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

STROJOVÉ UČENÍ V PŘIROZENÉM JAZYCE

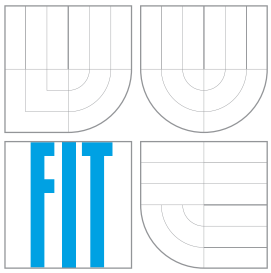
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUBOMÍR OTRUSINA

BRNO 2007

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

STROJOVÉ UČENÍ V PŘIROZENÉM JAZYCE

MACHINE-LEARNING IN NATURAL LANGUAGE PROCESSING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — LUBOMÍR OTRUSINA
AUTHOR

VEDOUCÍ PRÁCE — doc. RNDr. PAVEL SMRŽ, Ph.D.
SUPERVISOR

BRNO 2007

Zadání

Strojové učení v přirozeném jazyce

1. Seznamte se s pokročilými metodami strojového učení používanými v oblasti zpracování přirozeného jazyka.
2. Na základě získaných znalostí realizujte systém pro "učení" z textu.
3. Vytvořte testovací sadu pro vyhodnocení systému.
4. Vyhodnoťte vytvořený systém pomocí standardních metrik.

Kategorie: Umělá inteligence

Licenční smlouva

Licenční smlouva je uložena v archívu Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt

Tato práce se zabývá zjednoznačňováním slovních významů pomocí metod strojového učení. Čtenář je krátce seznámen s danou problematikou a jejím historickým vývojem. Jsou zde popsány nejpoužívanější metody a přístupy, speciálně pak naivní Bayesův klasifikátor, který je implementován v systému. Je zde uveden i názorný příklad pro tento klasifikátor. V praktické části je popsán návrh systému využívající tohoto klasifikátoru včetně popisu různých algoritmů použitých v systému. Na závěr je uvedeno vyhodnocení výsledků systému a jejich analýza. Implementovaný systém se zúčastnil soutěže v rámci mezinárodní konfernce sémantického vyhodnocování SemEval-2007.

Klíčová slova

strojové učení, učení s učitelem, zpracování přirozeného jazyka, zjednoznačňování slovních významů, naivní Bayesův klasifikátor, Senseval, Semeval

Abstract

This beachelor's thesis deals with word sense disambiguation problem using the machine learning techniques. There are shortly presented problems of word sense disambiguation and its timeline. There are described methods and approaches, especially the naive Bayes classifier that is implemented in the system. There's illustrated a simple example of using this classifier. In a practical section is described project of system based on naive Bayes classifier including description of various algorithms used in the system. Finally there are described evaluation and analysis of the system. This created system took part in an international competition on semantic evaluation workshop SemEval-2007.

Keywords

machine learning, supervised learning, natural language processing, word sense disambiguation, naive Bayes classifier, Senseval, Semeval

Citace

Lubomír Otrusina: Strojové učení v přirozeném jazyce, bakalářská práce, Brno, FIT VUT v Brně, 2007

Strojové učení v přirozeném jazyce

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D.

.....
Lubomír Otrusina
11. května 2007

Poděkování

Děkuji doc. RNDr. Pavlu Smržovi, Ph.D. za hodnotné rady a odborné vedení během mé práce.

Obsah

1 Úvod **3**

2 Zjednoznačňování slovních významů **4**
2.1 Vymezení základních pojmů . . . 4
2.2 Historický přehled metod zjednoznačňování slovních významů . . . 5
2.2.1 Metody založené na bázích znalostí . . . 6
2.2.2 Metody založené na korpusech . . . 6
2.2.3 Hybridní systémy . . . 7
2.3 WordNet . . . 7
2.3.1 Struktura WordNetu . . . 7
2.3.2 Značení synsetů ve WordNetu . . . 8

3 Strojové učení ve zjednoznačňování slovních významů **9**
3.1 Metody učení s učitelem . . . 9
3.2 Metody učení bez učitele . . . 9
3.3 Naivní Bayesův klasifikátor . . . 10
3.3.1 Pravděpodobnostní model . . . 10
3.3.2 Praktické využití . . . 11
3.3.3 Prázdné nebo chybějící atributy . . . 11
3.3.4 Příklad určování významu naivním Bayesovým klasifikátorem . . . 11

4 Návrh systému pro zjednoznačňování slovních významů **15**
4.1 Používané datové sady . . . 15
4.1.1 Senseval-3 . . . 15
4.1.2 Semeval-1 . . . 16
4.2 Využití WordNetu v systému . . . 16
4.3 Jednoduchý algoritmus učení s učitelem . . . 16
4.4 Princip práce s naivním Bayesovým klasifikátorem . . . 17
4.4.1 Načtení datových vstupů . . . 17
4.4.2 Analýza trénovacích dat . . . 17
4.4.3 Určení správného významu slova . . . 18
4.5 Od slov k vektoru příznaků . . . 18
4.6 Nastavování parametrů systému . . . 19

5 Vyhodnocení a výsledky systému **23**
5.1 Způsoby vyhodnocování zjednoznačňování slovních významů . . . 23
5.1.1 Vyhodnocování založené na počtu správně určených významů . . . 23
5.1.2 Vyhodnocování založené na křížové entropii . . . 24

5.1.3 Granularita rozlišování významů 25
5.1.4 Konference Senseval (Semeval) 26
5.2 Diskuze výsledků 26
5.2.1 Výsledky systému 26
5.2.2 Srovnání obou implementovaných algoritmů 27
5.2.3 Příklad špatného určení významu systémem 28
5.2.4 Návrh odstranění některých nedostatků 30
5.3 Technické parametry systému 31

6 Závěr **32**

A Přílohy **34**

Kapitola 1

Úvod

Přirozený jazyk je jazyk používaný lidmi k běžné komunikaci a sdělování informací. Narozdíl od počítačových programovacích jazyků se liší svou vágností. Ve své bakalářské práci se zabývám zjednoznačňováním slovních významů, což je jedním z mnoha problémů, kterými se zabývá obor zpracování přirozeného jazyka. Tento obor zasahuje jednak do oblasti umělé inteligence, ale také do lingvistiky. Jeho úkolem je určitá konverze mezi vágním přirozeným jazykem a nějakou formální reprezentací, které dokáže porozumět počítač.

Zpracováním přirozeného jazyka se odborníci zabývají již několik desítek let. Mezi hlavní problémy, které se snaží vyřešit, patří např. získávání a extrakce informací z textů, strojový překlad mezi různými jazyky, rozpoznávání mluvené řeči, korekce gramatických chyb nebo automatické odpovídání na otázky.

Pro úplné řešení všech těchto problémů je nezbytné, aby počítač porozuměl, nebo se alespoň pokusil analyzovat příslušný přirozený jazyk. Prostředkem pro porozumění přirozenému jazyku může být nějaká báze znalostí (např. ve formě sémantické sítě), která obsahuje vazby mezi slovy konkrétního přirozeného jazyka. Tyto báze znalostí bývají zpravidla vytvořeny ručně. Při analýze přirozeného jazyka se pak setkáváme s různými dílčími podproblémy, jako je vícevýznamovost slov nebo kontextové závislosti. Tyto podproblémy nám pomáhají řešit např. metody provádějící rozpoznávání slovních druhů (part-of-speech tagging) nebo zjednoznačňování slovních významů (word sense disambiguation).

Při řešení problémů z kategorie zpracování přirozeného jazyka se často využívá algoritmů strojového učení. Podle konkrétních případů pak může jít o algoritmy učení s učitelem, učení bez učitele nebo různé kombinace. Já jsem pro svou práci zvolil naivní Bayesův klasifikátor, což je algoritmus učení s učitelem. Tento klasifikátor je relativně jednoduchý a má velkou úspěšnost v mnohých odvětvích zpracování přirozeného jazyka.

Druhá kapitola (2) zahrnuje úvod do problematiky zjednoznačňování slovních významů, stručný popis vývoje a používané metody a přístupy. Ve třetí kapitole (3) jsou stručně popsány typy metod strojového učení, které jsou používány pro řešení tohoto problému. Zbytek kapitoly je pak věnován popisu naivního Bayesova klasifikátoru. Čtvrtá kapitola (4) se věnuje praktické části bakalářské práce. Je zde vysvětlena implementace a problémy vyskytující se při návrhu programu. Předposlední kapitola (5) obsahuje prezentaci a zhodnocení výsledků programu. Jsou zde také uvedeny způsoby vyhodnocování. V závěru (6) jsou pak stručně shrnuty dosažené výsledky a je zde uveden nástin dalšího vývoje v oblasti zjednoznačňování slovních významů.

Kapitola 2

Zjednoznačňování slovních významů

Zjednoznačňování slovních významů se snaží určit správný význam vícevýznamových slov v určitém kontextu. Tento problém řeší rodilý mluvčí často intuitivně a obyčejně mu nečiní žádné problémy. Jiná situace však nastává u počítače. Počítač přirozenému jazyku nerozumí, a proto je pro něj obtížné určit správný význam slova. Je nutné najít vhodné metody a vlastnosti, podle kterých by byl počítač s určitou úspěšností schopen správný význam identifikovat.

2.1 Vymezení základních pojmů

Pro uvedení do problému je nutné seznámit se nejdříve s některými základními pojmy. Jedná se zejména o definování pojmů týkajících se významu slova. Následující informace pocházejí z [4].

Pojem "význam slova nebo sousloví" (dále jen slovo) můžeme definovat jako informační obsah slova. Každé slovo může mít dva významy:

- slovní (lexikální) – označuje určitý jev skutečnosti; je to význam, který má slovo samo o sobě
- mluvnický (gramatický) – tento význam slovo nabývá až ve spojení s jinými slovy ve větě; vyjadřuje různé mluvnické kategorie (např. rod, číslo, pád ap.)

Ve své bakalářské práci se zabývám pouze určováním lexikálního významu slova. Mluvnickým významem se zde zabývat nebudu. Pokud se pokoušíme definovat lexikální význam (dále jen význam) nějakého slova, děláme to většinou použitím určitého jazyka. Význam slova může být definován pomocí

- přirozeného jazyka (včetně jazyka, v němž je slovo, jehož význam definujeme)
- formálního jazyka (např. vhodný matematický nebo logický kalkul)

Z toho se vymyká tzv. ostenzívní způsob definování významů slov přirozeného jazyka (např. toto je stůl, toto je zvíře,...). Na tomto způsobu je založeno učení se jazyku u člověka. Pokud má některé slovo více než jeden význam, pak mluvíme o vícevýznamovém slovu. Významy vícevýznamových slov můžeme rozdělit na:

- základní (primární, původní) např. oko – orgán zraku
- druhotný (sekundární, odvozeý) např. oko na punčoše

Pro každý přirozený jazyk je procento vícevýznamových slov odlišné. S vícevýznamovostí slov úzce souvisí pojmy polysémie a homonymie. I přesto, že se jedná o dva odlišné pojmy, jsou často zaměňovány. Polysémie neboli mnohoznačnost je existence více významů pro jednu formu, kde významy slova mají jistou genetickou podobnost (např. jazyk – orgán nebo jazyk u boty). Na druhé straně homonymie je existence více významů pro jednu formu, ale mezi významy není žádná spojitost – hlásková shoda je čistě náhodná (např. role – divadelní hra, pole, svitek). Homonymii můžeme dále dělit na:

- lexikální (např. vinný – od slova víno nebo vina)
- morfologická (např. bratra – vyjadřuje 2. i 4. p. sg.)
- slovně druhová (např. večer – podst. jm., příslovce)

Morfologickou a slovně druhovou homonymií se ve své bakalářské práci nebudu zabývat.

