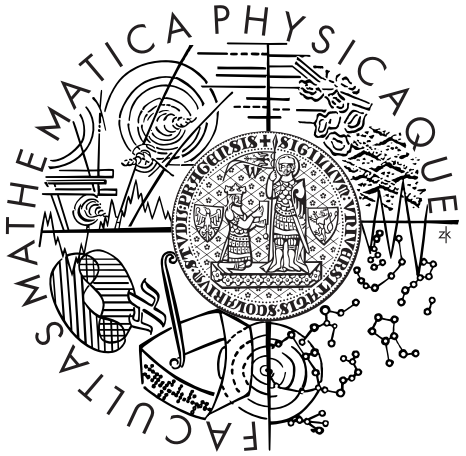
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Martin Kirschner

Automatické vytváření sémantických sítí

Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: RNDr. Pavel Pecina Ph.D.
Studijní program: Informatika
Studijní obor: Matematická lingvistika

Praha 2011

Na tomto místě bych rád poděkoval všem, kteří mají zásluhu na dotažení mého studia až k odevzdání této práce. V první řadě bych chtěl poděkovat mým rodičům a bratrovi za soustavnou podporu po celou dobu studia. Velký dík má ode mě i má snoubenka Ing. Vendula Trávníčková, také za podporu, trpělivost a za spásné nápady v těžších chvílích.

Tuto práci by nebylo možné vypracovat bez inspirativních rad, nápadů a doporučení vedoucího práce RNDr. Pavla Peciny Ph.D. touto cestou mu velmi děkuji. Za čas věnovaný anotaci získaných relací děkuji anotátorům Aleně Šebestové, Mgr. Lukáši Kopencovi a MUDr. Tomáši Boučkovi. Dík patří i všem korektorům a Mgr. Janu Šebestovi za technickou pomoc při zpracování práce.

Na závěr bych rád poděkoval ještě Ústavu formální a aplikované lingvistiky na MFF UK, za poskytnutí přístupu na jejich výpočetní cluster, bez něhož by vypracování této práce také možné nebylo, a Dr. Piaseckému a Dr. Brodovi z Univerzity ve Vratislavi, za vstřícnost a rady při využívání jejich software SuperMatrix.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Automatické vytváření sémantických sítí

Autor: Bc. Martin Kirschner

Katedra: Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: RNDr. Pavel Pecina Ph.D.

Abstrakt: Předložená práce si dává za cíl prozkoumat možnosti automatické konstrukce a rozšiřování sémantických sítí za použití metod strojového učení. Důraz je kladen na postup získávání rysů pro sadu dat. Práce prezentuje metodu získávání sémantických relací, založenou na distribuční hypotéze a trénovanou na datech z Czech WordNetu. Dále jsou prezentovány zatím první výsledky pro český jazyk v této oblasti. Součástí práce je sada programů pro zpracování a vyhodnocení dat a přehled a diskuze jejich výsledků na konkrétních datech. Výsledným nástrojem je možné zpracovávat data řádově v rozsahu stovek miliónů slov. Práce byla vypracována na českých morfologicky a syntakticky anotovaných datech, nicméně použité postupy nejsou na jazyce závislé.

Klíčová slova: sémantické sítě, automatické, vytváření, strojové učení

Title: Automatic construction of semantic networks

Author: Bc. Martin Kirschner

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Pavel Pecina Ph.D., pracoviště

Abstract: Presented work explores the possibilities of automatic construction and expansion of semantic networks with use of machine learning methods. The main focus is put on the feature retrieving procedure for the data set. The work presents a method of semantic relation retrieval, based on distributional hypothesis and trained on the data from Czech WordNet. We also show the first results for Czech language in this area of research. Part of the thesis is also a set of software for processing and evaluating of input data and a overview and discussion about its results on real-world data. The resulting tools can process data of amount in orders of hundreds of millions of words. The research part of the thesis used Czech morphologically and syntactically annotated data, but the methods are not language dependent.

Keywords: semantic networks, automatic, construction, machine learning

Obsah

1. Úvod

Definice. *Sémantickým zdrojem nazýváme v této práci obecně zdroj strukturovaných sémantických informací. Do této kategorie spadají sémantické slovníky, tezaury, sémantické sítě atd.*

V současné době dosahuje úroveň aplikací zpracování přirozeného jazyka takových výsledků, že další zlepšování je velmi obtížné. Mnohé úlohy NLP se potýkají čím dál více s problémem rozlišení různých významů jednoho slova, s jeho mnohoznačností. V tomto případě může využití kvalitního sémantického zdroje přinést viditelné zlepšení úspěšnosti. V oborech, jako strojový překlad, vyhledávání v textech a webových stránkách nebo strojové odpovídání otázek, lze údaje o definicích významu použít jako znalostní bázi, která přidá do aplikace umělé inteligence informace, potřebné k sémantickému zařazení slov, tedy k jistému druhu *„chápání“*. Dále je možné identifikaci významu využít například při strojovém překladu, hledání synonym, expanzi dotazů ve fulltextových vyhledávačích příbuznými slovy nebo jako referenci k sémantické rovině jazyka.

Jak již bylo zmíněno, přínosu pro úlohy zpracování přirozeného jazyka může sémantický zdroj dosáhnout pouze pokud je kvalitní. Kvalitou zde máme na mysli tato dvě kritéria.

1. **Dostatečný rozsah** — jednoznačně nutná podmínka pro širokou využitelnost sémantického zdroje. Zdroj musí pokrývat oblasti jazyka, pro které má být využit. Pokud doplňuje aplikaci pro vyhledávání zboží v e-shopech, musí pokrývat hlavně kategorie výrobků a produkty samotné. Očividně se jedná o jinou oblast dat, než například zdroj využívaný pro strojový překlad, kde je třeba pokrýt většinu aspektů běžného světa.

   Sémantický zdroj tedy musí být dostatečně rozsáhlý hlavně v oblasti, pro kterou je využíván. Z toho vyplývá, že nestačí pouze jeden globální sémantický zdroj, ale je potřeba více specifičtějších zdrojů pro různé oblasti využití, byť by měly stejné jádro obecných informací.

