovala důvěru, pokud by rozhodla shodně s ostatními klasifikátory, nebo by neměla na celkový výsledek žádný vliv, pokud by její rozhodnutí bylo různé. Naivní metoda by v takovém systému neměla mít vůbec místo. Implementována byla pouze jako první pro poskytnutí srovnání.

### 5.3.3 Příklad chybně určeného významu

Zaměřme se nyní na chybně určené významy. Chyby systému jsou dvojího druhu. Prvním druhem chyby je nedostatečný trénink klasifikátorů. Tato chyba je způsobena nepřítomností žádného klíčového slova v kontextu rozpoznávaného slova. Na všech chybně určených významech se tato chyba podílela 24,5%. Řešením by bylo použít širší kontext daného slova, než je nastavená hodnota, příp. kontext dynamický. Globálně je ale širší kontext nežádoucí, protože vede nejen k delšímu zpracování a vyšší náročnosti systému, ale hlavně k zanášení slov, které mohou potenciálně patřit úplně jinému významu.

Druhým typem chyby je vlastní chybná klasifikace. Klasifikátor sice dostane dostatečný počet dat, ale tato data nejsou vhodná. Zde jednoduché řešení neexistuje.

Příkladem takového chybného určení může být snaha o určení významu slova *stát* v následujícím kontextu. Slovo *stát*, alespoň tak, jak bylo systémem naučeno, může mít tyto významy: *stát(sloveso)*, *stát(podst. jméno)*.

*… vyhynuli jako všechny druhy živých i neživých systémů, které se nedokážou přizpůsobit prostředí, v němž působí. Krmení si nakonec opatřili. Přehrada téměř stojí. Turbíny budou taky stát. Alespoň to přislíbila federální vláda: ústy náměstka ministra zahraničních věcí Zdenka Pírka dala najevo, že když na tom budou ostatní země trvat, nezačnou turbíny vyrábět…*

Z textu je hned na první pohled zřejmé, že šířka kontextu je v tomto případě značně irelevantní. Relevantní informace je pouze v předcházejících šesti slovech v kontextu. Práce se šířkou kontextu bude v tomto případě k ničemu. Navíc další část kontextu je příznačná spíše pro význam *stát(podst. jméno)*. Není se tedy čemu divit, že systém selže. Výsledky jednotlivých metod pro tento případ jsou uvedeny v tabulce 5.6. Metoda klasifikačního stromu nedokázala odpovědět, proto není do výsledku zahrnuta.

| | **stát(sloveso)** | **stát(podst. jméno)** |
|---|---:|---:|
| naivní Bayesův klasifikátor | -53,236 | -29,276 |
| AdaBoost | 4,840 | 30,550 |
| metoda maximální entropie | -2,086 | -1,567 |
| naivní klasifikátor | -4,041 | 4,041 |

*Tabulka 5.6: Skóre jednotlivých metod při selhání určení významu*

Přesně podle všech předpokladů, ani jeden klasifikátor neurčil význam správně.

Podívejme se, jak k tomuto selhání došlo. Systém v prvním kroku vybral z kontextu klíčová slova. Jsou to slova:

*jako všechny které němž nakonec stojí budou taky vláda najevo když tom budou země*

Tabulka pravděpodobností výskytu těchto klíčových slov v závislosti na významu je následující (tabulka 5.7):

| | stát(sloveso) | stát(podst. jméno) |
|---|---|---|
| **celkem výskytů** | **31,97%** | **68,03%** |
| jako | 14,29% | 85,71% |
| všechny | 0,00% | 100,00% |
| který | 50,00% | 50,00% |
| jenž | 33,33% | 66,67% |
| nakonec | 0,00% | 100,00% |
| stojí | 0,00% | 100,00% |
| budou | 100,00% | 0,00% |
| taky | 100,00% | 0,00% |
| vláda | 0,00% | 100,00% |
| najevo | 0,00% | 100,00% |
| když | 0,00% | 100,00% |
| tom | 100,00% | 0,00% |
| budou | 100,00% | 0,00% |
| země | 0,00% | 100,00% |

*Tabulka 5.7: Pravděpodobnosti vybraných klíčových slov pro případ chybně určeného významu*

