Určení typu pojmenované entity

Jana Kravalová, gris@ucw.cz

14. února 2008

1 Zadání

Úkolem bylo určit typ jednoslovné pojmenované entity podle zadané klasifikační metodologie navržené kolektivem Ševčíková et al. (pozn: technická zpráva TR-2007-36 „Zpracování pojmenovaných entit v českých textech“). Zvolila jsem první, hrubší úroveň klasifikace, pojmenované entity jsem klasifikovala do tříd `a, c, g, i, m, n, o, p, q, t`, viz tabulka 1.

Tabulka 1: Klasifikační metodologie

a	čísla v adresách
c	bibliografické položky
g	zeměpisné názvy
i	instituce
m	média
n	specifická použití čísel
o	názvy artefaktů
p	osobní jména
q	množstevní výrazy
t	čas

Některé pojmenované entity byly označeny dalšími pomocnými značkami, jejichž popis uvádím v tabulce 2.

Tabulka 2: Pomocné značky

s	zkratka
f	slovo z cizího jazyka
segm	slovo napsáno velkými písmeny v důsledku chybné segmentace textu
cap	slovo napsáno velkými písmeny např. z typografických důvodů
lower	slovo napsáno chybně s malým písmenem
upper	slovo napsáno chybně s velkým písmenem
?	entita nespecifikovatelného typu, nezařaditelná
!	věta neanotována

2 Data

Data ve formátu `XML` byla rozdělena na trénovací data `train*xml`, vývojová data `dtest*xml` a testovací data `etest*xml`. Data sestávala z textu, ve kterém bylo pro každé slovo určena jeho forma v textu, jeho lemma určené morfologickou analýzou a 15 morfologických kategorií. Pojmenované entity byly vyznačeny a pro každou bylo uvedeno správné (ruční) přiřazení typu `a, c, g, i, m, n, o, p, q, t`.

Některé entity měly navíc uvedeno pomocné označení `s, f`, v takovém případě jsem entitě ponechala její klasifikaci a toto pomocné označení jsem odstranila. Instance, které byly označeny jednou ze značek `segm, cap, lower, upper, ?, !` k sobě neměly asociovánu žádnou pojmenovanou entitu. Např. v případě `segm` se jedná o slovo s velkým písmenem, které vzniklo chybnou segmentací vět, nikoli o pojmenovanou entitu. Instance typu `segm, cap, lower, upper, ?, !` jsem tedy z dat odstranila.

Některé pojmenované entity měly dvojí označení, např. pojmenovaná entita „Litoměřice“ ve spojení „fotbalový spolek Litoměřice“ získá označení `i` jako instituce a `g` jako místní název. (Poznámka: Nejčastějším typem dvojího označení je právě kolize typů `i` a `g`.) V takovém případě bylo nutné ponechat jenom jednu klasifikaci, zvolila jsem si náhodně jednu ze dvou možných.

Velikost trénovacích, vývojových a testovacích dat uvádím v tabulce 3. Rozložení tříd v původních trénovacích datech a upravených datech je uvedeno v tabulce 4.

Tabulka 3: Počty instancí v datech

data	původní	upravená
trénovací	6109	5302
vývojová	823	715
testovací	803	687

Z tabulky je zřejmé, že v trénovacích datech se nacházelo celkem 763 instancí označených třídami mimo klasifikaci `s, f, segm, cap, lower, upper, ?, !`. Zbylých 44 instancí, které tvoří rozdíl mezi velikostí původních a upravených dat, jsou instance, které měly dvojí označení.

Metody byly natrénovány na trénovacích datech `train*xml`, pokud bylo potřeba nastavit různé konstanty, jako například k u metody k-NN, hledala jsem je na vývojových datech `dtest*xml`. Stejně tak výběr klasifikačních rysů jsem prováděla odhadem pomocí výsledků na vývojových datech `dtest*xml`. Po vyladění metod na vývojových datech jsem spustila metody na testovací data `etest*xml`.

Z informací, které byly v datech k dispozici, bylo možné extrahovat různé typy klasifikačních rysů závisejících na formě, lemmatu, či morfologických kategoriích pojmenované entity a jejího kontextu. Zde uvádím možné klasifikační rysy, které by z daných dat bylo možné získat, i když ne všechny jsem nakonec použila:

- **forma pojmenované entity** – tu jsem vůbec nepoužila. V českých datech by se podle mého názoru každá forma objevila příliš vzácně, protože čeština jako flektivní jazyk má pro každé lemma hodně forem.