2. **Vysoká spolehlivost** — v přirozeném jazyce nejsou významy slov nikdy naprosto přesně definované, proto je obtížné je správně zařadit do sémantického zdroje s ohledem na jejich přesnou sémantiku. Navíc ani dva lidé často nevnímají přesný význam jednoho slova stejně, jak je ukázáno například právě v kapitole 4.2.1 při vyhodnocování výsledků ruční anotace relací. I kvůli tomu sebepečlivěji budovaný zdroj obsahuje určité procento nepřesností. Aby sémantický zdroj byl více přínosem, než zdrojem chyb, je nutné, aby toto procento chyb a nepřesností bylo co nejnižší.

1.1 Způsoby vytváření sémantických zdrojů

K vytváření sémantických zdrojů existují tři přístupy, jejichž výsledky se liší jak rozsahem, tak spolehlivostí. Jedním z dalších důležitých ukazatelů je i cena práce na konstrukci zdroje. Způsoby konstrukce s ohledem na charakteristiky výstupu jsou popsány níže.

1. **Ruční vytváření** — Konstrukci provádí anotátoři, kteří vybírají a zadávají údaje o významech slov do sémantického zdroje. Zdroje, vytvářené ručně, mají typicky nízké procento chybných relací, jejich nevýhodou ale bývá nízký rozsah. Ne vždy jsou do ručně budovaných sémantických zdrojů přidávána slova podle četnosti jejich užívání v běžném jazyce. Tím je způsobeno, že i zdroj obsahující relativně mnoho záznamů pokrývá jen malou část používaného jazyka.

   Příkladem je třeba Czech WordNet ve verzi 1.5 [29], který hustě pokrývá mimo jiné oblasti biologie, ale procento pokrytí používaného jazyka, reprezentovaného například korpusem PDT [14], se jeví horší, viz [6]. Další nevýhodou ručně vytvářených zdrojů je jejich vysoká cena. Na konstrukci se po dlouhou dobu musí podílet vyškolená skupina pracovníků, náklady tedy nejsou zanedbatelné.

   Příkladem ručně budovaného zdroje jsou ontologie *WordNet* [12] a *CyC* [20], nebo thesaury, například známý *Roget's Thesaurus* [17].

2. **Poloautomatické vytváření** — Částečným řešením problému vysoké ceny může být použití nástrojů, které anotátorovi nabízí slova, která by s určitou pravděpodobností mohla být v relaci s právě anotovaným konceptem. Tímto postupem je možné ušetřit čas a navíc zvýšit pokrytí slovníku, protože poloautomatické nástroje typicky jsou založené na korpusu jazyka. Metod využívaných k asistenci sémantické anotace existuje několik, většinou vznikají pro potřebu konstrukce konkrétních sémantických zdrojů, například MindNet[32] nebo pro PlWN [28].

3. **Automatická konstrukce** — Tento přístup se vyznačuje nízkou cenou práce v poměru k rozsahu slovníku. Automatická extrakce sémantických informací počítačem je prakticky bezplatná. Výhodou je také možnost výběru pokrytí oblastí jazyka výběrem korpusu, který bude automatickou metodou konstrukce zpracováván. Kladem tohoto přístupu jsou tedy dostatečný rozsah a nízká cena.

   Na druhou stranu kvalita (spolehlivost relací) výsledků prezentovaných prací je zatím nízká, proto tyto sémantické zdroje nebývají v dalších aplikacích používány. Proto jsou tyto programy využívány spíše jako pomoc při poloautomatické stavbě zdrojů.

1.2 Obsah práce

Tato práce prezentuje zřejmě první přístup, využívající strojové učení s učitelem, k plně automatické konstrukci sémantické sítě, založený na distribuční hypotéze. Metoda strojového učení je trénována na datech získaných ze strukturovaného, počítačem čitelného, sémantického zdroje. Dosažené výsledky jsou vyhodnocené jak automaticky, tak ručně a z porovnání těchto výsledků jsou vyvozeny závěry. Kromě automatické extrakce sémantických informací je možné postup využít i k poloautomatické konstrukci sémantických zdrojů. Práce sestává ze sady skriptů a následujících tří programů.

- **FeatureRetriever** — Nástroj pro získávání matic rysů ze vstupní matice pomocí aplikace sémantických metrik a filtrování.
- **EvaluateFeatures** — Program, který podporuje vyhodnocování úspěšnosti procesu získávání relací pomocí metody strojového učení, trénované na vstupních rysech a ukázkových vztazích ze sémantického zdroje.
- **WN–Transformer** — Program extrahující relace z WordNetu pro využití v programu EvaluateFeatures.

Teorii a zpracované řešení přibližuje text strukturovaný do sedmi kapitol, z nichž první je tento úvod a poslední obsahuje závěr práce.

Druhá kapitola popisuje existující metody automatického získávání sémantických informací a uvádí práce již v tomto odvětví prezentované.

Ve třetí a čtvrté kapitole je blíže popsaná metoda, která byla v této práci použita. Dále jsou zde analyzována použitá data a mezivýpočty. Na závěr čtvrté kapitoly jsou prezentovány dosažené výsledky.

Pátá a šestá kapitola popisují softwarové nástroje které jsou součástí této práce. Kapitola pět popisuje konfiguraci, datové formáty, způsob užití a funkcionalitu jednotlivých programů a skriptů. V této kapitole jsou také popsány použité externí nástroje. Šestá kapitola pak popisuje algoritmy a datové struktury, použité ve zmíněném software.

2. Metody získávání sémantických informací

Tato kapitola uvádí přehled existujících přístupů k automatické konstrukci sémantických sítí. V zásadě je možné tyto metody rozdělit do dvou skupin - získávání vztahů ze vzorců větné stavby a porovnání distribuce modelu kontextu zkoumaných slov. Oba přístupy jsou rozepsány v následujících podkapitolách, nejprve jsou zde ale definovány používané termíny.

Definice. *Lemma je obecně používaný termín pro základní tvar slova z hlediska morfologie. V této práci bude tento pojem využíván ve stejném významu.*

Postupy, použité v této práci, je možné využít i pro ostatní slovní druhy, v zájmu zjednodušení a zpřehlednění se ale budeme zabývat pouze podstatnými jmény.

Definice. *Sémantická relace, též označovaná jako sémantický vztah, zkráceně vztah, je binární relace mezi významy dvou lemmat.*

Neprovádíme desambiguaci, proto nejsou jednotlivé významy lemmat indexovány. Všechny významy lemmatu proto splývají do jednoho uzlu.