Proces výběru správného významu slova v daném kontextu se nazývá zjednoznačňování slovních významů.

2.2 Historický přehled metod zjednoznačňování slovních významů

Nejen ve zjednoznačňování slovních významů, ale v celé oblasti zpracování přirozeného jazyka existují dva směry řešení problémů. Oba jsou založeny na jiných principech a využívají odlišné techniky.

První přístup se snaží co nejvíce porozumět zpracovávanému textu. K tomu se nejčastěji využívá různých, většinou ručně vytvořených, bází znalostí. Mezi nejrozšířenější báze patří Machine Readable format (Oxford English Dictionary, Collins), thesaury (Rogetův thesaurus) nebo sémantické lexikony (WordNet, EuroWordNet). Z hlediska zjednoznačňování významů tyto báze typicky obsahují informace jako výčet významů, jejich definice, příklady použití, synonyma a různé vazby mezi nimi. Tyto metody se souhrnně označují jako metody založené na bázích znalostí.

Druhý přístup se nesnaží porozumět textu, pouze využívá obrovské množství dat, na základě jejichž analýzy hledá různé podobnosti, ze kterých pak vytváří pravidla pro zjednoznačňování. Zde se ve velké míře využívá různých algoritmů strojového učení. Může se jednat o algoritmy učení s učitelem, bez učitele nebo různé kombinace. Určování významů probíhá na základě pravidel, která systém automaticky generuje ze vstupních dat. Tato pravidla jsou tvořena během analýzy slov v kontextu. Mezi používané metody strojového učení patří např. shlukování a klasifikace. Tyto metody můžeme souhrnně nazvat jako metody založené na korpusech.

V této kapitole jsou popsány pouze některé významné kroky a směry ve zjednoznačňování slovních významů. Informace o historickém vývoji pocházejí z [1], kde může čtenář v případě zájmu najít vyčerpávající informace.

2.2.1 Metody založené na bázích znalostí

Koncem 50. let problém zjednoznačňování významů úzce souvisel se strojovým překladem. Několik vědců uvedlo, že pro úspěšný strojový překlad z jednoho jazyka do jiného, je určení správného významu slova nezbytné. Tento problém se týká slov, která mají po přeložení do cílového jazyka více možných alternativ. Například anglické slovo "bed" se může do francouzštiny přeložit jako "le lit" (postel) nebo "le parterre" (záhon).

Několik vědců zkoumalo vliv velikosti kontextu vícevýznamového slova na správné určení jeho významu. Pokusem bylo zjištěno, že úspěšnost překladu nebyla u kontextu tvořeným čtyřmi slovy nijak významně odlišná od případu, kdy byl poskytnut celý kontext. Této vlastnosti se využívá dodnes.

V 60. letech byl ve zjednoznačňování slovních významů poprvé použit Bayesův teorém. Skupina vědců se snažila odhadnout četnosti výskytů významů slov pro odlišná odvětví textů, na základě kterých pak určili pravděpodobnost každého významu pro určitý kontext. Tento přístup dosahoval tehdy úspěšnosti asi 90 %.

V 60. a 70. letech nastal velký růst výzkumů v oblasti umělé inteligence. Následkem toho je většina metod zjednoznačňování slovních významů z tohoto období založena na umělé inteligenci. Tyto metody se snažily detailně porozumět přirozenému jazyku a pokoušely se různým způsobem modelovat znalosti z lingvistické teorie. K tomu většinou využívaly ručně vytvořené sémantické sítě.

Ačkoliv byly některé metody založené na umělé inteligenci velmi zajímavé, jejich použitelnost byla značně omezena. Zejména proto, že používaly ručně vytvořené báze znalostí, které pokrývaly pouze specifickou oblast přirozeného jazyka.

Postupem času se začalo objevovat čím dál více použitelných bází znalostí. Takovými bázemi byly slovníky, thesaury nebo lexikony. Ačkoliv se zde objevila snaha o automatickou extrakci informací z dat, byly tyto rozsáhlé báze většinou vytvářeny ručně.

Při tvorbě lexikonů byl významný úspěch zaznamenán v roce 1990, kdy byl na Princetonské univerzitě vytvořen profesorem Millerem a jeho kolektivem lexikon anglického jazyka nazvaný WordNet. WordNet patří k tzv. výčtovým lexikonům, kde jsou uzly tvořeny známými významy pro slova. Opakem výčtového lexikonu je generativní lexikon, v němž jsou významové kategorie generovány na základě určitých pravidel. Vývoj WordNetu neustále probíhá, jeho současná verze je 3.0.

2.2.2 Metody založené na korpusech

V posledních letech se stávají stále více dostupné obrovské, většinou ručně anotované zdroje dat, které se dají využít pro empirické metody v mnohých odvětvích zpracování přirozeného jazyka. Tyto zdroje dat nazýváme korpusy.

Korpusem rozumíme vnitřně strukturovaný a ucelený soubor textů daného jazyka v elektronické podobě. Korpusy jsou organizovány se zřetelem na účel jejich použití, které může být všeobecného charakteru nebo specificky zaměřené na určitou oblast zpracování přirozeného jazyka. Mohou být složeny pouze z holého textu nebo mohou obsahovat metadata, která poskytují různé informace vzhledem k použití korpusu. Tato metadata mohou být do korpusu přidávána ručně nebo automaticky (strojově). Ruční anotování korpusů je vzhledem k jejich velikosti značně nákladné. Na druhou stranu automatické anotování může znamenat jisté zanesení chyby během anotování.

Prvním korpusem byl korpus anglického jazyka vytvořený na Brownově univerzitě v roce 1964. Tento korpus pomohl odhalit statistické charakteristiky slov (např. četnosti slov

a slovních druhů) v angličtině. V dnešní době má již mnoho států vytvořený svůj vlastní národní korpus, který je jistým způsobem reprezentativním vzorkem jazyka daného státu.

Hlavním problémem u metod zjednoznačňování slovních významů založených na korpusech je nerovnoměrnost dat. Tento problém se sice netýká pouze zjednoznačování, ale právě tady vyniká mnohem více než jinde. Potíž je v tom, že nikdo nemůže zaručit, aby se v korpusu vyskytovaly všechny významy vícevýznamových slov, a už vůbec nemůže být zaručeno, aby se vyskytovaly ve stejné míře. Například v Brownově korpusu, který obsahuje milion slov, se slovo `ash` vyskytuje pouze 8krát a z toho pouze jednou ve smyslu "jasan".

2.2.3 Hybridní systémy

Postupem častu se ukázalo, že lepších výsledků lze dosáhnout metodami využívajícími obou dvou přístupů. Tam, kde metody založené na bázech znalostí selhávají kvůli své nedokonalosti nebo nedostupnosti pro daný jazyk, můžeme tyto nedostatky vhodně doplnit metodami založenými na korpusech. V poslední době tak vznikají hybridní systémy využívající obou dvou přístupů.

Patří sem např. algoritmus bootstrapping. Algoritmus k funkci využívá malého množství anotovaných dat pro trénování, pocházejících z nějaké báze znalostí, a mnohem většího množství neanotovaných dat k testování. Algoritmus pracuje v cyklech, kdy v každém cyklu identifikuje vzory v trénovací množině, určí významy slov v testovacích datech a vybere několik nejvíce reprezentativních instancí, které pak přidá do trénovací množiny. Na takto nově vzniklé trénovací sadě se provede nové trénování algoritmu a celý proces se opakuje.

Jistou modifikací bootstrappingu je Yarowského algoritmus, který navíc obsahuje některé heuristické metody, mezi něž patří předpoklad neměnnosti významu slova pro všechny výskyty v jedné diskuzi, nebo předpoklad neměnnosti významu slova ve spojení s jiným slovem.

2.3 WordNet

WordNet [5] je sémantický lexikon anglického jazyka vyvíjený profesorem Georgem A. Millerem a kol. na Princetonské univerzitě od roku 1990. Je tvořen tzv. synsety, které reprezentují podstatná jména, přídavná jména, slovesa a příslovce. Tyto synsety jsou spojovány různými sémantickými vazbami. WordNet kombinuje výhody slovníků i thesaurů a snaží se tak vytvořit lexikon mnohem použitelnějším.

Od roku 2006 databáze WordNetu obsahuje asi 150 000 slov organizovaných do více než 115 000 synsetů. To všechno je komprimováno do necelých 12 MB dat.

2.3.1 Struktura WordNetu

Každý synset obsahuje množinu slov nebo slovních spojení mající stejný význam. Odlišné významy téhož slova jsou umístěny v odlišných synsetech. Význam každého synsetu je charakterizován krátkou glosou. Synsety jsou navzájem propojeny řadou sémantických vazeb. Příkladem některých vazeb v závislosti na typu synsetu může být:

- Podstatná jména
  - *hypernym* – Y je hypernym X, když každé X je (druh) Y (např. slovo "zvíře" je hypernym slova "pes")

  - *hyponym* – Y je hyponym X, když každé Y je (druh) X (např. slovo “stůl” je hyponym slova “nábytek”)
  - *holonym* – Y je holonym X, když X je část Y (např. slovo “strom” je holonym slova “větev”)
  - *meronym* – Y je meronym X, když Y je část X (např. slovo “klika” je meronym slova “dveře”)
- Slovesa
  - *hypernym* – sloveso Y je hypernym slovesa X, když aktivita X je (druhem) Y (např. sloveso “sportovat” je hypernym slovesa “běhat”)
  - *troponym* – sloveso Y je troponym slovesa X, když aktivita Y provádí nějakým způsobem X (např. sloveso “kráčet” je troponym slovesa “jít”)
  - *důsledek* – sloveso Y je důsledek X, když děláním X musí být uděláno Y (např. sloveso “spát” je důsledek slova “chrápat”)

Literály mohou být také spojovány různými lexikálními vztahy jako např. vztah označující antonymum.