Z tabulky jsou evidentní dvě věci: Zaprvé je vidět, že výběr klíčových slov obsahuje značné množství slov, které nenesou význam věty (jedná se především o zájmena *jako*, *které*, *němž*, *taky* a *tom*). Jejím odstraněním však nic nezískáme, právě spíš naopak. Pravděpodobnosti těchto slov jsou poměrně vyvážené, nebo dokonce přispívají kladně ke správnému významu. Mnohem markantněji se zde prosadil neoptimální výběr trénovacích dat. Jak již bylo napsáno, trénovací data byla vybírána z obecného korpusu a v tomto případě nám nikdo nezaručí, že data budou pro trénování vhodná. Nabízí se otázka, zda byly klasifikátory natrénovány dostatečně. Trénovací množina měla velikost pouze cca 364 tisíc slovních jednotek, což může být pro takový systém málo. Korpusy pro český jazyk ještě zdaleka nedosahují úrovně co do počtu dat, jak korpusy pro jazyk anglický. Navíc přístup k českým korpusům je mnohem obtížnější, než ke korpusům anglickým, a proto je obtížné sehnat dostatek trénovacích dat.

## 5.4 Návrh na vylepšení systému

V předchozí kapitole se objevilo několik myšlenek, jak odstranit některé chyby výsledků a nebo je aspoň eliminovat. V této části budou myšlenky dále rozvinuty.

První myšlenkou je použití různě širokého kontextu. Pokud by systém nenalezl v daném kontextu dostatečný počet klíčových slov, rozšířil by vyhledávání v kontextu

o určitý počet slov. Takto by postupoval, dokud by nenalezl předen stanovený minimální počet klíčových slov. V tomto případě by však musel být implementován lepší způsob výběru klíčových slov, než jen pouhé vyhledávání v natrénovaných kontextech, který by odstranil klíčová slova nepodílející se na významu věty (viz např. zájmena v příkladu chybného určení z kap. 5.3.3).

Další možností by bylo rozšířit trénovací data použitím dalších korpusů. Pro český jazyk se nabízí použití např. českého národního korpusu, který je mnohem rozsáhlejší než použitý korpus PDT 2.0. Větší rozsah korpusu umožní pokrýt lépe více případů víceznačných slov v jazyce.

Účinnou možností, jak vylepšit daný systém, je lepší výběr klíčových slov. Zde se nabízí možnost syntaktické analýzy k zjištění vazeb mezi slovy. Je pravděpodobné, že slovo bezprostředně se vázající na rozpoznávané slovo bude lépe rozlišovat daný význam.

Pro vylepšení systému by se jistě dalo nalézt mnoho dalších triků, jak zlepšit výsledky, výše uvedené nápady poskytují pouze několik náhledů na vylepšení.

# Závěr

Zpracování přirozeného jazyka prošlo od poloviny minulého století, kdy se začaly používat počítače, značným vývojem. Nejinak tomu bylo s problémem zjednoznačňování slovních významů. Již v počátcích bylo vymyšleno několik algoritmů, které tento problém dokázaly řešit. Bohužel v té době však tyto metody narážely na nízký výpočetní výkon počítačů, a tak se oblast zpracování přirozeného jazyka dostala do popředí až v posledních letech, kdy je výkon počítačů již dostačující pro zpracování rozsáhlých dat. V současnosti se používají jako zdroj dat pro zjednoznačňování slovních významů rozsáhlé, zatím stále ručně anotované, soubory dat, zvané korpusy. V posledních letech je snaha o automatizaci tohoto anotování. Aby mohla být anotace úspěšná, musí být vyvinuty kvalitní algoritmy, které dokážou rozhodnout nejen význam slova.

Hlavní cíl této práce byl ve vytvoření systému, který na základě natrénovaných dat dokáže určit význam vícevýznamového slova v českém jazyce. K tomu bylo implementováno několik algoritmů, které pomáhají rozhodnout mezi jednotlivými významy a jsou nosnou částí celé práce. Jedná se zejména o Naivní Bayesův klasifikátor, který na základě pravděpodobností výskytu klíčových slov určuje s jistou pravděpodobností jeden z významů. Dále byl implementován klasifikační algoritmus AdaBoost, který rozhoduje na základě tzv. slabých klasifikátorů, vytvořených z klíčových slov. Další použitou metodou byla metoda maximální entropie, vycházející z matematického modelu pro náhodné rozložení. Pro klasifikaci ustálených slovních spojení byl implementován rozhodovací strom, který doplňuje tyto tři metody. Poslední metodou byla jednoduchá metoda srovnávající pravděpodobnosti klíčových slov. Všechny metody jsou v práci popsány a názorně vysvětleny na příkladech.