V práci jsou využívány relace hyponymie a hyperonymie, odpovídající obdobným relacím ve WordNetu [12], a relace synonymie, kterou lze z WordNetu také získat.

- **Hypo/hyperonymie** — Hyperonyma lemmatu jsou slova jemu významově nadřazená, hyponyma naopak slova podřazená.

  Například *ovoce* je hyperonymem slova *jablko* a hyponymem slova *plod*.

  Hyponyma i hyperonyma tvoří orientované hrany grafu, pro které platí:

  - **Směr nadřazenosti** — Hyperonymická relace mezi lemmaty **a** a **b** implikuje hyponymickou relaci mezi **b** a **a**.
  - **Tranzitivita** — Pro každé tři lemmata **a**, **b**, **c** platí, že pokud **a** je hyperonymem **b** a zároveň **b** je hyperonymem **c**, je i **a** hyperonymem **c**. Stejný vztah platí i pro hyponyma.

- **Synonymie** — Synonyma jsou slova stejného, nebo velmi blízkého, významu.

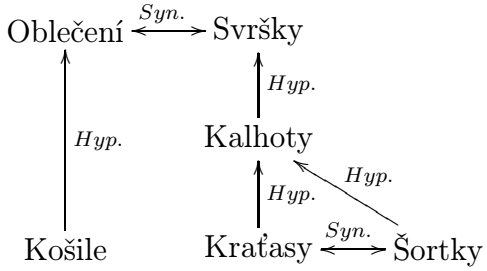
  Například *šálek* a *hrnek*, nebo *tvor* a *živočich*.

  Pro orientovanou relaci synonymie platí:

  - **Symetrie** — Synonymická relace mezi lemmaty **a** a **b** implikuje synonymickou relaci mezi **b** a **a**.
  - **Tranzitivita** — Pro každé tři různé uzly **a**, **b**, **c** platí, že pokud **a** je synonymem **b** a zároveň **b** je synonymem **c**, je i **a** synonymem **c**.

Definice. *Sémantická síť, zkráceně síť, je multigraf, jehož množinu vrcholů tvoří významy lemmat. Množina hran takového multigrafu je tvořena sémantickými relacemi.*

Obrázek 2.1 ukazuje příklad sémantické sítě s relacemi tvořenými synonymy a hyperonymy.



Obrázek 2.1: Schéma sémantické sítě

Definice. *Model kontextu lemmatu je množina slov, vyskytující se v jeho blízkosti v textu. Model kontextu může být definován přímým souseděním, maximální vzdáleností v počtu slov, větou, odstavcem nebo i dokumentem.*

Model kontextu může být na syntaktické rovině definován i vzorem vyskytujícím se ve větném stromě. Například mohou do modelu kontextu spadat slova v nadřazeném uzlu a v závislých uzlech větného stromu.

Definice. *Kontextový vektor je vektor četností slov, vyskytujících se ve zdrojových datech (typicky z korpusu) v modelu kontextu daného lemmatu.*

Definice. *Incidenční matice, také nazývaná matice kontextu, je matice jejímiž řádky jsou kontextové vektory zkoumaných lemmat. Popisy řádků incidenční matice jsou tedy zkoumaná lemmata a popisky sloupců všechny slova, vyskytující se v jejich modelech kontextu.*

Dále je v práci používáno označení *hodnota kontextu*. Tímto termínem označujeme hodnotu funkce vzdáleností lemmat, uloženou v modelu kontextu a poté i v incidenční matici.

2.1 Funkce sémantických zdrojů

Od sémantického zdroje je při jeho aplikacích vyžadována hlavně úloha co nejpřesněji definovat význam jednotlivých lemmat (viz Piasecki [28]). Tento úkol je řešen různými způsoby, více či méně vhodnými k užití v automatickém zpracování. V případě výkladových slovníků je definování významu dosaženo pomocí popisů a příkladů užití každého lemmatu. Tato forma je dobře čitelná člověkem, její strojové zpracování je ale netriviální.

Význam konkrétního lemmatu lze definovat také pomocí jeho vztahů k ostatním lemmatům. Případný slovní výpis této definice, pro potřeby člověka, je pak možné vygenerovat z těchto vztahů.

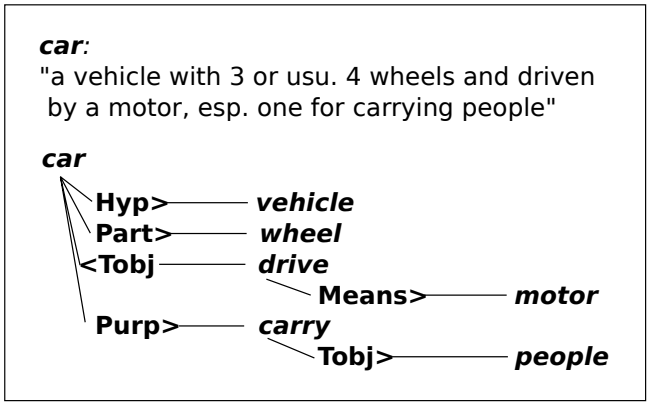
Význam lemmatu, určený relacemi, ve kterých se vyskytuje, je pro strojové zpracování srozumitelnější. Slovník takových definic pak je pouze seznamem relací mezi lemmaty. Pokud tyto relace splňují podmínku sémantické sítě, jak je definovaná výše, můžeme takový slovník za síť považovat.

Problémem u takové definice jsou jevy homonymie a polysémie, kdy jedno lemma má více různých významů. Při definici pomocí seznamu relací tak jednotlivé významy lemmatu splývají. Tento problém je možné řešit značkováním jednotlivých lemmat indexy jejich významů, tedy provedením desambiguace.

2.2 Získávání vztahů ze vzorců struktury věty

Jak bylo popsáno výše, konstrukce sémantické sítě odpovídá definování významů lemmat pomocí relací mezi nimi. Pokud již tyto významy jsou definované v nějakém jiném zdroji slovní formou, je možné z jejich stavby vět získat definici významu konceptu sémantickými relacemi.