2.3.2 Značení synsetů ve WordNetu

Je patrné, že z WordNetu lze získat mnoho informací. Mezi nejzákladnější možné informace však patří výpis významů určitého slova. Tyto významy jsou ve WordNetu přesně označeny a náleží jim specifický kód. Tento kód se používá pro označování významů i v datech, které používám. Kód má následujicí tvar zápisu:

`lemma%ss-type:lex-filenum:lex-id:head-word:head-id`,

kde `lemma` značí základní tvar slova, `ss-type` značí typ synsetu, `lex-filenum` reprezentuje lexikografický soubor obsahující synset, `lex-id` značí konkrétní význam v daném souboru. Zbylé části označení se používají ve zvláštních případech. Příkladem takového značení může být `bank%1:17:01::`.

Kapitola 3

Strojové učení ve zjednoznačňování slovních významů

Pro zjednoznačňování slovních významů můžeme s výhodou použít metod strojového učení. Metody strojového učení můžeme rozdělit do dvou základních skupin. První skupinu tvoří metody učení s učitelem, vyžadující jistá trénovací data, na kterých se "naučí" rozpoznávat významy. Druhou skupinou jsou metody učení bez učitele, které žádná taková data nemají, a musí provést přiřazení správného významu pouze na základě analýzy vstupních dat. Informace z této kapitoly pocházejí z [5] a [3].

3.1 Metody učení s učitelem

Tyto metody tvoří algoritmy využívající množinu trénovacích dat. Algoritmy identifikují vzory v příkladech daných pro jednotlivé třídy významu vícevýznamového slova. Tyto vzory jsou zobecněny do pravidel. Pravidla se pak následně použijí při klasifikaci nových případů.

Typicky se jedná o různé klasifikátory a klasifikační algoritmy. Těchto algoritmů je mnoho a většina z nich vykazuje na poli zjednoznačňování slovních významů dobré výsledky. Mezi nejpoužívanější metody učení s učitelem patří algoritmy podpůrných vektorů, rozhodovací stromy a seznamy, naivní Bayesův klasifikátor nebo neuronové sítě.

3.2 Metody učení bez učitele

Učení bez učitele zahrnuje metody, které se snaží ve velké množině dat identifikovat vzory. Nemají přitom k dispozici žádná anotovaná data, ani jiné externí znalosti. Vzory jsou používány pro rozdělení množiny dat do shluků. Tyto metody nemají žádné ponětí o skutečném rozdělení slov do významů. Je proto možné, že mohou najít odlišný počet shluků, než jaký je ve skutečnosti počet významů.

Zde se s výhodou používají různé druhy aglomerativního shlukování, které na začátku vytvoří tolik shluků, kolik instancí mají vstupní data. Následně je stanoven způsob měření podobnosti instancí a je nastavena prahová hranice. Pokud jsou dva shluky blíže než je minimální vzdálenost určená prahovou hranicí, dojde k jejich sloučení. Tento cyklus se opakuje do té doby, než obdržíme požadovaný počet shluků.

Jinou používanou metodou je zjednoznačňování slovních významů pomocí paralelních textů. Tato metoda nevyžaduje žádná anotovaná data, nicméně data v obou jazycích musí

být zarovnána. Taková data se dají běžně najít na internetu nebo ve speciálních korpusech (UN Parallel Text, Canadian Hansards). Pomocí zarovnaných textů jsou následně objevovány odlišnosti mezi významy.

3.3 Naivní Bayesův klasifikátor

Naivní Bayesův klasifikátor je jednoduchý klasifikátor založený na Bayesově teorému s předpokladem nezávislosti mezi příznaky. Tento klasifikátor můžeme s výhodou použít tam, kde jiné klasifikátory (rozhodovací stromy) selhávají kvůli nemožnosti použít velké množství příznaků. Naivní Bayesův klasifikátor může efektivně pracovat i s několika stovkami až tisíci příznaky. Tento klasifikátor je velmi úspěšný a velmi často se využívá k různým účelům. Mezi nejčastější použití patří klasifikace textů, zjednoznačňování slovních významů nebo filtrování spamu.

V praxi se klasifikátor používá tak, že se podle vzorce stanoví pravděpodobnosti pro všechny možné významy a z nich se vybere význam s největší pravděpodobností. Výhodou naivního Bayesova klasifikátoru je, že mu stačí pouze malé množství trénovacích dat pro odhadnutí potřebných pravděpodobností.

3.3.1 Pravděpodobnostní model

Pravděpodobnostní model pro klasifikátor je modelem podmíněným:

$$p(C \mid F_1, \ldots, F_n),$$

kde C je závislá třídní proměnná a $F_1, \ldots, F_n$ příznakové proměnné. Použitím Bayesova teorému můžeme tento vztah zapsat jako:

$$p(C \mid F_1, \ldots, F_n) \quad = \quad \frac{p(C) \cdot p(F_1, \ldots, F_n \mid C)}{p(F_1, \ldots, F_n)}$$

V praxi nás často nezajímá konkrétní hodnota pravděpodobnosti, ale její relativní poměr k ostatním pravděpodobnostem. Proto můžeme zlomek upravit odstraněním konstanty ve jmenovateli, čímž se výpočet zjednoduší. Upravený vzorec pak bude vypadat následovně:

$$p(C) \cdot p(F_1, \ldots, F_n \mid C)$$

Tento vztah můžeme dále upravit na:

$$p(C, F_1, \ldots, F_n)$$

Opakovaným aplikováním definice podmíněné pravděpodobnosti můžeme vzorec upravit na tvar:

$$\begin{aligned}
p(C, F_1, \ldots, F_n) &= p(C) \cdot p(F_1, \ldots, F_n \mid C) \\
p(C, F_1, \ldots, F_n) &= p(C) \cdot p(F_1 \mid C) \cdot p(F_2, \ldots, F_n \mid C, F_1) \\
p(C, F_1, \ldots, F_n) &= p(C) \cdot p(F_1 \mid C) \cdot p(F_2 \mid C, F_1) \cdot p(F_3, \ldots, F_n \mid C, F_1, F_2)
\end{aligned}$$

Nyní uplatníme předpoklad nezávislosti příznakových proměnných, který říká, že každý příznak F_i je podmíněně nezávislý na kterémkoliv jiném příznaku F_j pro $i \neq j$. Tuto podmínku můžeme zapsat následovně:

$$p(F_i \mid C, F_j) \quad = \quad p(F_i \mid C)$$

Proto můžeme předcházející vzorec přepsat na následující tvar:

$$\begin{aligned} p(C, F_1, \ldots, F_n) &= p(C)p(F_1 \mid C) \cdot p(F_2 \mid C) \cdot p(F_3 \mid C) \ldots \\ p(C, F_1, \ldots, F_n) &= p(C) \prod_{i=1}^{n} \cdot p(F_i \mid C) \end{aligned}$$

Celkově vzato můžeme pravděpodobnost toho, že při daných příznacích $F_1, F_2, \ldots, F_n$ bude jev patřit do třídy C, určit následujícím vzorcem:

$$p(C, F_1, \ldots, F_n) \quad = \quad \frac{1}{Z} \cdot P(C) \cdot \prod_{i=1}^{n} \cdot p(F_i \mid C),$$

kde Z vyjadřuje jistou konstantu, která je závislá pouze na nepodmíněných pravděpodobnostech. Ve skutečnosti můžeme Z vyjádřit jako:

$$Z \quad = \quad p(F_1, \ldots, F_n)$$

Výsledné rozhodování klasifikátoru pak můžeme zapsat vztahem:

$$classify(f_1, \ldots, f_n) \quad = \quad argmax_c P(C = c) \cdot \prod_{i=1}^{n} p(F_i = f_i \mid C = c)$$

3.3.2 Praktické využití

Jako příznaky se většinou berou výskyty klíčových slov v kontextu vícevýznamového slova. Všechny potřebné hodnoty, jako pravděpodobnosti výskytů slov, podmíněné pravděpodobnosti výskytů slov u významů nebo pravděpodobnosti výskytů významů, mohou být odhadnuty z množiny trénovacích dat.

3.3.3 Prázdné nebo chybějící atributy

Pokud se některé příznaky u jistého významu vůbec nevyskytují, je zřejmé, že pravděpodobnost jejich výskytu bude nulová. To může činit velký problém. Ve výsledném násobení všech pravděpodobností by vyšel celkový výsledek roven nule. Problém se řeší přidáním malé konstanty μ ke všem příznakovým hodnotám. Tento proces se nazývá Laplaceovo vyhlazování.

3.3.4 Příklad určování významu naivním Bayesovým klasifikátorem

Zde je zpracován jednoduchý příklad použití naivního Bayesova klasifikátoru pro zjednoznačňování slovních významů. Pravděpodobnosti výskytů slov a významů jsou odhadnuty z datové sady Senseval-3. Příklad je proveden pro anglický jazyk a demonstruje určování správného významu podstatného jména `disc`.

Množina trénovacích dat obsahuje 222 instancí. V příkladě budu počítat s hodnotami zaokrouhlenými na šest desetinných míst.

Slovo `disc` má podle sémantického lexikonu WordNet 2.1 následující významy (ve WordNetu jsou významy označeny pomocí speciálních kódů, pro lepší přehlednost jsem zvolil vhodná anglická označení):

- `disc%plate` A thin flat circular plate.
- `disc%music` Sound recording consisting of a disc with continuous grooves.
- `disc%computer` A memory device consisting of a flat disk covered with a magnetic coating on which information is stored.
- `disc%shape` Something with a round shape like a flat circular plate.

Mým programem byly pro tyto významy v trénovacích datech zjištěny následující četnosti.

Význam	**Četnost**	**Pravděpodobnost výskytu významu**
`disc%plate`	63	0,283784
`disc%music`	81	0,364865
`disc%computer`	45	0,202703
`disc%shape`	33	0,148649
celkem	222	1,000000

Tabulka 3.1: Četnosti a pravděpodobnosti výskytů významů.

Dále je potřeba vybrat vhodná klíčová slova, podle kterých bude klasifikátor rozhodovat. Princip výběru klíčových slov je popsán v kapitole 4.5. Těchto slov bývá mnoho, ale pro demonstraci našeho příkladu jich bohatě stačí jen několik. Spokojíme se tedy pouze s následujícími slovy:

`floppy, drive, computer, sand, musical, record, year, wolfcraft`.

Pro tato slova byly zjištěny následující podmíněné pravděpodobnosti. Tabulka 3.2 udává pravděpodobnosti výskytů klíčových slov pro všechny třídy významů.

Slovo	Třída významu			
	disc%plate	disc%music	disc%computer	disc%shape
floppy	0,000000	0,012346	0,266667	0,000000
drive	0,031746	0,000000	0,155556	0,000000
computer	0,000000	0,037037	0,422222	0,000000
musical	0,000000	0,074074	0,000000	0,000000
record	0,079365	0,617284	0,111111	0,030303
wolfcraft	0,111111	0,000000	0,000000	0,000000

Tabulka 3.2: Podmíněné pravděpodobnosti výskytů klíčových slov u daných významů.