Přínos práce jednoznačně spočívá ve vytvoření systému, který pracuje s českým jazykem. Nepodařilo se najít jiný systém, který by využíval česká data.

Výsledky systému nejsou nijak špatné ve srovnání se systémy pracující s anglickým jazykem. Dosažená úspěšnost 86,75%, která může být vhodným kombinováním metod ještě o několik desetin procenta zvýšena, je slušným výsledkem. Za takto vysokou úspěšností stojí ale zejména fakt, že český jazyk má mnohovýznamových slov relativně málo. Navíc vícevýznamová slova v češtině mají ve velkém procentu případů jeden hlavní význam, který je používán nejčastěji.

Přesto však systém stále nedosáhl stropu svých možností. Největším problémem je dostatek kvalitních testovacích dat. To je způsobeno hlavně faktem, že systém pro český jazyk využívá českých textů, kterých je ve srovnání např. s angličtinou poměrně málo a tedy i trénink jednotlivých klasifikátorů není úplně dostatečný.

Jedno je však jisté. Budovat systém pro zjednoznačňování slovních významů tak, aby byl použitelný, stojí více práce, než si člověk dokáže představit a každý, byť nepatrný posun, stojí mnoho a mnoho drahocenného času.

# Použitá literatura


[1] Jean Véronis Nancy Ide. Word Sense Disambiguation: The State of the Art. 1998

[2] Lubomír Otrusina. Strojové učení v přirozeném jazyce. Bakalářská práce: Brno. 2007

[3] www stránky. Homonym. http://en.wikipedia.org/wiki/Homonym

[4] Luc Devroye, Laszlo Gyorfi, Gabor Lugosi. A Probabilistic Theory of Pattern Recognition. New York 1996

[5] Yoav Freund, Robert E. Schapire. A Short Introduction to Boosting. http://www.site.uottawa.ca/~stan/csi5387/boost-tut-ppr.pdf

[6] Wojciech Skut, Thorsten Brants. A Maximum-Entrophy Partial Parser for Unrestricted Text: Sixth Workshop on Very Large Corpora. Montreal, Québec 1998

[7] www stránky. Rozhodovací stromy. http://cs.wikipedia.org/wiki/Rozhodovac%C3%AD_stromy

[8] www stránky. Algoritmus ID3. http://datamining.xf.cz/view.php?cisloclanku=2002102803

[9] www stránky. Pražský závislostní korpus. http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/cz/html/ch03.html#a-data-formats-pml

[10] www stránky. Konference Senseval (Semeval). http://www.senseval.org/

[11] Jiří Anděl. Základy matematické statistiky. Praha 2006

# Přílohy

**Příloha 1: Struktura ini souborů pro jednotlivá vícevýznamová slova**

V ini souboru jsou tyto sekce a proměnné (pod jednotlivými sekcemi je vysvětlení obsahu jednotlivých proměnných):

```
[tvary]
Tvar1
Tvar2
…
TvarN
Minimalni
```

Pokud se jedná o sklonné slovo, jsou zde uvedeny všechny jeho tvary. Dále je uveden tvar minimální, tj. největší skupina hlásek, která se vyskytuje ve všech tvarech. Pokud se jedná o nesklonné slovo, sekce chybí.

```
[hlavni]
Tvaru
CelkemVyznamu
Vyskytu
```

V této sekci jsou hlavní proměnné pro všechny významy. Jedná se o počet tvarů v sekci tvary, počet významů slova a dále celkový počet výskytů slova ve všech doposud trénovaných dokumentech.

```
[kontext0]
Pocet
Vyskyty
[kontext1]
…
[kontextN]
```

Tyto sekce obsahují základní informace o kontextech pro všechny významy. V proměnné počet je uveden počet slov v kontextu a proměnná výskyt obsahuje počet výskytu slova v daném významu. Počet sekcí je dán proměnou CelkemVyznamu v sekci hlavni.

```
[kontext0.Slovo0]
Slovo
Pocet
Strom
[kontext0.Slovo1]
…
[kontextN.SlovoM]
```

Poslední nejrozsáhlejší sekce obsahují informace o všech slovech v daném kontextu. Proměnná *slovo* obsahuje konkrétní hodnotu slova (řetězec znaků) a proměnná *pocet*

obsahuje počet výskytů slova ve všech natrénovaných kontextech. Je-li uveden proměnná strom, je její hodnota vzdálenost pozice pro rozhodovací strom.