Na obrázku 2.2 je vidět typická stavba věty glosy thesauru. Sémantické pársování generuje kvalitní relace, nicméně elektronické výkladové slovníky jsou dostupné jen pro málo jazyků a trpí stejnými nevýhodami, jako ručně budované sémantické sítě. Příkladem takto budovaného zdroje je MindNet [30].



Obrázek 2.2: Sémantické pársování glosy v MindNetu [30]

Podobným způsobem je možné získávat relace i z jiných zdrojů, například z elektronických encyklopedií a korpusů. I u nich lze hledat určité vzorce ve větné stavbě, jak dokazuje například Hearst [16].

Lingvistika je věda studující přirozený jazyk.
$\Longrightarrow$ *hyp(lingvistika, věda)*

Obrázek 2.3: Typická věta zpracovávaná z nestrukturovaných zdrojů

Obrázek 2.3 ukazuje typ věty, kterou je možné v těchto zdrojích nalézt. Při použití těchto zdrojů je ale procento chyb vyšší. Ne všechny věty odpovídající takovému vzorci mají obdobný smysl. Například metafory nebo jiné básnické obraty zanášejí do výsledků nechtěné relace. K eliminaci těchto chyb by bylo nutné přidat ještě analýzu pragmatické roviny [28], což je úkol, jehož obstojné řešení zatím nebylo představeno.

Výsledky extrakce vztahů ze vzorců struktury věty jsou velmi ovlivněné jednotlivými výskyty, postrádají tak stabilitu, která je při budování stabilního sémantického zdroje nezbytná.

2.3 Porovnávání distribuce modelu kontextu

Předchozí přístup spoléhal na nalezení přímo konkrétních výskytů definic ve zdroji dat. Myšlenka porovnávání distribuce modelů kontextu je naproti tomu založená na celkové charakteristice výskytů lemmat v datech.

Manifestem tohoto přístupu je distribuční hypotéza, kterou formuluje Harris [15]. Distribuční hypotéza říká, že existuje přímý vztah mezi pozorovanými užitími jazykové entity a jeho významem. Pokud za entitu z této hypotézy dosadíme lemma, odpovídají jeho požadovaná užití modelům kontextu, ve kterých se v korpusu vyskytuje. Po sečtení kontextových hodnot jednotlivých slov ze všech modelů kontextu lemmatu získáme distribuci jeho modelu kontextu. Podle distribuční hypotézy má distribuce modelu kontextu přímý vztah k významu jednotlivých lemmat. Porovnáním distribucí modelů kontextu dvou lemmat metrikou podobnosti získáme údaj o blízkosti jejich významů, viz příklad výpočtu kontextové metriky *cosinus*, který je zobrazený na obrázku 2.4.

Zdrojové věty:

- *V této zatáčce závodní vozy podřadí a jedou pomaleji.*
- *Před předjížděním si v autě podřaď, potom jeď na plný plyn.*

	zatáčka	závodní	podřadit	vůz	jet	pomalý	předjíždění
vůz	1	1	1	0	1	1	0
auto	0	0	1	0	1	0	1
zatáčka	0	1	1	1	0	0	0

$$Cos(\vec{x},\vec{y}) = \frac{\sum_{k=1}^{n} x_k \cdot y_k}{\sqrt{\sum_{k=1}^{n} {x_k}^2 \cdot \sum_{k=1}^{n} {y_k}^2}}$$

	vůz	auto	zatáčka
vůz	1	**0,5164**	0,3162
auto	**0,5164**	1	0,3333
zatáčka	0,3162	0,3333	1

Obrázek 2.4: Využití distribuční hypotézy k získání podobných konceptů.
K počítání podobnosti je zde použita metrika cosinus

Jak je ze zmiňovaného příkladu vidět, v praxi se využívá incidenční matice získaná z modelu kontextu, jak to aplikuje například Schütze v [31]. Pro získávání modelu kontextu a operace s maticí existuje mnoho přístupů, z nichž některé využívá například projekt SuperMatrix [7]. Metody využité v této práci jsou popsány v následujících kapitolách.

Metoda, využití distribuční hypotézy, není, na rozdíl od postupu popsaného v předchozí sekci, citlivá na nestandardní užití slov, například v již zmiňovaných metaforách. Nevýhodou je, že výstupem jsou pouze výsledky metrik, ne konkrétní sémantický vztah. Proto distribuční přístup hojněji používají spíše práce zaměřené například na určení *similarity* (sémantická podobnost) a *relatedness* (existence sémantického vztahu), zmíněné v následující sekci.

2.4 Další blízké úkoly počítačového zpracování přirozeného jazyka

Z ostatních oborů komputační sémantiky, blízkých problému extrakce sémantických relací, je možné využít ověřené postupy a metriky pro práci s daty.

Mezi úlohy příbuzné extrakci sémantických relací a konstrukci sémantické sítě patří rozpoznávání sémantických kolokací a měření *similarity* a *relatedness*.

2.4.1 Rozpoznávání sémantických kolokací

Rozpoznávání sémantických kolokací je úloha, kdy se pro dvojici slov na základě údajů získaných z korpusu rozhodne, zda společně mají jiný význam, než pouze kombinace významů obou slov.

U řešení úlohy kolokací, jak jej realizuje Pecina [26] je možné se inspirovat jak různými výskytovými metrikami, tak výsledným způsobem rozhodování. Ve zmíněné práci je využito několik desítek metrik, pracujících kromě jiného s frekvencemi slov ve dvojici a frekvencí jejich souvýskytů. Výsledky těchto metrik jsou pak předávány jakožto rysy lineárnímu klasifikátoru, který rozhoduje, jestli se jedná o sémantickou kolokaci, či nikoliv. Více viz například citovaná Pecinova práce.

2.4.2 Similarity a relatedness

Určení *similarity* a *relatedness* dvojice slov, jak je definují Budanitsky a Hirst [8], je velmi blízce podobnou úlohou, jako extrakce sémantických relací. Vysoká hodnota *similarity* říká, že jsou si porovnávaná slova významově velmi podobná a blíží se synonymům. Vysoká hodnota *relatedness* zase ukazuje, že z hlediska člověka patří obě slova do stejného oboru a může mezi nimi existovat sémantický vztah.