- **lemma pojmenované entity** – lemma pojmenované entity bychom jistě dokázali využít například slovníkovým způsobem: Najdeme-li lemma `Praha` ve slovníku měst, nejspíš se jedná o město. Samozřejmě mohou existovat výjimky, jako například `hotel Praha`, ale

Tabulka 4: Popis rozložení tříd v trénovacích datech

třída	původní data	upravená data
a	23	23
c	0	0
g	1133	1133
i	451	416
m	79	79
n	0	0
o	297	289
p	2632	2631
q	0	0
t	731	731
s	385	0
f	18	0
segm	255	0
cap	55	0
lower	7	0
upper	4	0
?	39	0
!	0	0

například pro křestní jména by tato metoda mohla být užitečná. Její nevýhodou bude nutnost nějakým způsobem slovníky získat, ať už najít předzpracované slovníky, nebo se pokusit slovník vygenerovat z trénovacích dat. Navíc slovníky budou asi velmi velké. Další zajímavou informací je typ pojmenované entity vyznačený ručně v lemmatu slova za `_;`, např. `_;G` pro zeměpisné názvy. Nakonec se nabízí velká skupina pravdivostních (0/1) rysů popisujících, jak slovo vypadá, jak je napsáno, např. „Začíná slovo na velké písmeno“, „Je to zkratka, tj. všechna písmena velká“, apod. Tyto rysy dokážeme pomocí regulárních výrazů snadno získat.

- **morfologické kategorie pojmenované entity** – kategoriální rysy slovní druh, poddruh, číslo, rod, pád a další.

- **kontext** – informace o okolních slovech, jejich lemma, jejich slovní druhy, případně sofistikovanější rysy jako vzdálenost od začátku věty a podobně. Velikost kontextu je samozřejmě nutné dobře zvolit, protože příliš velký kontext kvůli řídkosti dat nebude příliš informativní.

V kapitole o praktickém řešení této úlohy popisuji, jakým způsobem jsem vybrala nejúspěšnější kombinaci klasifikačních rysů. Přesný seznam použitých klasifikačních rysů lze nalézt v tabulce 11 v příloze.

3 Teoretický popis řešení

3.1 Rozhodovací stromy

Rozhodovací strom sestavený nad množinou instancí popsaných klasifikačními rysy je strom, kde v každém uzlu jsou data rozdělena na disjunktní množiny podle hodnoty vybraného klasifikačního rysu. Klasifikace pak probíhá postupným procházením stromu od kořene přes uzly, kde se rozhoduje, do kterého podstromu daná instance náleží podle hodnoty klasifikačního rysu náležející danému uzlu. Výsledná klasifikace je uložena v listu stromu. Důležitý je způsob výstavby stromu. V každém uzlu je třeba rozhodnout, podle kterého klasifikačního rysu se mají data rozdělit. K rozhodování slouží funkce, které si lze v programu `R` vybrat a kterým lze dále upravit parametry. Pokud se nenastaví žádná konkrétní funkce, program `R` se pokusí inteligentně některou funkci vybrat.

Rozhodovací strom lze zjednodušit, případně i zvětšit jeho úspěšnost pomocí následného ořezávání (pruning) příliš specifikovaných větví.

3.2 k-Nearest Neighbour

Metoda k-Nearest Neighbour je příkladem učení založeného na příkladech. Při tomto způsobu učení nedefinujeme žádnou explicitní vyhodnocovací funkci, trénovací příklady si pouze uložíme a při klasifikaci nového příkladu se pokusíme tento příklad ohodnotit podle již viděných trénovacích příkladů, které jsou „nejvíce podobné“.

Konkrétně v metodě k-NN vytvoříme mnohadimenzionální prostor, kde každému klasifikačnímu rysu odpovídá jedna dimenze (jedna osa). Příklady poté „rozmístíme“ do tohoto prostoru podle jejich hodnot klasifikačních rysů a při klasifikaci nového, testovacího příkladu, najdeme takové příklady, které jsou tomuto příkladu nejblíže. Metrikou pro určení vzdálenosti je v programu `R` eukleidovská vzdálenost. Pokud nastane rovnost hlasů, vybere se náhodně jedna z navrhovaných klasifikací.