V tabulce 3.2 je vidět, že některé hodnoty pravděpodobností jsou nulové. To je způsobeno tím, že v trénovacích datech nebyl pro daný význam nalezen žádný výskyt slova.

Nyní se můžeme pomocí naivního Bayesova klasifikátoru pokusit určit správný význam slova `disc` v následující větě:

Operating system decided to eject a floppy disc from the drive.

V této větě se nacházejí pouze některá klíčová slova a to `floppy` a `drive`. Proto budeme ve vzorci počítat pouze s těmito dvěma slovy.

Pravděpodobnost toho, že pro příznaky $F_1, \ldots F_n$ bude instance zařazena do třídy C, je podle zjednodušeného vzorce dána:

$$p(C \mid F_1, \ldots F_n) \quad = \quad p(C) \cdot \prod_i p(F_i \mid C)$$

Uvážíme-li pak pouze vybraná dvě slova, bude vzorec vypadat následovně:

$$p(C \mid floppy, drive) \quad = \quad p(C) \cdot p(floppy \mid C) \cdot p(drive \mid C)$$

Protože počítáme s příliš malými hodnotami a jejich násobením dostaneme hodnoty ještě menší, je vhodné celý vzorec zlogaritmovat na tvar:

$$ln(p(C \mid floppy, drive)) \quad = \quad ln(p(C)) + ln(p(floppy \mid C)) + ln(p(drive \mid C))$$

Při dosazování do vzorce nesmíme zapomenout ošetřit nulové hodnoty pravděpodobností přičtením vhodné konstanty. Tuto konstantu jsem zvolil 0,001.

$$
\begin{aligned}
ln(p(disc\%plate \mid floppy, drive)) &= ln(p(disc\%plate)) + \\
&+ ln(p(floppy \mid disc\%plate)) + \\
&+ ln(p(drive \mid disc\%plate)) = \\
&= (-1,259542) + \\
&+ (-6,907755) + \\
&+ (-3,449989) = \\
&= -11,617286
\end{aligned}
$$

$$
\begin{aligned}
ln(p(disc\%music \mid floppy, drive)) &= ln(p(disc\%music)) + \\
&+ ln(p(floppy \mid disc\%music)) + \\
&+ ln(p(drive \mid disc\%music)) = \\
&= (-1,008228) + \\
&+ (-4,394423) + \\
&+ (-6,907755) = \\
&= -12,310466
\end{aligned}
$$

$$
\begin{aligned}
ln(p(disc\%computer \mid floppy, drive)) &= ln(p(disc\%computer)) + \\
&+ ln(p(floppy \mid disc\%computer)) + \\
&+ ln(p(drive \mid disc\%computer)) = \\
&= (-1,596013) + \\
&+ (-1,321755) + \\
&+ (-1,860749) = \\
&= -4,778517
\end{aligned}
$$

$$
\begin{aligned}
ln(p(disc\%shape \mid floppy, drive)) &= ln(p(disc\%shape)) + \\
&+ ln(p(floppy \mid disc\%shape)) + \\
&+ ln(p(drive \mid disc\%shape)) = \\
&= (-1,906167) + \\
&+ (-6,907755) + \\
&+ (-6,907755) = \\
&= -15,721677
\end{aligned}
$$

Správný význam slova `disc` je reprezentován nejvyšší z těchto hodnot, což je $-4,778517$, která přísluší významu `disc%computer`.

Kapitola 4

Návrh systému pro zjednoznačňování slovních významů

Z kapitoly 3 je zřejmé, jak se naivní Bayesův klasifikátor používá k řešení problému zjednoznačňování slovních významů. V této kapitole se pokusím vysvětlit postup použití tohoto klasifikátoru v mém systému `wsd` s důrazem na jisté odlišnosti proti standardním postupům. Je zde popsán celý proces od analýzy dat přes výběr klíčových slov, ladění programu až po konečné určení významů. Nejprve však čtenáře seznámím s vlastní strukturou používaných dat. Stručně zde také popíšu další metodu učení s učitelem, která je rovněž implementována do systému.

4.1 Používané datové sady

Při řešení problému je nutné si nejprve obstarat vhodná data, na kterých se bude vlastní zjednoznačňování provádět. Takových dat se dá sehnat relativně mnoho, a to v různých formátech. Jistý problém však nastává ve chvíli, kdy si pro práci zvolíme konkrétní přirozený jazyk. Většina dat je totiž v anglickém jazyce. Pro češtinu je velmi obtížné sehnat vhodnou datovou sadu. Proto jsem se rozhodl použít anglický jazyk. Ze všech možných datových sad jsem vybral ty, které byly vytvořeny v rámci konferencí Senseval-3 a Semeval-1. Vzhledem k dlouholeté tradici těchto konferencí lze předpokládat dobrou úroveň těchto datových sad. Obě sady jsou tvořeny pro řešení problému zjednoznačňování pouze vybraných vícevýznamových slov a obsahují tyto slovní druhy: podstatná jména, přídavná jména a slovesa. V obou datových sadách jsou rozlišena data pro trénování a testování. Data jsou ve formátu XML, jejich zpracování je tedy velmi snadné.

4.1.1 Senseval-3

Data z konference Senseval-3 pocházejí z roku 2004 a byla pořízena přes systém Open Mind Word Expert.[1] Pro zvýšení spolehlivosti dat byla každá instance kontrolována alespoň dvěma nezávislými zdroji. Tato datová sada obsahuje přibližně 60 podstatných a přídavných jmen a sloves. Podstatná a přídavná jména jsou anotována pomocí významů pocházejících

[1] Open Mind Word Expert je systém využívající lidské schopnosti zjednoznačňování slovních významů a dává tak počítačům výhody lidských znalostí. Více informací lze získat na [6].

z WordNetu 1.7.1. Slovesa jsou anotována pomocí definic významů z thesauru Wordsmyth. Všechna data pocházejí z Britského národního korpusu. Z datové sady jsou vyjmuty instance obsahující slovní spojení s rozpoznávanými slovy. S daty je rovněž dodávána mapa významů, která zmenšuje množinu podobných významů na jeden. Této mapy se využívá při vyhodnocování úspěšnosti. Díky této mapě můžeme různě zohledňovat případy, kdy systém určí význam, který sice není úplně správný, ale patří do podobné významové skupiny jako správný význam slova.

4.1.2 Semeval-1

Data pocházející z konference Semeval-1 jsou velmi podobná datům ze Sensevalu-3. Opět jsou zde obsaženy tři základní slovní druhy a to podstatná a přídavná jména a slovesa. Data tentokrát pocházejí z tisku Wall Street Journal a Brownova korpusu. Zásadní změna oproti datům Senseval-3 je ta, že tvůrci se tentokrát rozhodli neponechat možnost způsobu určování významů založeného na jemné granularitě. Místo klasicky definovaných slovních významů zde nalezneme pouze jakési významové množiny, které vznikly sloučením několika skutečných významů. Každému vícevýznamovému slovu pak přísluší několik významových skupin, které jsou tvořeny významy z WordNetu.

4.2 Využití WordNetu v systému

Ve vstupní množině dat se vyskytuje značné množství slov. Tato slova ovšem nemusí být ve svém základním tvaru (lemma). Proto program pracuje např. s anglickými slovy `has` a `have` jako se zcela odlišnými slovy. To by mohlo negativně ovlivnit vlastní proces zjednoznačňování, protože by mohlo dojít k výběru jiných klíčových slov. Pokud bychom např. vyšetřovali slovo `computer` jako jednoho z možných kandidátů na klíčové slovo, mohlo by dojít k situaci, kdy by pravděpodobnost výběru tohoto slova byla oslabena výskyty slova `computers`, což jistě nechceme.

Proto jsem se rozhodl před vlastní analýzou dat nejprve všechna slova převést na jejich základní tvar. Lemma je pro každé slovo získáváno z WordNetu. Pro jeho získání je ovšem potřeba znát slovní druh slova. Některá slova totiž mohou nabývat podle situace více slovních druhů. Každý slovní druh pak může mít jiné lemmma. Například slovo `starting` má ve WordNetu pro podstatné jméno lemma `starting` a pro sloveso lemma `start`. Ve vstupních datech ovšem není uveden slovní druh pro každé slovo. Nemůžeme tedy jednoznačně určit jeho lemma. Bylo by možné použít speciálních nástrojů pro doplnění slovního druhu ke každému slovu. Tento způsob je však vzhledem k povaze problému značně složitý. Rozhodl jsem se zvolit jednodušší způsob. Z WordNetu se zjistí všechna lemmata pro dané slovo a vybere se to, které se vyskytuje nejčastěji. Ve WordNetu jsou uvedeny četnosti nejčastěji se vyskytujících významů slov. Je pravděpodobné, že chybně určená lemmata nebudou mít na funkčnost systému nijak velký vliv. Lemmata jsou generována mnou vytvořeným programem `lemmatizer`. [2]

4.3 Jednoduchý algoritmus učení s učitelem

Kromě naivního Bayesova klasifikátoru, který bude popsán dále, je v systému implementován jednoduchý algoritmus zjednoznačňování slovních významů založený na metodě učení

[2] Program lemmatizer se nachází na přiloženém CD.

s učitelem. Tento algoritmus je implementován z důvodu možnosti určitého srovnání s naivním Bayesovým klasifikátorem. Algoritmus je převzat z [3], kde je možné také najít bližší informace.

Algoritmus je založen na principu tvorby určitých skupin slov pro každý význam. Na začátku algoritmus vytvoří tolik skupin, kolik má vícevýznamové slovo významů. Následně prochází všechny instance trénovacích dat. Nejprve se pro každou instanci zjistí skutečný význam slova. Následně se vloží do skupiny slov s příslušným významem všechna slova vyskytující se v kontextu vícevýznamového slova. Tento postup se opakuje pro všechny instance trénovacích dat. Ve skupině slov je přitom každé slovo pouze jednou. Algoritmus pak dostane testovací data, u kterých má určit správný význam. Prochází postupně všechna slova v kontextu testovací instance. Každé skupině slov pak zvýší skóre o jedničku, pokud se právě zpracovávané slovo ve skupině nachází. Po zpracování všech slov v kontextu algoritmus vybere správný význam na základě nejvyššího dosaženého skóre. V pseudo-kódu můžeme algoritmus zapsat následujícím způsobem:

```
For each word Wi in C
        For each sense Si in S
        If Wi is in SENSE_i_BAG then
                Si = Si + 1;

correct_sense = max(Si);
```

kde C značí kontext, W slovo a S význam slova.