Ve velkém počtu případů sice mezi slovy s vysokou *relatedness* vztah není, bývají však v sémantické síti blízko (měřeno počtem hran nejkratší neorientované cesty). Často například sourozenci ve stromě hyperonym.

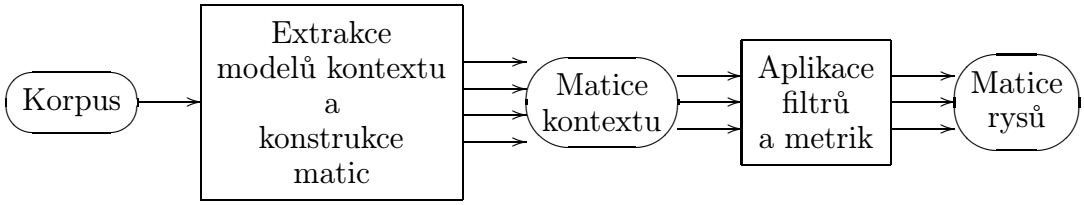
Inspirativní prací v tomto oboru byl například článek Agirre et al. [4], který studuje zjišťování *similarity* a *relatedness* s využitím distribucí modelu kontextu na velkém korpusu. Ve zmiňovaném článku jsou mimo jiné popisovány postupy získávání distribucí modelu kontextu, které jsou relevantní i pro tuto práci. Uvedené metody jsou ale laděny pro angličtinu.

3. Konstrukce trénovacích a testovacích dat

Metoda získávání sémantických rysů, prezentovaná v této práci, využívá metod strojového učení trénovaných na rysech vypočítávaných z incidenční matice modelů kontextu slov získané z rozsáhlého korpusu. Jedná se o jednu z prvních aplikací metod strojového učení s učitelem k řešení tohoto problému.

Metoda je robustní – její výsledek nezávisí na jednotlivých výskytech lemmat, ale na celkových distribucích jejich modelů kontextu.

Trénovací data jsou sestavena na základě relací, obsažených v Czech WordNetu (CWN). Nejprve definujeme používané termíny, a poté v následujících sekcích rozebereme jednotlivé kroky metody, znázorněné na obrázku 3.1.



Obrázek 3.1: Postup získávání rysů pro metodu strojového učení

Definice. *Matice rysu je v našem případě matice, která má jako popisky řádků i sloupců lemmata a v políčkách obsahuje hodnotu daného rysu pro dvojici lemmat v řádku a sloupci.*

Definice. *Datasetem označujeme matici, která je po řádcích tvořena instancemi dat pro strojové učení. Tato instance je tvořena vektorem rysů a cílovou třídou.*

Definice. *Doména rysu je termín, kterým označujeme množinu všech lemmat, pro jejíž kartézský součin (všechny dvojice) je hodnota rysu k dispozici.*

V praxi se tato množina rovná množině popisek řádků matice rysu. Označujme doménu rysu r jako $Dom_f(r)$

Definice. *Doména sémantického zdroje je termín, kterým označujeme množinu všech lemmat, která se v sémantickém zdroji vyskytují.*

V případě WordNetu se tak jedná o všechna různá lemmata ve všech synsetech. Označujme doménu sémantického zdroje S jako $Dom_{ss}(S)$.

3.1 Zdroje dat

Metody, jejichž výsledky jsou založené na distribuci modelu kontextu, jsou velmi závislé na kvalitě dat (ve smyslu rozsahu a pokrytí). Z tohoto důvodu je jednou z priorit této práce, aby software, který byl vytvořen jako její součást, byl schopen operovat s velkými objemy dat. Tato práce si navíc dává za cíl aplikování zvolené metodiky na co největší dostupný rozsah textů.

Jakožto vstup prezentované metody může být použit i prostý text bez předchozích úprav, ze kterého jsou získávány frekvence slov. Při zpracovávání jazyků s bohatou morfologií, mezi něž patří i čeština, je ale problém s *ředěním* dat, kdy jsou slova zastoupena v textu v mnoha formách, lišících se inflexí. Tento problém lze řešit pomocí lemmatizace, kdy jsou slova převedena na základní tvar. Pro každé lemma je tak jeho hodnota modelu kontextu získána sečtením hodnot modelů kontextu všech jeho tvarů.

Dalším zvýšením kvality dat je jejich syntaktická anotace do závislostních stromů. Model kontextu lemmatu získaný z takto připravených dat, například zahrnutím všech závislých uzlů stromu, je s zřejmě relevantnější, než model kontextu, získaný z povrchové struktury věty.

Data byla extrahována z Pražského závislostního korpusu (PDT), který je tvořen texty z českých novin z 90. let a z Českého národního korpusu (ČNK).

1. **Pražský závislostní korpus** je rozdělený na soubory dat podle úrovně anotace [14]. Sady jsou ručně anotované a obsahují morfologickou, syntaktickou i hloubkovou (tektogramatickou) úroveň. Každá úroveň anotace zároveň obsahuje i anotace nižších úrovní. Proto pro zisk modelu kontextu ze syntaktických stromů můžeme použít poslední dvě zmiňované sady a pro ostatní metody extrakce modelu kontextu sady všechny.

2. **Český národní korpus (ČNK)** je tvořen texty z českých novin, publicistických textů a beletrie, strojově anotovanými na morfologickou úroveň [27]. K získávání modelu kontextu z morfologické roviny byla využita tři referenční vydání – *syn2000* [1], *syn2005* [2] a *syn2006pub* [3].

   Korpus *syn2000* je stejně jako *syn2005* žánrově vyvážené vydání, lišící se hlavně léty vydání zdrojových textů. Verze *syn2000* obsahuje převážně texty z let 1990 – 1999 a verze *syn2005* obsahuje texty pokrývající období 2000 – 2004. Třetí použité vydání korpusu, *syn2006pub*, je souborem publicistických textů z let 1989 – 2004.

Objem dat získaný z jednotlivých zdrojů přehledně znázorňuje tabulka 3.1.

Zdroj	Celkový počet slov
ČNK syn2000	100 mil.
ČNK syn2005	100 mil.
ČNK syn2006pub	300 mil
PDT 2.0 m	450 tis.
PDT 2.0 a *	670 tis.
PDT 2.0 t *	830 tis.
Celkem *	1,5 mil.
Celkem	502 mil.

Tabulka 3.1: Velikosti jednotlivých zdrojů dat.