**Příloha 2: Lemmatizace a seznam českých koncovek**

Pseudoalgoritmus použitý pro lemmatizaci českých slov:

*Vstup: <slovo k lemmatizaci> slovo*
*Výstup: <lemmatizované slovo> lemma*
*for <všechny koncovky od nejdelší po nejkratší> k*
    *if <slovo> končí na <k> then*
        *<lemma> = <slovo> - <k>*
        *break*

Seznam českých koncovek v pořadí, v jakém je algoritmus vyhledává. Pokryty jsou koncovky podstatných jmen, přídavných jmen a sloves.

| | | | | | |
|---|---|---|---|---|---|
| atech | ata | ímu | ovi | ěl | ům |
| iných | aty | ina | ovo | em | ův |
| inými | ech | ině | ovu | ém | ým |
| ových | ého | ini | ovy | eš | u |
| ovými | eme | ino | ých | ím | a |
| atům | ému | inu | ými | in | e |
| etem | ete | iny | ám | me | i |
| ouce | eti | ouc | án | ou | y |
| ovým | ích | ova | at | te | o |
| iným | ího | ové | en | ši | ě |
| ách | ími | ově | el | še | |

**Příloha 3: Seznam všech rozpoznávaných vícevýznamových slov a jejich určované významy**

| | | | | |
|---|---|---|---|---|
| **abeceda** | znaky | základní pravidla | | |
| **admirál** | velitel lodi | motýl | | |
| **agent** | jednatel | náhončí | tajný policista | |
| **akademický** | týkající se akademie věd | vysokoškolský | vyučený | teoretický |
| **akademie** | háj v Aténách | vědecká instituce | škola | zábavný večer |
| **akademik** | člen akademie věd | absolvent nebo posluchač | | |
| **akord** | hudební souzvuk | úkolová práce | dohoda | |
| **akt** | čin | úřední spis | divadelní jednání | výtvarné dílo |
| **aktovka** | brašna | divadelní hra | | |
| **alkohol** | derivát uhlíku | nápoj | | |
| **alpaka** | lama | slitina mědi | | |
| **atlas** | sbírka map | obratel | tkanina | pohoří |
| **balík** | hanlivé označení | zásilka | neurčené množství | |
| **banda** | hudební těleso | skupina | | |
| **banket** | hostina | okrajový pás silnice | | |
| **bar** | zvýšený pult | jednotka tlaku | | |
| **barvička** | barva | barvící žena | | |
| **bašta** | opevněné místo | dobrý požitek | | |
| **bez** | mínus | šeřík | | |
| **biskup** | církevní hodnostář | kus drůbeže | | |
| **bob** | bylina | plod | saně | |
| **bolero** | tanec | vesta | | |
| **Boleslav** | mužské jméno | město | | |
| **bor** | chemický prvek | borovicový les | | |
| **box** | hovězí kůže | sport | oddělené místo | |
| **bratr** | sourozenec | v klášteře | | |
| **brnění** | rytířské | podráždění nervů | | |
| **březí** | březový porost | obřezlý | | |
| **burák** | rostlina | oříšek | | |
| **bylina** | rostlina | ruská píseň | | |
| **cenit** | odhadovat množství | odhalovat | | |
| **časovat_** | stanovit čas | ohýbat sloveso | | |