4.4 Princip práce s naivním Bayesovým klasifikátorem

Celý systém pro zjednoznačňování je implementován objektově. Běh algoritmu je řízen vhodným voláním metod různých objektů. Tento proces má určitý řád, který je potřeba dodržet.

4.4.1 Načtení datových vstupů

Po spuštění programu je vytvořen hlavní objekt `Chief`, který zahrnuje veškeré další objekty a datové struktury. Následuje načtení všech potřebných souborů s daty. Těchto souborů je několik a mohou být ve formátu Semeval-1 nebo Senseval-3. Prvním ze souborů je trénovací datová sada. Tento soubor je tvořen stovkami až tisíci instancemi trénovacích dat. Každá instance obsahuje vícevýznamové slovo s celým jeho kontextem, včetně určeného správného významu. Podobný formát má testovací datová sada. Ta se liší pouze v tom, že neobsahuje správně doplněné významy. Dalším souborem je soubor s lemmaty všech slov, která se vyskytují v trénovacích nebo testovacích datech. Tento soubor je možno vygenerovat pomocí programu `lemmatizer`. Posledním souborem je soubor se správnými odpověďmi pro testovací data. Soubor s lemmaty ani soubor se správnými odpověďmi není nezbytný ke správné funkčnosti programu.

4.4.2 Analýza trénovacích dat

Po načtení všech souborů program zahájí analýzu trénovacích dat. Nejprve je v hlavním objektu `Chief` vytvořena kolekce objektů `Lexelt`, která reprezentuje jednotlivá rozpoznávaná lemmata. Každý objekt `Lexelt` obsahuje mnoho dalších kolekcí. Mezi nejvýznamnější patří

kolekce objektů `SenseItem` reprezentující význam slova, kolekce objektů `LexeltItem` reprezentující konkrétní vícevýznamová slova, včetně jejich kontextu. Dalším objektem je `Answer`, který slouží pro práci s výsledky. Po vytvoření všech potřebných objektů je pro všechny objekty `Lexelt` zavolána metoda `createWordsTable()`. Tato metoda vypočítá četnosti všech slov pro nalezené významy. Následuje volání metody `createDeviationTable()`, která pro všechna slova každého objektu `Lexelt` vypočítá potřebné statistické údaje. Tyto údaje jsou následně poskytnuty metodě `sortSignificantWords()`, která podle těchto vlastností vybere mezi všemi slovy nejvhodnější kandidáty na klíčová slova. Metoda `selectSignificantWords()` následně pro každý objekt `LexeltItem`, patřící do množiny testovacích dat vybere N klíčových slov ze seznamu všech možných kandidátů. Pro tato klíčová slova jsou pak metodou `computeWordsProbability()` vypočítány potřebné pravděpodobnosti pro naivní Bayesův klasifikátor.

4.4.3 Určení správného významu slova

Jsou-li provedeny všechny potřebné analýzy trénovacích dat, přijde na řadu vlastní klasifikace. Klasifikaci provádí metoda `makeBayes()`, která vypočítá ohodnocení každého významu pro všechny instance testovacích dat. Ohodnocení je počítáno podle výše uvedeného vzorce z vektoru příznaků tvořeného slovy vybranými metodou `selectSignificantWords()`. Po výpočtu ohodnocení pro všechny instance je zavolána metoda `makeResult()`, která pro každou instanci vybere na základě nejlepšího ohodnocení předpokládaný správný význam. Pokud je programu dodán soubor se správnými výsledky, vypíše metoda `printHitRate()` úspěšnost pro všechna zpracovávaná lemmata. Vypsána je také průměrná, minimální a maximální úspěšnost. Řešení je uloženo do souboru metodou `saveResult()`, která umožňuje uložit výsledky jako prostý text nebo ve formátu XML.

4.5 Od slov k vektoru příznaků

Naivní Bayesův klasifikátor je pevně daný algoritmus, který dává pro určitý vstupní vektor příznaků vždy stejné výsledky. Přiřazení správného významu slova je tedy velmi závislé na vektoru příznaků. Špatná volba příznaků může mít zásadní dopad na funkčnost celého systému. Je tedy nutné věnovat tvorbě tohoto vektoru velkou pozornost.

V podobných systémech založených na naivním Bayesově klasifikátoru je většinou vektor příznaků tvořen několika tisíci položkami. Každá položka reprezentuje jedno klíčové slovo. Hodnota položky vektoru pak značí, jestli se klíčové slovo nachází v kontextu vícevýznamového slova či nikoliv. Je zřejmé, že počet příznaků reprezentujících slova, která se v kontextu vyskytují, nebude nijak velký. V případě tohoto způsobu použití se stanoví vektor příznaků pro každé určované lemma pouze jednou. Tento vektor je opakovaně používán na všechny instance testovacích dat pro dané lemma.

Já jsem zvolil poněkud odlišný způsob tvorby vektoru příznaků. Můj vektor příznaků obsahuje jen velmi málo položek (něco okolo 10-ti). Vektor není tvořen globálně pro všechny instance stejného lemmatu, ale je tvořen lokálně pro každou instanci individuálně. Slova jsou do vektoru příznaků vybírána tak, aby se v kontextu vícevýznamového slova vždy vyskytovala. Tento způsob zaručí to, že každá instance bude charakterizována přesně specifikovaným vektorem N položek. Nemůže tedy nastat případ, kdy kontext vícevýznamového slova nebude zrovna reprezentativní a dojde k situaci, kdy z několikatisícového vektoru bude v kontextu obsaženo pouze velmi málo slov nebo dokonce žádné. V případě, že by v kontextu nebylo obsaženo žádné klíčové slovo, klasifikátor by rozhodoval pouze na základě

pravděpodobností výskytů významů.

Zůstává tedy otázka, jak vybrat správně N klíčových slov pro každou instanci trénovacích dat. Tento výběr je řešen ve více krocích a tvoří klíčovou část programu. Obecně řečeno, program provede ohodnocení všech slov v trénovacích datech a pro každé vícevýznamové slovo pak vybere nejvhodnější slova z jeho kontextu na základě tohoto ohodnocení.

Nejprve jsou vypočítány podmíněné pravděpodobosti výskytů slov pro všechny uvažované významy slova. Následně je z těchto pravděpodobností určeno několik hodnot, podle kterých se počítá výsledné ohodnocení slova. První z hodnot `dev`, je směrodatná odchylka pravděpodobností výskytů vydělená jejich průměrnou hodnotou. Směrodatné odchylky pravděpodobností výskytů mohou nabývat pro různá slova značně odlišných hodnot. Pokud je vydělíme průměrnou hodnotou, dojde tak k určité normalizaci a hodnoty lze lépe porovnávat. Dalšími hodnotami jsou maximální pravděpodobnost výskytu slova pro příslušné významy `max` a jí odpovídající četnost `count`.

Z těchto hodnot se vypočítá ohodnocení všech slov obsažených v trénovacích datech. Ohodnocení se vypočítává podle následujícího vzorce:

$$\text{ohodnocení} \quad = \quad X \cdot \frac{dev}{Dev} + Y \cdot \frac{max}{Max},$$

kde dev, max jsou hodnoty uvedené výše a Dev, Max jsou jejich maximální hodnoty v rámci všech slov pro dané lemma. Dělení maximální hodnotou je prováděno opět kvůli normalizaci. X a Y jsou vhodně zvolené konstanty. Jejich nastavení se věnuje kapitola 4.6.

Teoreticky by ohodnocení mělo být počítano pro úplně všechna slova. Zvolil jsem ale jistá omezení, která vybírají pouze určitá vhodná slova, pro která bude počítáno ohodnocení. Toto omezení je založeno na hodnotách `dev` a `count`, kde jsem zvolil minimální hodnoty, při kterých bude pro dané slovo ohodnocení počítáno. Tyto hodnoty jsou opět vhodně nastaveny. Tímto zjednodušením jsem se vyhnul výpočtům ohodnocení u většiny slov. Vynechaná slova by měla ohodnocení velmi malé a jen stěží by byla vybrána jako klíčová.

4.6 Nastavování parametrů systému

Pro správný běh programu je třeba nejprve program vyladit na používanou datovou sadu. Ladění programů probíhá ve formě nastavování několika jeho parametrů. Jedná se zejména o délku použitého vektoru příznaků `N`, dále parametr μ vyjadřující konstantu potřebnou pro ošetření nulových hodnot některých pravděpodobností při rozhodování naivního Bayesova klasifikátoru, parametry `X` a `Y` potřebné pro výpočet ohodnocení slov a parametry určující zamítnutí slova, kdy nedojde k počítání ohodnocení. Tyto parametry jsem nazval `min_dev` a `min_count`. Nastavování vhodných hodnot je prováděno empiricky na základě úspěšnosti programu pro různé hodnoty parametrů. Hodnoty těchto parametrů můžou ovlivnit výkon i úspěšnost programu. Je proto vhodné jim věnovat patřičnou pozornost.

Při nastavování délky vektoru příznaků se ukázalo, že nejvhodnější délka je 8 položek. Při menší délce docházelo již k poklesu úspěšnosti, což je způsobeno tím, že počet slov charakterizující daný význam je nedostatečný. Při zvyšování délky vektoru úspěšnost nejdříve chvíli stagnovala, a pak začala rovněž klesat. Tento jev je způsoben tím, že se do vektoru příznaků začínají dostávat méně významná slova. Zprvu ještě tato slova nemají takový vliv na klasifikaci, ale s jejich vzrůstajícím počtem začínají negativně ovlivňovat výsledek.

Vhodné nastavení parametru μ je pro běh systému klíčové. Pokud by byl nastaven na příliš vysokou hodnotu, byly by klasifikátorem zvýhodňovány významy, pro něž mají některá slova nulové pravděpodobnosti výskytů. Pokud bychom tento parametr nastavili na příliš nízkou hodnotu, docházelo by k opačnému jevu a klasifikátor by znevýhodňoval významy, pro které mají některá slova nulovou pravděpodobnost výskytu. Tento parametr byl nastaven na hodnotu $0,001$.

Parametry `X` a `Y` již nemají takový vliv na úspěšnost programu. Během experimentování s jejich nastavením bylo zjištěno, že hodnota `dev` má na úspěšnost programu větší vliv než hodnota `max`. Parametr `X`, který vyjadřuje váhovou hodnotu proměnné `dev`, by měl být proto nastaven na vyšší hodnotu než parametr `Y` vyjadřující váhovou hodnotu proměnné `max`. Parametr `X` byl nastaven na hodnotu 20 a parametr `Y` na hodnotu 12.