Ohodnocení nového příkladu je pak takové, jaké převládá mezi k nejbližšími trénovacími příklady. Konstantu k jsem odhadla vyzkoušením všech možností v intervalu $[1, 30]$ a porovnáním výsledků na vývojových datech.

3.3 SVM klasifikace

Metoda SVM umístí všechny trénovací příklady do vektorového prostoru a pak nalezne (lineární) separátor. Separátor je nadrovina, která v prostoru odděluje různě klasifikované příklady od sebe. Nejlepší lineární separátor je ten, který je maximálně vzdálen od trénovacích příkladů, duální úloha je pak zadána funkcí $min_\alpha \frac{1}{2}\alpha^T y_i y_j K(x_i, x_j)\alpha - e^T\alpha$, kde $K(y_i, y_j) = \theta(x_i)\theta(x_j)$ je transformace prostoru tak, abychom mohli hledat lineární separátor. Způsob transformace prostoru je určen tzv. kernel funkcí. Program `R` používá jako výchozí RBF (radial basis function): $K(x, y) = exp(-\gamma||x - y||^2), \gamma \geq 0$. Výchozí hodnota γ je nastavená na 1/(dimenze klasifikačního vektoru). Pokud data nejsou vůbec separabilní, zavede se cena za překročení separátoru a nový vztah pro lineární separátor počítá i s touto cenou. Výchozí hodnota konstanty C, která odpovídá ceně za překročení separátoru, je v programu `R` rovna 1.

V případě většího počtu tříd, jako v této úloze, kde klasifikujeme do tříd `a, g, i, m, n, o, p, q, t`, program `R` používá přístup „one-to-one“, při kterém se natrénuje $\frac{k(k-1)}{2}$ binárních klasifikátorů a výsledná klasifikace se vypočítá hlasováním všech klasifikátorů.

4 Praktické provedení

4.1 Implementace

Součástí implementace jsou následující skripty:

- Pro předzpracování textu v `XML` formátu jsem použila `awk` skript `prep.awk`.
- Extrakci klasifikačních rysů provádí perlovský skript `prep.pl`.
- Pro samotné učební metody jsem použila program `R`, zdrojový kód lze nalézt ve skriptu `ml.r`.
  - **rozhodovací stromy** – `R` balíček `rpart`.
  - **k-Nearest Neighbour** – `R` balíček `class`, funkce `knn()`
  - **SVM classification** – `R` balíček `e1071`, funkce `svm()`, konkrétně jsem použila `C-classification`
- Celou sestavu lze spustit shellovým skriptem `run.sh`

4.2 Výběr klasifikačních rysů

Klasifikační rysy jsem zjišťovala v rámci kontextu pojmenované entity, tj. slov sousedících s pojmenovanou entitou v povrchovém slovosledu. Do kontextu jsem zahrnula dvě slova, která se v povrchovém slovosledu nacházela před pojmenovanou entitou, samotnou pojmenovanou entitu a jedno slovo za pojmenovanou entitou. Označíme-li pojmenovanou entitu jako w_i, kde i označuje pořadí v dokumentu, pak její kontext jsou slova w_{i-2}, w_{i-1}, w_i, w_{i+1}. Pro všechna slova z kontextu (tedy dohromady pro čtyři slova) jsem zjistila jejich morfologické vlastnosti. Zde je namístě poznamenat, že jako možné hodnoty morfologických značek jsem použila pouze hodnoty nalezené v trénovacích datech, nikoli všechny možné hodnoty.

Dalším rysem je typ pojmenované entity tak, jak je označen v lemmatu za znaky `_;`, konkrétně jsem si vybrala `_;Y`, `_;S`, `_;E`, `_;G`, `_;K` a `_;R`. Ostatní možné popisy se mi zdály příliš podrobné a skutečně po pokusném přidání do klasifikačních rysů nezvýšily úspěšnost.

Experimentovala jsem i s vlastnostmi slov, které by popisovaly, jak slovo vypadá, jak se píše. Použila jsem pravdivostní (0/1) rys „Začíná pojmenovaná entita velkým písmenem?“ a pravdivostní rys „Je slovo zkratka (je napsáno velkými písmeny)?“. Tyto rysy přinesly určité zlepšení, ale další podobné rysy už se mi nepovedlo úspěšně zapojit.