| | | | |
|---|---|---|---|
| **čelo** | část hlavy | hudební nástroj | začátek |
| **dav** | skupina lidí | sloveso | |
| **delta** | ústí řeky | trojúhelník | |
| **deka** | pokrývka | 10 gramů | |
| **dešťovka** | dešťová voda | žížala | |
| **dieta** | náhrada | výloh | výživa nemocných |
| **disk** | kotouč | harddisk | CD |
| **div** | zázrak | téměř | |
| **dlažka** | kostka | dlaha | |
| **dny** | nom.pl. den | gen.sg. dna | |
| **dojetí** | ukončení jízdy | citový stav | |
| **dokázat** | ukončit kázání | dosvědčit | podařit |
| **domácí** | podst.jméno | příd.jméno | |
| **dolní** | spodní | důlní | |
| **dovádět** | rozpustile skotačit | ukončovat vedení | |
| **dožít** | ukončit život | ukončit žetí | |
| **dračka** | žena deroucí peří | shon po něčem | tříska dřeva |
| **drb** | klep | šťouchnutí | |
| **dřevní** | dřevěný | dřívější | |
| **džin** | duch | kapela | lihovina |
| **fagot** | klení | hudební nástroj | |
| **granát** | cenný kámen | výbušná zbraň | |
| **háček** | malý hák | problém | diakritické znaménko |
| **haluz** | větev | štěstí | |
| **hlas** | zvuk | volba | |
| **hláska** | element řeči | strážní věž | |
| **hnát** | končetina | sloveso | |
| **hod** | vrh | svátek | hodiny |
| **hodit** | vrhnout | být vhodný | |
| **holí** | instr.sg. hůl | sloveso | příd.jméno |
| **holina** | holá oblast | gumová bota | |
| **hora** | kopec | velké množství | |
| **hoře** | žal | kopci | |
| **hospodský** | podst.jm. | příd.jméno | |
| **hradní** | podst.jm. | příd.jméno | |
| **hrana** | ostrý kraj | vyzvánění za zemřelého | |
| **hruška** | plod | strom | |
| **hřeben** | na česání | hor | |
| **hyperbola** | kuželosečka | nadsázka | |
| **chvat** | spěch | způsob uchopení | |
| **jak** | zájmeno | býložravec | |
| **jablko** | plod | v koleni | královské |
| **jahoda** | plod | keřík | |
| **jasan** | souhlas | strom | |

| | | | | |
|---|---|---|---|---|
| **jazyk** | orgán | řeč | | |
| **jeřáb** | stroj | strom | pták | |
| **jez** | podst. jméno | sloveso | | |
| **káča** | Kateřina | hanlivé označení | hračka | kachna |
| **kačka** | Kateřina | kachna | koruna | hračka |
| **kapsář** | zloděj | několik kapes | | |
| **kar** | pohřební hostina | ledovcový kotel | | |
| **kára** | vozík | vzor | | |
| **karma** | plynový ohřívač vody | buddhistický souhrn činů | | |
| **kartáč** | nástroj na odstraňování nečistot | druh dělové střely | souhrn výtek | |
| **Karolína** | jméno | stát USA | | |
| **kecka** | bota | mluvná žena | | |
| **klíč** | zamykací nástroj | k utahování matic | pomůcka k vysvětlení | |
| **klíček** | zamykací nástroj | k utahování matic | rostlinný | malý kel |
| **klika** | u dveří | štěstí | | |
| **klička** | malá klika u dveří | oko na provaze | uhnutí v běhu | |
| **kmínek** | kmen | kmín | | |
| **kohoutek** | zvíře | na umyvadle | | |
| **kolej** | dopravní | vysokoškolská | | |
| **koleno_** | ohbí nohy | spojovací součást | generace | |
| **kopačky** | obuv | rozchod | | |
| **korpus** | hudební | souhrn textu | | |
| **koruna** | královská | měna | zubní | stromu |
| **kostička** | kost | krychle | | |
| **kousek** | část | vzdálenost | | |
| **kozák** | ruský voják | hřib | | |
| **kraj** | okraj | oblast | | |
| **krajíček** | zdrob. od kraj | zdrob. od krajíc | | |
| **králíček** | malé zvíře | zpěvný pták | zdrob. od krále | |
| **králík** | malé zvíře | zdrob. od krále | | |
| **kus** | část oddělená od celku | samostatný předmět jistého druhu | divadelní hra | skutek |
| **lebeda** | bylina | pohodlí | | |
| **leč** | zájmeno | hon | | |
| **lekat** | budit strach | hynout | | |
| **lenoch** | líný člověk | opěradlo u židle | linkovaná podložka | |
| **lež** | podst. jméno | sloveso | | |
| **let** | podst. jméno | sloveso | | |