Posledními parametry, které zbývá nastavit jsou hodnoty `min_dev` a `min_count`. Tyto hodnoty rozhodují o zamítnutí kandidátních slov. Při nastavování bylo zjištěno, že úspěšnost programu je na parametr `min_dev` nejméně citlivá. Je to způsobeno tím, že pokud není nastaven do extrémních hodnot, tak jen způsobuje výpočty ohodnocení u více kandidátních slov. Pokud program dokáže vybrat z určitého množství kandidátních slov N klíčových slov, tak v případě zvýšení množství kandidátních slov vybere program pravděpodobně stejná klíčová slova. Tento parametr může výrazněji ovlivňovat úspěšnost, jen pokud je nastavena délka vektoru příznaků na velkou hodnotu a počet kandidátních slov tak nestačí k jeho naplnění. Parametr `min_dev` byl nastaven na hodnotu $0,800$. Zbývající parametr `min_count` má na celkovou úspěšnost programu mnohem větší vliv než parametr `min_dev`. Hodnota `dev` totiž může za jistých okolností nabývat vysoké hodnoty, i když je dané slovo nevýznamné. Tento případ nastává u slov, která se vyskytují pouze u jednoho významu a to v malém počtu. Pokud se např. slovo `planet` při určování významu slova `bank` bude vyskytovat pouze jednou, a to jenom u jednoho významu, vyjde hodnota `dev` vysoká, což způsobí velké ohodnocení slova. Je zřejmé, že toto slovo nemá mít na klasifikaci žádný vliv. Parametr `min_count` byl nastaven na hodnotu $1,500$, což znamená, že jsou odfiltrována všechna slova, která se vyskytují v kontextu pouze jednou.

Nyní uvedu několik tabulek znázorňujících závislost úspěšnosti programu na nastavování různých parametrů.

Hodnota parametru `N`	**Úspěšnost**
1	78,4787 %
2	79,7361 %
3	81,0761 %
4	81,5914 %
5	81,7151 %
6	81,7357 %
7	81,8594 %
8	82,1892 %
9	81,9419 %
10	81,9419 %
15	81,7976 %
30	81,7770 %

Tabulka 4.1: Úspěšnost systému pro různé hodnoty parametru N, při konstantních zbylých parametrech. Měřeno na datové sadě Semeval-1.

μ	**Úspěšnost**
0,1	18,9402 %
0,01	62,8296 %
0,001	64,5030 %
0,0001	64,1734 %
0,00001	63,8945 %

Tabulka 4.2: Úspěšnost systému pro různé hodnoty parametru μ, při konstantních zbylých parametrech. Měřeno na datové sadě Senseval-3.

Poměr parametrů `X:Y`	**Úspěšnost**
1:2	63,4888 %
1:1	64,0467 %
2:1	64,2748 %
3:1	64,3763 %
5:1	64,3256 %
10:1	64,2495 %

Tabulka 4.3: Úspěšnost systému pro různé hodnoty poměru parametrů X:Y, při konstantních zbylých parametrech. Měřeno na datové sadě Senseval-3.

Hodnota parametru `min_dev`	**Úspěšnost**
0,500	64,1481 %
0,800	64,5030 %
0,900	64,1734 %
1,000	64,1227 %
1,200	64,1481 %
1,500	63,2860 %
2,000	59,8631 %

Tabulka 4.4: Úspěšnost systému pro různé hodnoty parametru min_dev, při konstantních zbylých parametrech. Měřeno na datové sadě Senseval-3.

Hodnota parametru `min_count`	**Úspěšnost**
0,5	59,6349 %
1,5	64,5030 %
2,5	63,3874 %
3,5	62,5761 %
4,5	60,7759 %

Tabulka 4.5: Úspěšnost systému pro různé hodnoty parametru min_count, při konstantních zbylých parametrech. Měřeno na datové sadě Senseval-3.

Ze všech uvedených tabulek je vidět, že vhodným nastavením parametrů lze docílit zvýšení úspěšnosti programu. Nejzásadnější vliv na úspěšnost programu má parametr μ ovlivňující vlastní klasifikaci. Ostatní parametry mají na systém již výrazně menší vliv.

Kapitola 5

Vyhodnocení a výsledky systému

V této kapitole budou uvedeny nejpoužívanější metody vyhodnocování systémů pro zjednoznačňování slovních významů. Bude provedeno vyhodnocení mého systému podle standardních metrik na konferencích Senseval-3 a Semeval-1. Je zde uveden příklad selhání naivního Bayesova klasifikátoru na konkrétní instanci datové sady Senseval-3. V příkladě se pokusím zjistit příčiny selhání tohoto klasifikátoru. Na závěr následuje krátký popis systému po technické stránce. Informace o metodách vyhodnocování jsou převzaty z [2].

5.1 Způsoby vyhodnocování zjednoznačňování slovních významů

Pro vyhodnocování systémů vykonávajících zjednoznačňování slovních významů se používá několik metod. Narozdíl od jiných problémů z oblasti zpracování přirozeného jazyka zde prozatím ještě nebyla stanovana žádná přesná pravidla, podle kterých by se mělo vyhodnocování řídit. Mnozí výzkumníci nebo skupiny pracující v tomto odvětví mají stanoveny vlastní způsoby vyhodnocování, které se od způsobů jiných skupin liší.

5.1.1 Vyhodnocování založené na počtu správně určených významů

Mezi nejjednodušší a zároveň nejpoužívanější způsoby vyhodnocování patří metoda vycházející z následujícího vzorce:

$$\text{úspěšnost} = \frac{\text{počet správně určených významů}}{\text{celkový počet určovaných slov}}$$

Tato metoda ovšem přináší do vyhodnocování jisté nevýhody. Uvažujme např. následující text:

... bought an interest in Lydak Corp. ...

Představme si nyní následující přiřazení pravděpodobností všem významům pomocí čtyř hypotetických systémů, které zobrazuje tabulka 5.1.

Význam	Systém			
	1	**2**	**3**	**4**
(1) monetary	0,47	0,85	0,28	1,00
(2) stake of share	0,42	0,05	0,24	0,00
(3) benefit/advantage/sake	0,06	0,05	0,24	0,00
(4) intellectual curiosity	0,05	0,05	0,24	0,00

Tabulka 5.1: Úspěšnosti hypotetických systémů.

Je patrné, že ani jeden ze zvažovaných hypotetických systémů neurčil význam slova v uvedené větě správně. Správný význam ve větě je význam s číslem 2. Vidíme, že všechny systémy zvolily význam č. 1. Podle výše uvedeného kritéria vyhodnocování tedy mají v uvedeném případě všechny systémy úspěšnost 0 %. Podíváme-li se blíže na pravděpodobnosti přiřazené jednotlivým významům systémy, vidíme, že systém č. 1 přiřadil správnému významu největší pravděpodobnost ze všech systémů. Od určení správného významu slova ho navíc dělil pouze malý krok. Ostatní systémy vždy silně preferovaly pouze jeden význam, nebo určily velmi podobné pravděpodobnosti pro všechny možné významy. Je velmi vhodné, aby systém určitým způsobem vyjadřoval důvěru, kterou jednotlivým významům přikládá. Tato vlastnost je důležitá, pokud bychom chtěli výstupy těchto systémů použít jako vstupy nějakého jiného systému pracujícího s pravděpodobnostmi. V takovém případě by systém č. 1 vyhovoval nejvíce.

5.1.2 Vyhodnocování založené na křížové entropii

Tento problém řeší způsob vyhodnocování založený na křížové entropii. Úspěšnost systému je zde počítána podle následujícího vzorce:

$$-\frac{1}{N}\sum_{i=1}^{N} log_2 P_s(cs_i \mid w_i, context_i),$$

kde N je počet testovacích instancí, P_s je pravděpodobnost přiřazená správnému významu cs_i vícevýznamového slova w_i v kontextu $context_i$ systémem s. Tímto způsobem bude systému č. 1 přiřazeno větší ohodnocení, protože správnému významu přiřadil velkou pravděpodobnost na rozdíl od ostatních systémů. Následuje tabulka 5.2 s vyhodnocením úspěšností pomocí tohoto kritéria.

Metoda křížové entropie	Systém			
	1	**2**	**3**	**4**
ohodnocení	1,25	4,32	2,05	∞

Tabulka 5.2: Ohodnocení hypotetických systémů pomocí křížové entropie.

Pro interpretaci ohodnocení je potřeba si uvědomit, že nejlepší ohodnocení zde představuje nejmenší z hodnot. Vidíme, že se zde ukázal podle očekávání nejúspěšnější systém č. 1. Systém č. 4, který dal správnému významu pravděpodobnost rovnu nule, má při tomto způsobu vyhodnocování nejmenší ohodnocení.

Je třeba si uvědomit, že mnoho systémů není schopno určit pravděpodobnost pro každý význam. V takovém případě je nutné použít nějakých alternativních způsobů, které jsou uvedeny například v [2].

5.1.3 Granularita rozlišování významů

Dalším problémem vyskytujícím se při vyhodnocování systémů je existence různých sémantických nebo komunikativních vazeb mezi jednotlivými významy. Je zřejmé, že různá vícevýznamová slova mají některé významy podobné. Je proto vhodné významy uspořádat do jisté hierarchické struktury. Ukázkou takové struktury může být např. tabulka 5.3 významů pro anglické slovo `bank`.

I	Bank	-REPOSITORY
	I.1	Financial Bank
		I.1a - the institution
		I.1b - the building
II	Bank	-GEOGRAPHICAL
	II.1	Shoreline
	II.2	Ridge/Embankment
III	Bank	-ARRAY/GROUP/ROW

Tabulka 5.3: Hierarchická struktura významů slova bank.

Je vhodné při vyhodnocování brát ohled na tento fakt a zavést určitý způsob penalizace reflektující toto hierarchické uspořádání. Pokud systém určí význam, který patří do stejné kategorie nebo podkategorie, měla by být penalizace menší, než pokud systém určí nesprávný význam patřící do úplně jiné kategorie. Penalizaci pak můžeme provádět např. na základě penalizační matice, kterou určíme pro každé vícevýznamové slovo. Matice pro slovo `bank` může vypadat následovně:

	I.1a	**I.1b**	**I.2**	**II.1**	**II.2**	**III**
I.1a	0	1	2	4	4	4
I.1b	1	0	2	4	4	4
I.2	2	2	0	4	4	4
II.1	4	4	4	0	1	4
II.2	4	4	4	1	0	4
III	4	4	4	4	4	0

Tabulka 5.4: Penalizační matice pro významy slova bank.