Také jsem experimentovala se slovníkovými rysy, které by popisovaly, zda už jsem dané slovo (lemma, formu) viděla a jakou mělo hodnotu, ale nepodařilo se mi pomocí takových rysů dosáhnout výrazného zlepšení, naopak zapojení těchto slovníkových rysů bylo velmi náročné na výpočet. Může to být tím, že slovník jsem si sama vytvořila z trénovacích dat a byl tedy příliš malý. Lepší by asi bylo zapojit například do slovníku všechna křestní jména z kalendáře, apod.

Všechny kategorické klasifikační rysy jsem převedla na číselné binární vektory, kde pozice ve vektoru odpovídá možné hodnotě klasifikačního rysu a 1 se nabývá právě v hodnotě, která platí pro danou instanci. Toto vyjádření je nezbytné pro matematické metody jako je SVM a k-NN. Ukázalo se, že převedení kategoriálních rysů na číselné binární vektory mělo pozitivní vliv na výpočet rozhodovacích stromů. Při ponechání kategoriálních rysů trvalo trénování rozhodovacích stromů velmi dlouho a výsledky byly při porovnání s metodami SVM a k-NN velmi slabé.

I když vytvořené binární vektory byly velké (dimenze 150 až 200), **převedením kategoriálních klasifikačních rysů na číselné binární vektory dosáhly v této úloze rozhodovací stromy úspěšnosti srovnatelné s metodami SVM a k-NN.**

Nakonec jsem se pokusila zjednodušit výsledné modely. Ukázalo se, že pro metodu SVM a k-NN je výhodné používat pouze morfologické značky č. 1,3,4,5, tedy slovní druh, rod, číslo a pád. Ostatní značky úspěšnost na vývojových datech nezvyšovaly a jejich odstraněním došlo k zjednodušení modelů. Výběr vhodných morfologických rysů jsem provedla tak, že z původní velké množiny všech dostupných morfologických rysů v kontextu jsem pokusně odebírala jednotlivé morfologické rysy, a pokud se úspěšnost nezhoršila, daný klasifikační rys jsem úplně odstranila. Na rozhodovací stromy neměl výběr klasifikačních rysů vliv, tvar i úspěšnost rozhodovacího stromu byla stejná při použití všech dostupných klasifikačních rysů i při ručním výběru jistých klasifikačních rysů. Rozhodovací strom jednoduše zbytečné klasifikační rysy nezarhnul do rozhodovacího procesu. Závěr tedy je, že **v této úloze pro metody SVM a k-NN vede ruční výběr vhodných klasifikačních rysů k výraznému zvýšení úspěšnosti, kdežto rozhodovací stromy dokáží samy eliminovat zbytečné klasifikační rysy**, a nepotřebují tedy ruční výběr.

V tabulce 11 v příloze uvádím přesný popis použitých klasifikačních rysů a v tabulce 5 uvádím výsledky jednotlivých pokusů.

4.3 Optimalizace metod

Pro rozhodovací stromy jsem vyzkoušela prořezávání (pruning) implementované v programu `R` funkcí `prune()`. Protože strom, který vznikl nad ručně vybranou podmnožinou klasifikačních rysů je stejný jako strom, který vznikl nad úplnou množinou klasifikačních rysů, je jedno, který strom se rozhodneme prořezat. **Protože už samotný rozhodovací strom pro tuto úlohu je velmi jednoduchý (má pouze 5 uzlů), nedošlo při pruningu k žádnému zjednodušení stromu.** Výsledný rozhodovací strom lze nalézt na obrázku 2.