| | | | | | |
|---|---|---|---|---|---|
| **líc** | vrchní strana | líce | | | |
| **líčení** | básnický popis | soudní jednání | malování | předstírání | na zvířata |
| **list** | papíru | na stromě | noviny | | |
| **lištička** | malá liška | malá lišta | | | |
| **lízat** | olizovat | hladit | brát | | |
| **lokál** | hospodská místnost | pádová otázka | toto místo (příslovce) | | |
| **los** | zvíře | náhodná hodnota | výherní poukázka | název | |
| **loupit** | krást | loupat | | | |
| **mandle** | plod | v krku | | | |
| **masový** | z masa | hromadný | | | |
| **matka** | žena | k šroubku | | | |
| **místo** | pozice | náhrada | | | |
| **moc** | mnoho | síla | | | |
| **močit** | vylučovat moč | namáčet | | | |
| **mol** | jednotka koncetrace | motýl | přístavní molo | | |
| **mořit** | trápit | namáčet | moře | | |
| **muk** | citoslovce | druh jeřábu | | | |
| **myška** | hlodavec | část zbroje | zdrob. jméno | | |
| **naběračka** | nástroj na nabírání | nabírající žena | | | |
| **náčelník** | vedoucí | věc na čele | | | |
| **nadýmat** | vytvořit dým | nafouknout | | | |
| **najít** | vniknout | nalézt | | | |
| **nakupovat** | shlukovat | kupovat | | | |
| **napojit** | přiletovat | dodávat tekutiny | | | |
| **náprava** | část vozu | oprava, polepšení | | | |
| **naříkat** | namluvit | bědovat | | | |
| **navíječka** | nástroj k navíjení | navíjecí žena | | | |
| **novela** | literární útvar | úprava normy | | | |
| **noviny** | nové zprávy | denní tisk | | | |
| **objetí** | sevření | obkroužit jízdou | | | |
| **obloha** | vzdušný prostor | obložení | | | |
| **odpustit** | prominout | vypouštěním ubrat | | | |
| **ohryzek** | zbytek po ovoci | v krku | | | |
| **oko** | lidské | mastné | na oblečení | pytlácké | |
| **okno** | díra ve zdi | výpadek paměti | | | |
| **opera** | hudební dílo | budova | | | |
| **operátor** | pracovník | výraz | | | |
| **osádka** | skupina osob | malá osada | | | |
| **otava** | druhá tráva | zotavení | řeka | | |

| | | | |
|---|---|---|---|
| **ožít** | stát se živým | osekat | |
| **packat** | kazit | klopýtat | |
| **páčit** | násilím oddělovat | vynucovat | líbit se |
| **palác** | sídlo | velká síň | |
| **paleta** | malířská deska | deska na zboží | |
| **papoušek** | pták | mechanický opakovač | |
| **pár** | dvojice | neurčené malé množství | |
| **parabola** | kuželosečka | přirovnání | |
| **pářit** | vypouštět páru | dávat do dvojic | |
| **pas** | úřední doklad | přihrávka | místo trupu nad boky |
| **pás** | pruh | místo trupu nad boky | |
| **pasovat** | povyšovat | slušet | zasazovat |
| **pero** | na psaní | brk | |
| **piják** | savý papír | opilec | |
| **pila** | nářadí | sloveso | |
| **plít** | odstraňovat plevel | plivat | |
| **plotna** | deska | svrchní část kamen | |
| **plynný** | mající ráz plynu | souvisle probíhající | |
| **pojit** | spojit | napájet | |
| **pokoj** | místnost | klid | |
| **polka** | tanec | obyvatelka Polska | |
| **pomalu** | malou rychlostí | skoro | |
| **pop** | pravoslavný kněz | hudební styl | |
| **pór** | rostlina | drobný otvor | |
| **pošta** | podnik dopravující zásilky | souhrn písemností | |
| **potah** | tažné zvíře | povlak | |
| **potěr** | právě vylíhlé mláďata | vrstva na pevném podkladu | |
| **pračka** | stroj na praní | pranice | |
| **prášek** | sypká hmota | medikament | |
| **prodej** | směna | sloveso | |
| **prostředník** | prst | zprostředkovatel | |
| **přede** | předložka | sloveso | |
| **předek** | přední část | předchůdce | |
| **přezdít** | znovu zdít | pojmenovat | |