Zdroje označené hvězdičkou obsahují věty anotované na syntaktickou úroveň.

3.2 Získávání modelu kontextu

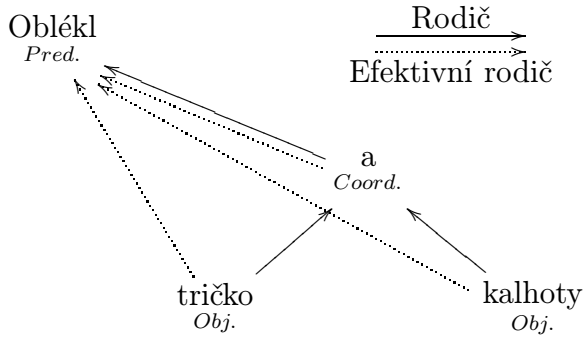
Ve fázi extrakce modelu kontextu jsou zpracovávána textová data do formy čtveřic *(první lemma; vztah; druhé lemma; hodnota modelu kontextu)*, ze kterých je v další fázi sestavená incidenční matice. Sloupce této matice jsou označené spojením vztahu ze čtveřice a druhého lemmatu. Tento způsob konstrukce incidenční matice je využívá v případě modelu kontextu, získaného ze syntakticky označkovaných dat. V tom případě je vztahem ve čtveřici syntaktický vztah obou lemmat. V případě ostatních modelů kontextu je za vztah dosazena konstanta, nehraje tedy roli. Řádky jsou označené prvním lemmatem. Do matice se zapisuje součet všech hodnot modelu kontextu (nejčastěji počet výskytu) dvojice lemmat a vztahu.

Následující sekce popisují čtyři použité metody získávání modelu kontextu. Z každého ze vzniklých seznamů čtveřic byla zkonstruována samostatná matice kontextu.

3.2.1 Získávání modelu kontextu ze syntakticky anotovaných dat

Využití závislostních vztahů ze syntaktických stromů dává datům lepší výpovědní hodnotu. Zatímco načtením modelu kontextu povrchového získáme množinu typických sousedů lemmatu, načtením syntaktického modelu kontextu získáme množinu slov, která typicky s lemmatem opravdu mají nějaký vztah. Tato konfigurace se přibližuje definici významu jak je popsána v sekci 2.1, můžeme tedy očekávat lepší výsledky metrik měřících podobnost.

Pro každé slovo je do modelu kontextu zahrnut jeho *efektivní rodič* a *efektivní potomci*, pokud existují, společně s typem jejich vztahu a směrem závislosti. Pojmy *efektivní rodič* a *efektivní potomek* zde označují nejbližší uzly závislostního stromu (co do počtu hran vzhůru, respektive dolů) na každé orientované cestě vedoucí do (resp. z) probíraného uzlu, které mají jinou než pomocnou, nebo koordinující funkci. Přehledně význam těchto pojmů ukazuje obrázek 3.2. Matice získané touto metodou budeme označovat prefixem *a*.



Obrázek 3.2: Efektivní rodič, efektivní potomek.

3.2.2 Získávání modelu kontextu z morfologicky anotovaných dat

Většina dat z nestrukturovaných zdrojů není na syntaktickou úroveň anotovaná, proto je nutné implementovat i techniky získávání modelu kontextu z nižších vrstev anotace. Zdroje nemusí být anotované ani na morfologickou úroveň, to však lze s velmi vysokou úspěšností (na rozdíl od strojového syntaktického značkování) provést automaticky a v relativně (vzhledem k velikosti dat) krátké době. Není tedy třeba získávat model kontextu přímo z neanotovaných dat, bez možnosti využití lemmatizace a filtrů podle slovních druhů. Automatická anotace na syntaktickou úroveň je ale náročnější a méně spolehlivá, proto je nutné aplikovat i získávání modelu kontextu z dat označkovaných pouze morfologicky.

Pro zisk modelu kontextu slova z věty se nejčastěji používají tři postupy, rozebrané v následujících podsekcích.

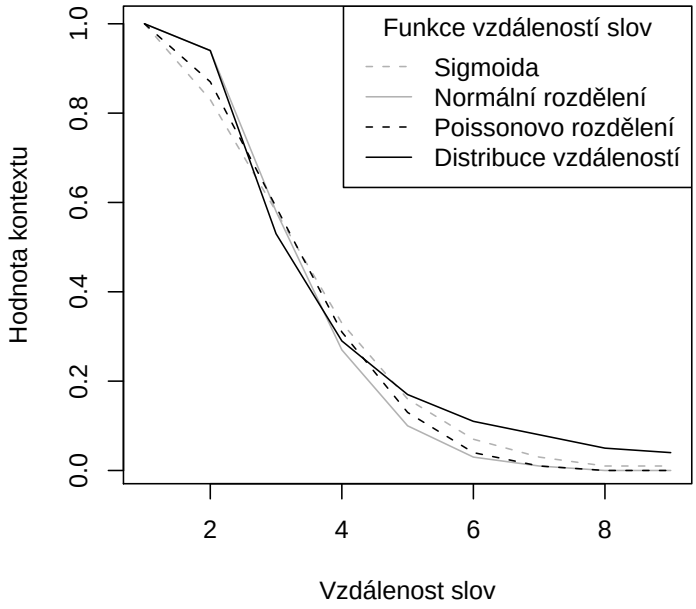
Slova z celé věty jako model kontextu

Do modelu kontextu lemmatu jsou zde započítána všechna slova ve větě. To znamená, že každé lemma je s každým slovem v rámci věty vzájemně v modelu kontextu.

Při aplikování této metody jsou získávány násobně vyšší hodnoty v polích incidenční matice, než u následujících způsobů. Postihem za extrakci všech slov, která s konceptem opravdu souvisí, je více šumu (vyšších počtů u nesouvisejících slov) v incidenční matici. Čím větší je ale rozsah dat, tím je poměr hodnot modelu kontextu správných a nesouvisejících slov lepší, což je od určité úrovně možné filtrovat. Toto *vytříbení* pomocí nasčítání dat je způsobeno tím, že velikost modelu kontextu, definující význam konceptu (tedy počet souvisejících slov), je v poměru k celkovému počtu slov ve zdroji dat velmi nízká, na druhou stranu jeho výskyt v blízkosti konceptu je častější. Matice získané touto metodou budeme označovat prefixem *ms0*.