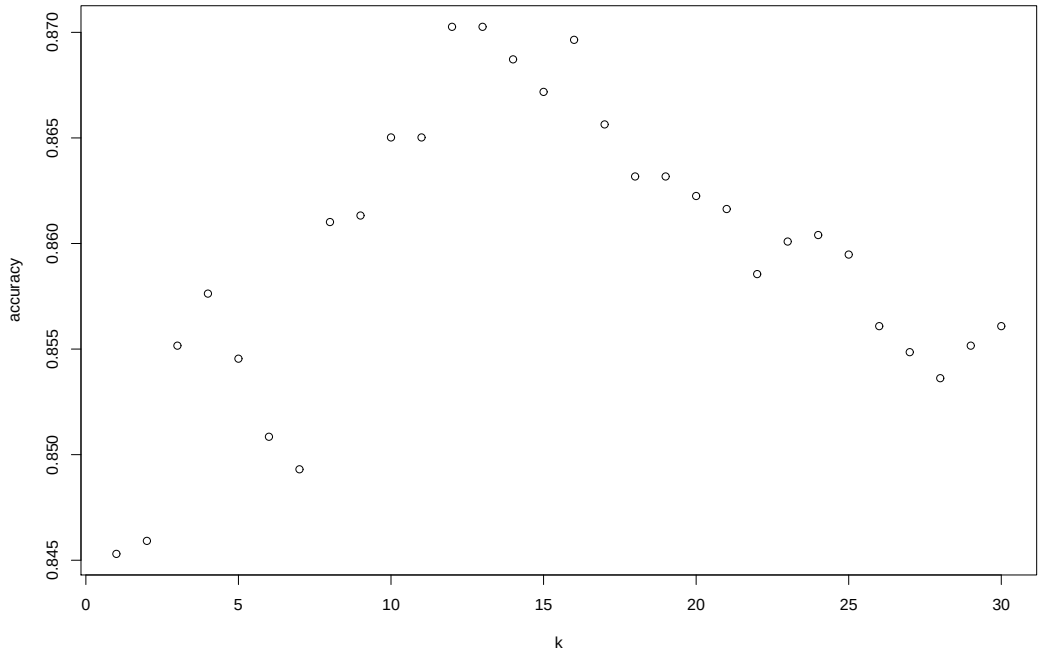
Pro metodu k-NN jsem hledala vhodné k podle výsledků na vývojových datech. Vyzkoušela jsem všechny celočíselné hodnoty v intervalu $[1, 30]$. V grafu 1 uvádím závislost úspěšnosti na zvolené hodnotě k.

Metoda SVM je velmi citlivá na správné nastavení tzv. kernel funkce, jejích paramatrů a ostatních parametrů (cena za překročení separátoru, aj.). Zkoušela jsem různé kombinace parametrů, ale všechny dávaly mnohem horší výsledky. Při jakékoli změně parametru γ, C nebo při změně jádra spadla úspěšnost na $50 - 70\%$, takže jsem nechala nastavení, které je výchozí v programu `R`.

V programu `R` existuje funkce `tune`, která má poddruhy `tune.rpart()`, `tune.knn()`, `tune.svm()` a která nabízí optimalizaci všech konstant a dokonce nastavení vah jednotlivých klasifikačních rysů, takže například funkce v rozhodovacích stromě by se pak na základě vážených klasifikačních rysů lépe rozhodovala, jak stavět strom. Pro metodu k-NN by takové vážení rysů bylo ještě užitečnější, protože více důležitým rysům by se prodloužily osy v eukleidovském prostoru. Funkci `tune()` se mi nepodařilo dostatečně využít, její trénování trvalo neúnosně dlouho. Nejspíš je to daň za příliš velké klasifikační vektory, které používám. **Pro tuto úlohu se použití funkce `tune()` ukázalo jako technicky nevýhodné.**

Alespoň pro metodu k-NN jsem se pokusila udělat primitivní vážení klasifikačních rysů. Vycházela jsem z předpokladu, že nejdůležitější jsou rysy odpovídající samotné pojmenované entitě a potom pro vzdalující se slova v kontextu důležitost klesá. Proto jsem všechny rysy odpovídající bližším slovům násobila většími konstantami, naopak vzdálená slova malými kon-

Obrázek 1: Úspěšnost metody k-NN v závislosti na konstantě k



stantami. V prvních experimentech bylo toto vážení velmi užitečné, v konečném modelu už se výrazné zlepšení neprojevilo, takže v nejúspěšnějším modelu žádné vážení není implementováno (resp. všechny váhy jsou nastavené na 1).

5 Výsledky a diskuze

Jako vyhodnocovací funkci používám procento instancí, které jsou správně klasifikovány.

Pro porovnání algoritmů Alg_X a Alg_Y používám dvouvýběrový t-test (v `R` funkce `t.test`). Klasifikaci, kterou provádí algoritmu Alg_X, odpovídá binomická náhodná veličina X, kde jednotlivé alternativní pokusy se nabývají 1 v případě správné klasifikace a 0 v případě chybné klasifikace. Klasifikaci, kterou provádí algoritmus Alg_Y, odpovídá nezávislá, stejně rozdělená náhodná veličina Y. Hypotéza H je „Alg_X není lepší než Alg_Y“ $\Leftrightarrow \mu_X \leq \mu_Y$ a alternativní hypotéza A je „Alg_X je lepší než Alg_Y“ $\Leftrightarrow \mu_X > \mu_Y$, při zvolené hladině významnosti 0.95.