| | | | | |
|---|---|---|---|---|
| **připadat** | padáním nahromadit | dostávat | jevit se | |
| **přístavek** | malý přístav | co je dodatečně přistavěno | volný přívlastek | |
| **puk** | rozpuk | puklina | záhyb na části obleku | kotouč |
| **raketa** | sportovní náčiní | vesmírná loď | zbraň | |
| **rameno** | jeřábu | zvířecí či lidské | | |
| **rázem** | najednou | tlakem | | |
| **recese** | zpomalení | žert | | |
| **remíza** | plichta | kryté stanoviště vozidel | | |
| **restaurace** | hostinec | oprava | | |
| **rozchod** | vzdálenost | odloučení | | |
| **rub** | spodní strana | sekání | | |
| **rum** | odpadky a úlomky | lihovina | | |
| **rys** | zvíře | výkres | charakteristika | |
| **říz** | pohyb a zvuk při říznutí | prudký pád | ráznost | příchuť |
| **řízek** | plátek | část rostliny schopná zakořenit | pokrm z masa | |
| **seč** | prudký boj | sečba | mýtina | |
| **servis** | příbor | podání | služba | |
| **sestra** | sourozenec | v nemocnici | | |
| **sláva** | slavnost | vrcholné uznání | | |
| **spoušť** | zpustošení | zařízení uvádějící v činnost | | |
| **stát** | sloveso | území vymezené hranicemi | | |
| **stavět** | přerušit pohyb | budovat | | |
| **stolice** | na sezení | exkrement | | |
| **stolička** | na sezení | zub | | |
| **stopař** | kdo vyhledává stopy | kdo cestuje autostopem | | |
| **stopovat** | vyhledávat stopy | zastavovat auta | měřit čas | |
| **strop** | místnosti | horní hranice | | |
| **suk** | ve dřevě | uzel | jméno | |
| **šlupka** | slupka | velká rána | | |
| **šum** | směsice nejasných zvuků | rozruch | pěna | |
| **švih** | prudký pohyb | elegance | | |
| **tesák** | lovecký nůž | zub | | |
| **tisknout** | svírat | vtlačovat | přenášet na papír | |

| | | | | | |
|---|---|---|---|---|---|
| **topit** | udržovat teplotu | nořením do vody zabíjet | | | |
| **travička** | malá tráva | žena | | | |
| **třída** | společenská skupina lidí | skupina se společnými znaky | vyučovací stupeň | vyučovací místnost | ulice |
| **tvrzení** | výrok | zpevňování | | | |
| **ucho** | lidské | u nádobí či tašky | | | |
| **vál** | na těsto | vítr | | | |
| **vaz** | zadní část krku | vláknité pojivo mezi kostmi a svaly | druh jilmu | | |
| **vlečka** | železniční | na závoji | dopravní prostředek | | |
| **vinice** | vinohrad | vinná žena | | | |
| **viola** | strunný nástroj | fialka | | | |
| **vlna** | hustá jemná srst | vzedmutá kapalina | | | |
| **vrstevnice** | stejně stará žena | pomyslná čára | | | |
| **výklad** | výloha | vysvětlení obsahu | vyložení nákladu | | |
| **výtah** | zdviž | zhuštěný obsah | | | |
| **zabavit** | konfiskovat | najít zábavu | | | |
| **zájem** | prospěch | soustředěná pozornost | záliba | | |
| **zámek** | stavba | na dveřích | | | |
| **zápolí** | soupeří | místo za polem | | | |
| **zástava** | překážka | záruka | prapor | | |
| **závod** | podnik | měření sil | | | |
| **zdar** | kladný výsledek | pozdrav | | | |
| **zebra** | zvíře | přechod pro chodce | | | |
| **zima** | roční období | nízká teplota | | | |
| **znak** | charakteristická vlastnost | značka | záda | plavecký způsob | |
| **zub** | v ústech | nerovnost | | | |
| **žena** | osoba | přechodník hnát | | | |
| **ženu** | podst. jméno | sloveso | | | |
| **žínka** | zdrob. k žena | kus hrubší tkaniny k mytí | | | |