Okénko určité velikosti

Model kontextu je zde získáván z bezprostředního okolí lemmatu. Do modelu kontextu jsou zahrnuta slova, jejichž vzdálenost od lemmatu v rámci věty je shora limitována určitým číslem.



Obrázek 3.3: Distribuce vzdáleností dvojic lemmat v syntaktickém vztahu.

V našem případě byla přidávána slova se vzdáleností od lemmatu menší než čtyři. Toto číslo bylo zvoleno po prozkoumání distribuce vzdáleností dvojic slov, získaných ze syntakticky označkovaných dat s tím, že vzdálenější slova mají s lemmatem vztah v příliš nízkém procentu případů. Distribuci vzdáleností dvojic lemmat v syntakticky označkovaných datech ukazuje obrázek 3.3. Tato metoda získávání modelu kontextu byla inspirována prací Agirre et al. [4], kde je zkoumána pod názvem *bag of words*. Matice získané touto metodou budeme označovat předponou *mw*.

Celá věta jako model kontextu s hodnotou závislou na vzdálenosti

Tato metoda se, stejně jako předchozí, snaží o aproximaci distribuce vzdáleností. Jako model kontextu lemmatu je zde brána celá věta, ale jakožto hodnota modelu kontextu je brán výsledek funkce vzdálenosti slov od lemmatu, viz obrázek 3.3. Použitá funkce transformující vzdálenost je volena tak, aby co nejvíce korelovala s distribucí vzdáleností dvojic lemmat v syntakticky označkovaných datech. Jako kandidáti byly zkoušeny parametrizované funkce *sigmoidní*, distribuční funkce *normálního* rozdělení a distribuční funkce *Poissonova* rozdělení. Tabulka 3.2 uvádí odmocněnou střední kvadratickou chybu (RMSD) testovaných funkcí. Funkce s nejmenší odchylkou od distribuce vzdáleností syntakticky vztažených slov byla distribuční funkce *Poissonova* rozdělení (viz obrázek 3.3), proto byla využita pro výpočet hodnoty modelu kontextu dvojice slov v závislosti na jejich vzdálenosti. Matice získané touto metodou budeme označovat prefixem *ms1*.

Funkce	RMSD
Sigmoida	0.04174
Distribuční funkce normálního rozdělení	0.04246
Distribuční funkce Poissonova rozdělení	**0.04152**

Tabulka 3.2: Aproximace distribuce vzdáleností dvojic v synt. mod. kontextu.

Odmocněná střední kvadratická chyba, hojně využívaná k ukázání rozdílů mezi dvěma vektory hodnot, se počítá následujícím vzorcem.

$$RMSD(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{n}}$$

3.3 Metody filtrování a vyhlazování

Jak již bylo zmíněno, hromadně získaná data obsahují určité procento *šumu*, který zhoršuje výsledky. Pro jeho odstranění, nebo alespoň zmírnění, existuje opět mnoho metod, z nichž ty využité budou zmíněny v následujících podsekcích. Jedná se nejčastěji o různé druhy filtrů, které část šumu odstraňují.

3.3.1 Lokální filtrování

Na začátku, ještě před konstrukcí matice kontextu, je možné odfiltrovat slova, která nenesou žádný význam (spojky, předložky, částice, zájmena apod.) a nespecifická slova, tedy slova která se vyskytují ve většině modelů kontextu a proto mají minimální rozlišovací sílu (slovesa být, mít apod.). K rozpoznání těchto slov není potřeba znát globální počty slov v modelech kontextu, proto tento druh filtru nazýváme lokálním. Odstraněním lemmat, vyhovujících seznamu s nežádoucími slovními druhy, nebo seznamu s nespecifickými slovy, z lemmat, pro které chceme získávat rysy, získáme základní množinu lemmat pro konstrukci matice kontextu.

Seznam zakázaných slov může být využit také pro filtrování obecnějších vzorů, například vyřazení slov obsahujících číslice, nebo slov začínajících velkým písmenem. Seznamy slov a slovních druhů k tomuto filtru se nacházejí na přiloženém CD v adresáři */software/lists*.

3.3.2 Globální filtrování

Matice vzniklé lokálním filtrováním obsahují stále ještě šum, například v podobě náhodných výskytů ve vzájemném modelu kontextu, nebo další nespecifická slova, nezachycená při lokálním filtrování. V těchto případech lze aplikovat globální filtry, které omezí minimální a maximální frekvence slova, minimální počet nenulových hodnot ve sloupci nebo řádku matice kontextu nebo jeho minimální entropii.

Odstranění řádků matice je provázáno se změnou filtrovaných hodnot sloupců a naopak, proto je nutné proces filtrování opakovat v několika iteracích, dokud nebudou všem podmínkám globálních filtrů vyhovovat jak řádky, tak sloupce matice.

Jednotlivé filtry byly nastaveny tak, aby při filtrování minimální frekvencí nebo minimální entropií odstranily okolo 4% celkového součtu hodnot modelu kontextu v první iteraci. Hodnota 4% je zvolena s ohledem na dobu trvání dalších transformací výsledné matice, tedy na její velikost.

Filtrování maximálním počtem (aplikované pouze na sloupce, v řádcích nám častá slova nevadí) bylo nastaveno tak, aby bylo odstraněno přibližně 40% celkového součtu hodnot modelu kontextu, což je přibližně objem, který bývá odstraněn zavedenými seznamy nežádoucích slov pro angličtinu, viz Chandna [9].

Hodnota minimálního počtu nenulových prvků vektorů byla nastavena na polovinu minimální frekvence slova vektoru příslušného. Konkrétní použité hodnoty filtrů lze nalézt v konfiguračních souborech v adresáři */software/cfg/* na CD.

Matice vzniklé aplikací filtrů jsou označeny názvem jejich konfiguračních souborů před přeponou přidaných za označení zdrojové matice (např. *a_ffa* nebo *ms0_ffms0*).

3.3.3 Booleanizace

Booleanizace je druh filtru, který hodnoty v matici nižší než určitý práh nahradí nulou a ostatní změní na jedna. Tento postup provádí opět určitý druh vyhlazování, metody strojového učení díky tomu získají jiný pohled na data.