Pokud použijeme nejjednodušší možnou metodu, tedy pokud každému testovacímu příkladu přiřadíme nejpravděpodobnější hodnotu, kterou je hodnota `p` (měřeno na trénovacích datech), dosáhneme úspěšnosti 46.71% na vývojových datech a 49.62% na testovacích datech.

Nyní popíšu dílčí výsledky na vybraných experimentech:

1. Experimenty jsem začala tak, že jsem použila morfologickou značku č. 1, 2, 3 a 4 na danou pojmenovanou entitu.

2. Pro kontext $w_{i-2}, w_{i-2}, w_{i-1}, w_i, w_{i+1}$ jsem použila všechny dostupné morfologické značky.

3. Doplnila jsem klasifikační rys, který zohledňoval typ pojmenované entity tak, jak je označen přímo v lemmatu pojmenované entity.

4. Přidala pravdivostní klasifikační rys „Začíná slovo velkým písmenem?“

5. Zkoušela jsem různé kombinace šířek kontextů a morfologických značek, až jsem vybrala kontext $w_{i-2}, w_{i-1}, w_i, w_{i+1}$ a pro tato čtyři slova jsem vybrala morfologické značky č. 1, 3, 4, 5.

Konstantu k, neboli počet zkoumaných sousedů, jsem nastavila na 13, konstanty pro SVM jsem nevyplnila, takže implementace tohoto algoritmu v `R` použila výchozí hodnoty. Výsledky těchto dílčích experimentů uvádím v tabulce 5.

Tabulka 5: Výsledky na vývojových datech pro vybrané experimenty

experiment	rozhodovací stromy	k-NN	SVM
1	72.59%	too many ties	73.15%
2	78.04%	77.06%	79.58%
3	87.69%	81.00%	86.43%
4	87.69%	81.26%	86.71%
5	87.69%	84.20%	88.95%

Poznámka: V metodě k-NN v prvním experimentu nelze při zadaných datech a malém počtu klasifikačních rysů věrohodně určit klasifikaci.

Vidíme, že už použití několika málo klasifikačních rysů v prvním experimetu dává poměrně pěkné výsledky, asi 72%.

Velmi užitečná se ukázala informace o typu pojmenované entity v lemmatu. Při jejím použití došlo k statisticky významnému $4-9\%$ zlepšení. To není překvapivé, protože informace je ručně vytvořená, a měla by tedy být správná.

Informace o tom, jak je slovo napsáno (zda je první písmeno velké a zda je slovo zkratka), přineslo mírné zlepšení o asi 0.2% které sice není statisticky významné, ale použití podobných klasifikačních rysů by mohla být dobrá cesta, jak se vyhnout typickým chybám.

Z výsledků posledního pokusu vyplývá, že je pro metody SVM a k-NN je výhodné ručně vybrat užitečnější klasifikační rysy. Po ručním výběru užitečných klasifikačních rysů ze všech dostupných morfologických kategorií v kontextu několika okolních slov nejenže získáme jednodušší modely a trénování trvá kratší dobu, ale i úspěšnost se významně změní. Rozhodovací stromy ruční výběr klasifikačních rysů nepotřebují.

V tabulce 6 uvádím nejlepší výsledky pro jednotlivé metody. Použila jsem klasifikační rysy z tabulky 11. Konstanta k u metody k-NN je nastavená na 13. SVM používá výchozí kernel funkci s přednastavenými parametry (v programu `R`). Při zvolené hladině významnosti 0.95 je rozdíl mezi úspěšností algoritmů SVM, k-NN a rozhodovacích stromů na testovacích datech statisticky nevýznamný. Lze tedy říct, že **všechny tři algoritmy dávají statisticky srovnatelně dobré výsledky**.