Rozlišování významů na úrovni jemné granularity nijak nezohledňuje hierarchickou strukturu všech významů. Pro správné určení významu je nutné určit i jeho konkrétní podskupinu. To vede k výraznému snížení úspěšnosti systému. Obzvláště pak pokud nejsou použity další metody zohledňující pravděpodobnosti, které systém přidělil jednotlivým významům. Na druhé straně stojí rozlišování významů na úrovni hrubé granularity snažící se určitým způsobem zvýhodnit podobnosti některých významů. Často se to provádí tak, že se ze skutečných existujících významových skupin vytvoří nové skupiny, které vznikají sloučením několika skupin původních a reflektují tak podobnosti některých významů. Přitom

není nikde uvedeno do kolika významových skupin se mají původní významy sloučit. Tento způsob není tak dokonalý jako např. u penalizační matice, ale je to určitý krok kupředu. S tímto způsobem vyhodnocování se můžeme setkat i při vyhodnocování výsledků soutěží v rámci konferencí Senseval-3 a Semeval-1. Různá úroveň granularity při rozlišování významů má velký vliv na vyhodnocování.

5.1.4 Konference Senseval (Semeval)

Senseval je mezinárodní konference pořádaná organizací ACL-SIGLEX zabývající se problémy zjednoznačňování slovních významů. Jejím účelem je zkoumání silných a slabých stránek systémů zabývajících se touto problematikou. Senseval-3 je otevřená konference a může se jí zúčastnit kdokoliv. Poskytuje jednotná data a způsoby jejich vyhodnocování. Letos se v Praze koná již 4. ročník této konference, která nese název Senseval-4/Semeval-1 a již není zaměřena pouze na zjednoznačňování slovních významů. Snaží se řešit i jiné problémy sémantického zpracování přirozeného jazyka. Mezi některé zajímavé problémy patří např. klasifikace sémantických rolí, víceúrovňová sémantická anotace katalánštiny a španělštiny, rozlišování metonymií nebo extrakce sémantických struktur. Do soutěže v rámci této konference jsem se zapojil i já se svým systémem při řešení problému zjednoznačňování vybraných vícevýznamových slov pro angličtinu.

5.2 Diskuze výsledků

V této podkapitole se pokusím provést zhodnocení systému pro použité datové sady. Jsou zde uvedeny úspěšnosti i pro jiné systémy na datové sadě Semeval-1.

5.2.1 Výsledky systému

Systém byl testován na dvou datový sadách pocházejících z konferencí Senseval-3 a Semeval-1.

Datová sada pro Senseval-3 poskytuje dodatečné informace umožňující vyhodnocování na různé úrovni granularity. Protože můj systém tyto informace nevyužívá, je prováděno pouze vyhodnocování pro jemnou granularitu. U tohoto způsobu bylo dosaženo nejlepší úspěšnosti 64,5030%. Bohužel mi nejsou známy výsledky jiných systémů pro tuto datovou sadu, takže není možné provést nějaké objektivní srovnání. Uvážíme-li, že vyhodnocování je prováděno tou nejjednodušší možnou metodou, kdy se při zpracování řešení neberou v potaz žádné pravděpodobnosti ani hierarchie významů, je takový výsledek docela očekávaný.

Datová sada pro Semeval-1 se od Senseval-3 liší tím, že umožňuje pouze vyhodnocování založené na hrubé granularitě významů. Systém zde rozhoduje mezi pseudo-skupinami, které jsou vytvořeny sloučením několika podobných významů převzatých ze sémantického lexikonu WordNet. Vyhodnocování zde není, stejně jako u Senseval-3, založeno na zohledňování přiřazené pravděpodobnosti správnému významu, ale pouze na přiřazeném významu. Pro tuto datovou sadu bylo dosaženo průměrné úspěšnosti 82,1892%. Podle očekávání je úspěšnost systému výrazně lepší než u předešlé datové sady. Na této datové sadě byla založena i soutěž v rámci letošní konference Semeval-1, které se systém zúčastnil. Protože vyhodnocování konference probíhalo v době, kdy systém ještě nebyl zcela hotov, nebyla tehdy úspěšnost tak velká. Na soutěži obsadil 8. místo z 13 systémů s úspěšností 80,3%. Následuje tabulka 5.5 srovnání úspěšností systému v soutěži na konferenci Semeval-1.

Pořadí	Úspěšnost
1.	88,7 %
2.	86,9 %
3.	86,4 %
4.	85,7 %
5.	85,1 %
6.	85,1 %
7.	83,8 %
8.	80,3 %
9.	79,9 %
10.	79,6 %
11.	74,3 %
12.	53,8 %
13.	52,1 %

Tabulka 5.5: Úspěšnosti systémů pro zjednoznačňování vybraných vícevýznamových slov pro angličtinu na soutěži v rámci konference Semeval-1.

5.2.2 Srovnání obou implementovaných algoritmů

Nejprve se zaměřím na srovnání úspěšností dvou implementovaných metod. Je to jednoduchý algoritmus učení s učitelem (`simple`) a naivní Bayesův klasifikátor (`bayes`). Jak je vidět z tabulky 5.6, naivní Bayesův klasifikátor vykazuje pro obě datové sady vyšší úspěšnost.

Algoritmus	Senseval-3	Semeval-1
`simple`	63,1085 %	81,1791 %
`bayes`	64,5030 %	82,1892 %

Tabulka 5.6: Úspěšnosti obou implementovaných algoritmů.

Je vidět, že rozdíl v úspěšnosti mezi oběma implementovanými algoritmy není velký. Nyní se zaměřím na srovnání obou algoritmů vzhledem ke slovním druhům vícevýznamových slov.

Slovní druh	`simple`	`bayes`
`podst. jména`	61,9042 %	63,6067 %
`slovesa`	63,9454 %	65,1724 %
`příd. jména`	41,5596 %	40,3589 %

Tabulka 5.7: Úspěšnosti obou implementovaných algoritmů v závislosti na slovních druzích vícevýznamových slov. Měřeno na datové sadě Senseval-3.

Z tabulky 5.7 vyplývá, že nejmenší úspěšnost měly oba dva systémy při určování významů přídavných jmen. Ve všech kategoriích slovních druhů (kromě této) vedl naivní Bayesův klasifikátor. Tato odchylka je způsobena spíše náhodnou chybou, než menší vhodností naivního Bayesova klasifikátoru pro přídavná jména.

V tabulce 5.8 je přehledně zobrazeno srovnání úspěšností obou systémů z hlediska počtu instancí v trénovací množině dat.

Počet instancí	`simple`	`bayes`
0-50	43,3181 %	47,4565 %
50-100	58,1685 %	58,7537 %
100-150	65,7971 %	65,9821 %
150-200	68,1938 %	66,6547 %
200-250	64,2686 %	67,0584 %
>250	59,5435 %	64,2035 %

Tabulka 5.8: Úspěšnosti obou implementovaných algoritmů v závislosti na velikosti množiny trénovacích dat. Měřeno na datové sadě Senseval-3.

Je vidět, že u obou systémů je úspěšnost závislá na velikosti trénovací množiny. S narůstající množinou trénovacích dat u obou algoritmů úspěšnost nejprve roste a pak začíná klesat. Maximální úspěšnost se nachází u každého systému pro rozdílný počet instancí trénovací množiny. U jednoduché metody `simple` je to 150 až 200 instancí a u metody `bayes` 200 až 250 instancí. Při více instancích ve vstupní trénovací množině dat již dochází k přetrénování obou algoritmů a poklesu jejich úspěšnosti. Z tabulky 5.8 je vidět, že naivní Bayesův klasifikátor je více robustnější a s problémem přetrénování se vyrovná lépe.

Nyní se pokusím provést zhodnocení vlivu použití lemmat. V tabulce 5.9 uvádím hodnoty úspěšností obou implementovaných algoritmů při použití lemmat a bez použití lemmat.

Algoritmus	**bez lemmat**	**s lemmaty**
`bayes`	64,3763	64,5030
`simple`	63,0578	63,0325

Tabulka 5.9: Úspěšnost obou implementovaných algoritmů v závislosti na použití lemmat. Měřeno na datové sadě Senseval-3.

Z hodnot je vidět, že zavedení lemmat nemá u obou algoritmů téměř žádný vliv na celkovou úspěšnost systému. Nicméně zavedení lemmat způsobí zrychlení systému a snížení jeho paměťové náročnosti. Bylo zjištěno, že zavedením lemmat pro datovou sadu Senseval-3 klesl počet zpracovávaných slov z 42980 na 32257.

5.2.3 Příklad špatného určení významu systémem

Nyní se pokusím ukázat názorný příklad určení špatného významu naivním Bayesovým klasifikátorem. Stručně zde popíšu postup určování významu. Pokusím se zjistit příčiny selhání naivního Bayesova klasifikátoru a navrhnout opatření, která by tyto nedostatky eliminovala.

V příkladu předvedu určení významu systémem pro slovo `disc`. Všechny existující významy tohoto slova jsou uvedeny v tabulce 3.1. Jako příklad jsem vybral následující instanci testovacích dat Senseval-3.

It has to be something important to you. I thought: Will I get to surf Sunset before I leave? I tossed the three discs. The combination of different faces produced a score which

required me to inscribe a divided line or a single unbroken line on a sheet of paper. After six throws I had a hexagram which looked like this:

Pro slovo `disc` systém určil význam ve smyslu `disc%music`, zatímco správný význam je ve smyslu `disc%shape`. Rodilému mluvčímu tento příklad nepůsobí jistě žádné velké potíže. Při rozhodování systémem však dojde k selhání. Nejprve se podívejme na ohodnocení, které systém jednotlivým významům přiřadil.

Význam	Ohodnocení
`disc%plate`	-30,6123
`disc%music`	-27,7916
`disc%computer`	-34,4026
`disc%shape`	-42,8523

Tabulka 5.10: Ohodnocení pro jednotlivé významy slova disc. Pozn. Nejlepší ohodnocení je reprezentováno nejvyšší hodnotou.

Z tabulky 5.10 je zřejmé, že v tomto případě se klasifikátor správnému významu ani nepřiblížil. Navíc ohodnocení pro správný význam bylo nejhorší ze všech ohodnocení. Tato skutečnost je způsobena již samotnou instancí testovacích dat. Pokud se na tuto instanci podíváme, tak zjistíme, že vícevýznamové slovo se váže pouze k velmi malému kontextu. Kontext, podle kterého nejspíše určíme správný význam slova je následující:

Will I get to surf Sunset before I leave? I tossed the three discs.

Můj systém pracuje s celým kontextem slova, který je v tomto případě do značné míry irelevantní, což do rozhodování vneslo značnou chybu. Podívejme se nyní na slova, která pro určení významu systém označil jako klíčová.

Slovo	Ohodnocení	Pořadí v žebříčku
`single`	21,4815	6.
`score`	20,5714	102.
`require`	14,6518	538.
`different`	12,2239	797.
`three`	11,4493	836.
`produce`	10,0129	922.
`think`	9,5599	973.
`after`	9,3001	983.

Tabulka 5.11: Klíčová slova vybraná pro daný příklad systémem. Obsahuje klíčová slova, jejich ohodnocení a pořadí v celkovém žebříčku.

Z tabulky 5.11 je patrné, že nebylo vybráno téměř žádné klíčove slovo z relevantního kontextu. Největší vliv na rozhodování zde mají slova `single` a `score`, která ale zcela jistě nepřispívají svým charakterem k určení správného významu. Ostatní klíčová slova se již takovou mírou nepodílejí na určování významu. Vliv těchto slov není tak velký kvůli jejich umístění ke konci žebříčku.

Velikost kontextu tedy v tomto případě hraje významou roli. Zvážíme-li např. jinou testovací instanci pro slovo `disc` se správným významem ve smyslu `disc%music`, vidíme, že zde má na správný význam vliv téměř celý kontext.

Philips, the Dutch electronics company which invented the CD, long ago dropped its original advertising slogan: Perfect sounds forever. a spokesman said yesterday that the company promised a lifetime's use if the CDs were manufactured correctly. Some manufacturers guarantee CDs for only 30 years but Mr Bert Gall, general manager of optical systems at Philips, said the life of the average disc should be more than 1,000 years, though it could be reduced to 50 if manufacturing was poor and the discs were mistreated. The main problem is that the stored sound can be lost when the thin layer of aluminium which coats a CD's plastic surface is oxidised losing its ability to reflect laser light. European CD manufacturers have begun a regular exchange of information on quality to help improve the discs longevity and hope to agree on an international standard testing procedure involving exposure of the discs to heat and humidity.

Tato rozsáhlá instance má naprosto odlišný charakter. Zde se na určování významu podílejí i slova, která jsou hodně vzdálená od vícevýznamového slova. Je zřejmé, že tyto dvě instance si svým charakterem odporují. Je proto velmi těžké pracovat s takto odlišnými instancemi při použití konstantní velikosti zpracovávaného kontextu.

5.2.4 Návrh odstranění některých nedostatků

Z předcházejícího příkladu se naskýtá následující otázka. Jaký může mít vliv velikost zpracovávaného kontextu na úspěšnost celého systému? Je zřejmé, že v některých případech by to při dostatku trénovacích dat mohlo výsledek zlepšit. Je však tato metoda vhodná i pro datové sady Senseval-3 a Semeval-1? Podle rozmanitosti jednotlivých instancí obsažených v těchto datových sadách je odpověď nejistá. Rozhodl jsem se proto otestovat vliv velikosti kontextu na úspěšnost systému a dodatečně jsem implementoval možnost nastavení velikosti zpracovávaného kontextu. Tabulka 5.12 ukazuje vliv velikosti kontextu na úspěšnost systému.

Velikost kontextu	**Ohodnocení**
2	61,7901 %
5	61,5112 %
10	61,8408 %
20	62,4746 %
50	64,4523 %
100	64,5030 %
`max`	64,5030 %
`auto`	63,0325 %

Tabulka 5.12: Úspěšnosti systému v závislosti na velikosti uvažovaného kontextu. Měřeno na datové sadě Senseval-3.

Hodnota velikosti kontextu zde označuje maximální velikost kontextu na jednu stranu. Celkový kontext pro hodnotu 20 tedy může být maximálně 40 slov. Úspěšnost programu je pro dané datové sady nejlepší, pokud systém počítá s úplně celým kontextem nebo

alespoň s kontextem blížícím se maximální možné hodnotě. Pokud budeme velikost kontextu snižovat, dojde i ke snížení úspěšnosti systému. Další alternativou je pak použití pro každou instanci zcela jiného kontextu omezeného např. na větu, ve které se vícevýznamové slovo vyskytuje. Tento způsob práce s kontextem je uveden v tabulce 5.12 pod názvem `auto`. Je vidět, že ani tento na první pohled zajímavý způsob nepřináší žádné zlepšení. U druhé metody `simple` se také neprojevilo žádné zvýšení úspěšnosti při snižování velikosti použitého kontextu. Pokles úspěšnosti zde byl ještě markantnější. Žádné zlepšení nepřineslo ani nastavení odlišné velikosti kontextu pro trénovací a testovací data. Je vidět, že tyto datové sady obsahují příliš různorodá data, pro která nepřináší omezení kontextu vícevýznamového slova žádné výhody.

Dalším možným zlepšením by mohlo být zavedení odlišného výběru klíčových slov, založeném na jiných hodnotách než směrodatné odchylce a maximální hodnotě podmíněných pravděpodobností výskytů slov pro jednotlivé významy. Je docela obtížné vybrat vhodná klíčová slova, podle kterých bude klasifikátor rozhodovat. Jak bylo vidět v předcházejícím příkladě, má na správné určení významu velký vliv kvalita trénovacích dat. Pokud bude tato množina různorodá jako v mnou používaných datových sadách, může být výběr slov sebelepší, ale kvůli různé velikosti kontextu to nebude mít na úspěšnost systému velký vliv.

Otázkou tedy zůstává, kde vzít kvalitní trénovací množinu dat, která by pomohla zlepšit úspěšnost rozpoznávání slovních významů. Jak je uvedeno výše, všechna data používaná mým systémem pocházejí z korpusů. V korpusech se objevují texty z mnoha různých zdrojů, a je proto těžké zaručit jejich kvalitu. Pokud nemůžeme zaručit stejnou nebo podobnou délku relevantního kontextu, bylo by určitým zlepšením alespoň zvětšit počet instancí v trénovacích datech.

5.3 Technické parametry systému

Celý systém `wsd` je napsán v jazyce C++ a využívá výhod objektového programování. Pro načítání XML souborů je použit volně dostupný XML parser TinyXML. Program pracuje se dvěma datovými sadami – Senseval-3 a Semeval-1. Pro jejich použití je nutné nastavit typ použité datové sady jako parametr programu. Přes parametr programu lze také pracovat s oběma implementovanými algoritmy. Nastavování parametrů je možné provést pomocí konfiguračního souboru, který je popsán v souboru `README`. Program je funkční pod platformami MS Windows a Linux.

Součástí systému je také program `lemmatizer`, který provádí generování lemmat pro vstupní datovou sadu. Tento program vyžaduje instalaci sémantického lexikonu WordNet a je funkční pouze pod platformou MS Windows. Protože je však generování lemmat pro celou vstupní datovou sadu časově náročné, jsou soubory s lemmaty pro obě používané datové sady přiloženy k programu. K tomuto programu je také přiložen návod ve formě `README` souboru.

Jelikož program pracuje s rozsáhlými datovými sadami obsahujícími tisíce instancí dat, je značně časově i paměťově náročný. Paměťovou i časovou náročnost programu lze snížit vhodným nastavením parametrů programu či použitím lemmat. Tato nastavení snažící se omezit paměťovou nebo časovou náročnost programu mohou mít však v určitých mezích negativní vliv na úspěšnost celého systému. Snížit paměťové nároky lze nejlépe omezením velikosti žebříčku kandidátních slov pomocí parametrů `min_dev` a `min_count` nebo omezením velikosti zkoumaného kontextu slova.

Kapitola 6

Závěr

Oblast zpracování přirozeného jazyka prošla za poslední desetiletí značným vývojem. Nejinak je tomu i u problému zjednoznačňování slovních významů. Během této doby byly prosazovány různé metody a přístupy. Některé byly úspěšné již ve své době, jiné si na svůj průlom musely kvůli nedostatku výpočetních zdrojů několik let počkat a některé metody se neujaly vůbec. V poslední době se ve zjednoznačňování slovních významů využívá obrovských ručně anotovaných zdrojů dat zvaných korpusy. Současným trendem je do jisté míry odloučení se od ruční anotace a celý proces automatizovat. Využívá se přitom stále více internetu, který slouží jako obrovský korpus. Internet společně s kombinací vhodných algoritmů (např. bootstrapping) představuje v současnosti jedno z nejlepších možných řešení.

Mým úkolem bylo vytvořit systém, který vykonává zjednoznačňování slovních významů libovolným algoritmem strojového učení. Ve svém systému jsem zvolil naivní Bayesův klasifikátor a k němu jako vhodný doplněk jednoduchou metodu využívající strojového učení s učitelem, která je založená na principu tvorby jistých skupin slov pro každý význam. Obě metody jsou v mé práci vysvětleny a vzájemně srovnány. Pro testování úspěšností obou těchto algoritmů jsem zvolil ze všech možností snad tu nejdostupnější. Oba algoritmy jsem testoval na datových sadách poskytnutých v rámci konferencí Senseval-3 a Semeval-1, které se již dlouhá léta zabývají zjednoznačňováním slovních významů.

Svůj systém jsem rovněž přihlásil do soutěže v rámci konference Semeval-1 pro řešení úkolu zjednoznačňování vybraných vícevýznamových slov pro angličtinu. V této soutěži jsem mohl svůj systém srovnat s jinými existujícími systémy podle přesně definovaných kriterií vyhodnocování. Můj systém při srovnání s ostatními nijak nevybočoval z mezí očekávaných úspěšností a umístil se někde kolem středu. Je vidět, že existují systémy, které dokáží na daných datových sadách určovat významy slov o pár procent lépe.

Nabízí se otázka k zamyšlení. Jak zvýšit úspěšnost systémů vykonávajících zjednoznačňování slovních významů? Systémy na konferenci Semeval-1 jsou zcela jistě založeny na metodách strojového učení, které jsou vhodně doplněny různými informacemi, poskytovanými některými bázemi znalostí. Vyšší úspěšnosti je pravděpodobně také dosaženo trénováním algoritmů na jiných než dodaných datových sadách. Pravidla soutěže totiž nijak nezakazují použití jakýchkoliv externích korpusů či bází znalostí. Nabízí se zde proto i použití vhodných informací ze sémantického lexikonu WordNet. Principy všech použitých systémů mi zatím nejsou známy a budou představeny až na konferenci Semeval-1, která se koná koncem června 2007. Ať už je lepší úspěšnosti jiných systémů dosaženo jakkoliv, je zřejmé, že za zvýšením úspěšnosti o každé procento stojí desítky a desítky hodin práce.

Literatura

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

[2] David Yarowsky Philip Resnik. *A Perspective on Word Sense Disambiguation Methods and Their Evaluation*. 1997.

[3] Ted Pedersen Rada Mihalcea. *Advances in Word Sense Disambiguation*. Tutorial at AAAI-2005, 2005.

[4] Marie Sochrová. *Český jazyk v kostce pro střední školy*. Fragment, 1999.

[5] WWW stránky. Natural language processing. `http://en.wikipedia.org/`.

[6] WWW stránky. Open mind word expert. `http://teach-computers.org/`.

Dodatek A

Přílohy

Příloha 1: Datový nosič CD se zdrojovými kódy, programovou dokumentací, souborem README, konfiguračním souborem a elektronickou kopií této technické zprávy.