Tabulka 6: Nejlepší výsledky na vývojových a testovacích datech

metoda	vývojová data	testovací data
rozhodovací stromy	87.69%	86.61%
k-NN	84.20%	86.46%
SVM	88.95%	89.67%

6 Popis výsledných modelů

Výsledný rozhodovací strom je znázorněn na obrázku 2, kde jednotlivé uzly V_i odpovídají i-té dimenzi klasifikačního vektoru. Jejich slovní popis je uveden v tabulce 7. Vidíme, že první (nejdůležitější) rozhodnutí rozdělí pojmenované entity na ty, které jsou zeměpisným názvem a ostatní, a kritériem pro rozhodování je informace `_;G` v lemmatu pojmenované entity. Popis chyb rozhodovacího stromu na testovacích datech je uveden v tabulce 8. Zajímavé je, že strom patnáctkrát nerozpoznal zeměpisný název a místo něj přiřadil osobní jméno. Tento typ chyby je překvapivý proto, že víme, jak se strom rozhoduje, zda je pojmenovaná entita zeměpisným názvem: podívá se na lemma na značku `_;G`. Nabízí se vysvětlení, že by příčina mohla být v datech, a skutečně některé pojmenované entity typu `g` nemají v lemmatu uvedenou značku `_;G`.

Popis chyb metody k-NN na testovacích datech lze nalézt v tabulce 9 a popis chyb metody SVM na testovacích datech uvádím v tabulce 10. U všech tří metod je zřejmé, že nejčastější typ chyby je ten, kdy je pojmenované entitě nesprávně přiřazena klasifikace `p`. Odstranění této chyby by nejvíc pomohlo rozhodovacím stromům, protože jiné chyby u nich téměř nenajdeme.

Tabulka 7: Klasifikační rysy použité v rozhodovacím stromu seřazené podle důležitosti

V156	je instance označena v lemmatu jako zeměpisný název, tedy jako `_;G` ?
V78	je slovní druh pojmenované entity číslovka, tj. `C`?
V93	je rod pojmenované entity mužský, tedy `M`?
V157	je instance označena jako „společnost, organizace, instituce“, tj. `_;K`?
V40	je slovní druh slova před pojmenovanou entitou číslovka, tedy `C`?

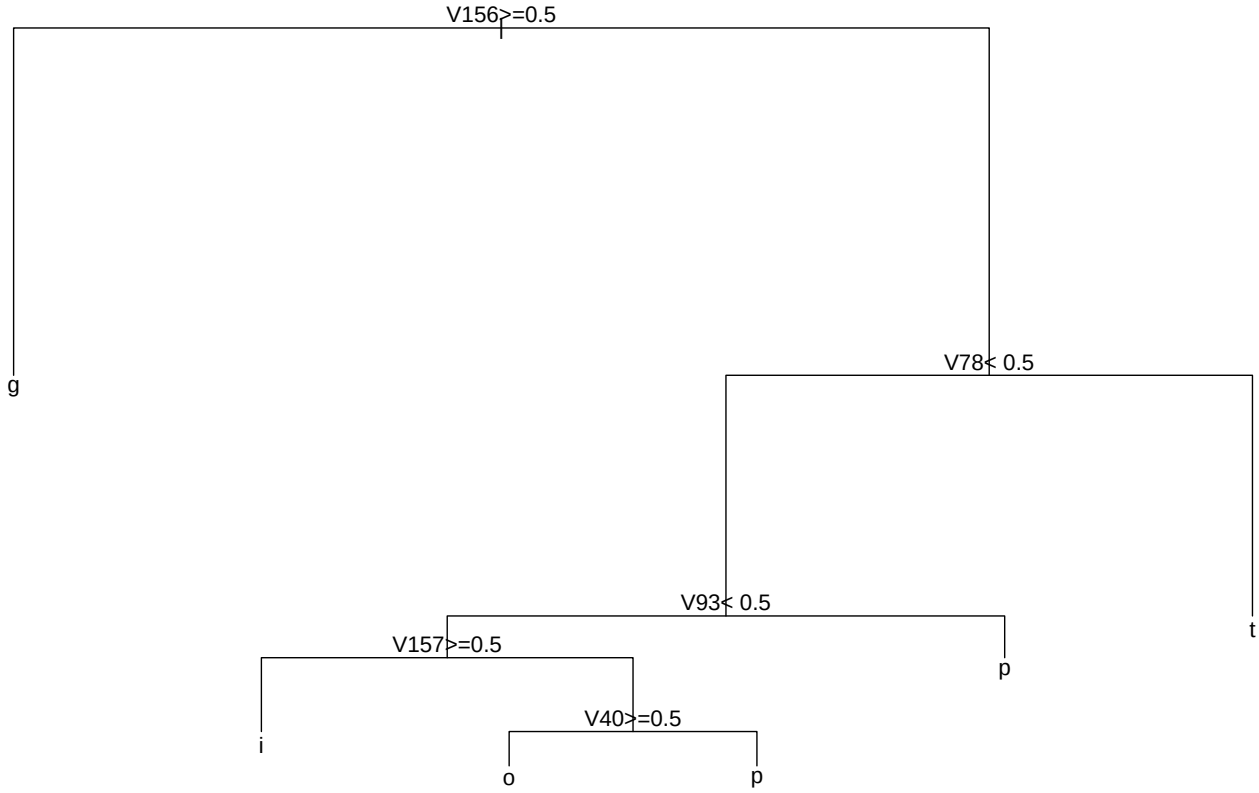
7 Závěr

V této práci jsem řešila problém rozpoznání pojmenované entity podle zadané metodologie Ševčíková et al. Použila jsem metody rozhodovací stromy, k-NN a SVM implementované v programu `R`. Všechny tři metody měly srovnatelně dobré výsledky na testovacích datech. Pomocí rozhodovacích stromů jsem na testovacích datech dosáhla úspěšnosti 86.61%, metodou k-NN 86.46% a metodou SVM 89.67%.

V průběhu řešení jsem došla k několika závěrům:

- Při dobrém nastavení dosáhly všechny tři metody statisticky srovnatelně dobrých výsledků.
- Pro metodu rozhodovacích stromů je lepší používat číselné binární klasifikační rysy než kategoriální klasifikační rysy.

Obrázek 2: Rozhodovací strom



- Pro metodu k-NN a SVM je výhodné ručně vybrat užitečné klasifikační rysy, zatímco rozhodovací stromy si klasifikační rysy dokáží vybrat samy i mezi větším množstvím zbytečných rysů.
- Vhodné klasifikační rysy pro tuto úlohu jsou morfologické značky slov v nejbližším kontextu a typ pojmenované entity vyznačený v lemmatu.

8 Příloha

V tabulce 11 uvádím přesný seznam použitých klasifikačních rysů. Značení w_i znamená samotnou pojmenovanou entitu, w_{i-1} příslušně vzdálené slovo v kontextu, indexace $w_i[1]$ odkazuje na příslušnou morfologickou kategorii (čísluji od 1).

Tabulka 8: Chyby rozhodovacího stromu na testovacích datech

pred ↓, true →	a	g	i	m	o	p	t
a	0	0	0	0	0	0	0
g	0	134	0	0	0	12	0
i	0	1	22	3	2	0	0
m	0	0	0	0	0	0	0
o	0	0	0	0	19	0	0
p	0	15	11	10	23	340	15
t	0	0	0	0	0	0	80

Tabulka 9: Chyby metody k-NN na testovacích datech

pred ↓, true →	a	g	i	m	o	p	t
a	0	0	0	1	0	0	0
g	0	134	9	1	2	12	1
i	0	2	13	2	5	3	0
m	0	0	0	2	0	0	0
o	0	0	3	0	28	2	0
p	0	14	6	6	7	334	8
t	0	0	2	1	2	1	86

Tabulka 10: Chyby metody SVM na testovacích datech

pred ↓, true →	a	g	i	m	o	p	t
a	0	0	0	0	0	0	0
g	0	137	1	0	0	9	1
i	0	2	25	5	9	2	0
m	0	0	0	0	0	0	0
o	0	0	1	0	24	0	0
p	0	11	6	7	10	340	4
t	0	0	0	1	1	1	90

Tabulka 11: Nejúspěšnější kombinace klasifikačních rysů pro metodu SVM a k-NN

klasifikační rys	počet hodnot
$w_{i-2}[1]$	12
$w_{i-2}[3]$	11
$w_{i-2}[4]$	6
$w_{i-2}[5]$	9
$w_{i-2}[10]$	4
$w_{i-1}[1]$	12
$w_{i-1}[3]$	11
$w_{i-1}[4]$	6
$w_{i-1}[5]$	9
$w_{i-1}[10]$	4
$w_i[1]$	12
$w_i[3]$	11
$w_i[4]$	6
$w_i[5]$	9
$w_i[10]$	4
$w_{i+1}[1]$	12
$w_{i+1}[3]$	11
$w_{i+1}[4]$	6
$w_{i+1}[5]$	9
$w_{i+1}[10]$	4
informace v lemmatu	6
první písmeno velké	bool (0/1)
všechna písmena velká	bool (0/1)