Při Booleanizaci byla jako prahová hodnota zvolena hodnota 1 – Hodnoty menší než 1 byly vynulovány a větší sníženy na 1. Důvodem této volby byl záměr více neředit již tak řídké (z hlediska počtu nenulových hodnot) vektory kontextu.

Matice, které prošly Booleanizací, mají ke svému názvu přes podtržítko přidané označení *bool* (např. *a_ffa_bool*).

3.3.4 Výsledné zdrojové matice

Po aplikaci metod extrakce modelu kontextu a filtrování matic na vstupní data z korpusů bylo vytvořeno celkem osm matic. První čtyři vznikly ze vstupních dat aplikací popsaných metod získávání modelu kontextu a lokálním a globálním filtrováním. Další čtyři matice vznikly jejich Booleanizací. Tabulka 3.3 uvádí rozměry čtyř zdrojových matic.

Matice	Počet řádků	Počet sloupců
a_ffa	7 563	46 776
mw4_ffmw4	11 216	30 609
ms0_ffms0	12 232	31 769
ms1_ffms1	11 454	30 821

Tabulka 3.3: Rozměry zdrojových matic.

3.3.5 Scaling

Scaling, česky také škálování, je transformace již vypočtených matic rysů, aby se všechny hodnoty nacházely v určeném rozmezí. Tento postup nemění informační hodnotu rysů, používá se pro zjednodušení práce metody strojového učení, která rozpoznává relace.

V této práci je na všechny vypočtené rysy před jejich kompilací do datasetu použitý *scaling* lineární, který pouze transformuje hodnoty odečtením minimální hodnoty a podělením rozdílu minimální a maximální hodnoty. Dodaný software dále podporuje ještě škálování sigmoidní.

K názvu matic, které prošly škálováním, byl přidán identifikátor *scal* (např. *a_ffa_bool_cos_scal*).

3.4 Získávání rysů

V této sekci jsou popsány metody získávání informací z modelu kontextu, v podobě metrik na něj aplikovaných. Záměrem této práce je prozkoumat jejich schopnost popsat sémantické vztahy modelem kontextu vyjádřené a pomocí hodnot metrikami vypočtených rozpoznat typ relace. K abstrakci kombinace různých charakteristik modelu kontextu je v další kapitole použito metod strojového učení, trénovaných na již existující sadě sémantických relací. Tento postup je pak možný použít při rozšiřování zdrojové množiny sémantických relací na obory slov, které ještě nejsou sémantickým zdrojem pokryté.

3.4.1 Míry hodnotící výskyty a souvýskyty slov

Nejdůležitější míry dvojic slov, ať už z hlediska sémantiky, informační teorie nebo statistiky, používané v pracích s podobnou tématikou, jako je tato jsou popsány níže. Práce, které je uvádějí, jsou například Budanitsky a Hirst [8] nebo Chandna [9].

Matice rysu vzniklá aplikací metriky na všechny dvojice řádků vstupní matice je nazvána kombinací názvu vstupní matice a zkratky názvu použité metody. Například matice rysu *a_ffa_bool_cos* vznikla aplikováním metriky *cosinus* na matici *a_ffa_bool*.

1. **Cosine** — Nejznámější míra podobnosti vektorů počítá *cosinus* úhlu, který svírají. Na vektory kontextu $\vec{x}$ a $\vec{y}$ je aplikován tento vzorec:

$$Cos(\vec{x}, \vec{y}) = \frac{\sum_{k=1}^{n} x_k \cdot y_k}{\sqrt{\sum_{k=1}^{n} {x_k}^2 \cdot \sum_{k=1}^{n} {y_k}^2}}$$

2. **Dice** — Míra, která, obdobně jako F-measure [22], kombinuje oba vektory. Parametr α určuje, který z obou vektorů má mít větší váhu. V našem případě byl parametr α ponechán na výchozí hodnotě 0,5 a oba vektory tak měly stejnou váhu.

$$Dice(\vec{x}, \vec{y}) = \frac{\sum_{k=1}^{n} x_k \cdot y_k}{\alpha \cdot \sum_{k=1}^{n} {x_k}^2 + (1-\alpha) \cdot \sum_{k=1}^{n} {y_k}^2}$$

3. **Jaccard** — Jaccardova míra počítá poměr velikosti průniku modelů kontextu ku velikosti jejich sjednocení. Pro vektory kontextu $\vec{x}$ a $\vec{y}$ vypadá výsledek takto:

$$Jaccard(\vec{x}, \vec{y}) = \frac{\sum_{k=1}^{n} x_k \cdot y_k}{\sum_{k=1}^{n} {x_k}^2 + \sum_{k=1}^{n} {y_k}^2 - \sum_{k=1}^{n} x_k \cdot y_k}$$

4. **Lin** — Metrika, kterou publikoval Lin [21] je založená na zpracování pravděpodobnosti výskytu dvojice slov v určité závislostní relaci. To lze za cenu menší informační hodnoty zobecnit na libovolnou relaci, například na tu, kterou dostáváme při tvorbě seznamů kontextových čtveřic. Popis výpočtu hodnoty této míry je dále rozepsán v citované práci.

5. **Distance** — Jednoduchá míra, která vrací vzdálenost vektorů dvou konceptů. V práci je použita Euklidovská vzdálenost, která se pro vektory $\vec{x}$ a $\vec{y}$ počítá způsobem, popsaným vzorcem níže. Tuto míru zavádí například projekt SuperMatrix [7].

$$EuclidDist(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

6. **Coverage** — Pokrytí je jediná použitá nesymetrická míra. Jejím základem je idea, že čím větší je model kontextu daného lemmatu, tím větší jsou možnosti jeho použití. Dále počítá s tím, že čím větší jsou možnosti použití lemmatu, tím je lemma obecnější. Informační přínos této metriky je potvrzen v sekci 4.1.4. Výpočet pokrytí pro vektory $\vec{x}$ a $\vec{y}$ ukazuje následující vzorec:

$$Coverage(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sum_{i=1}^{n} y_i}$$

7. **Spearman** — Spearmanův korelační koeficient porovnává podobnost dvou pořadí. Zdrojové vektory je tak nutné nejprve konvertovat do vektorů, kde je každý zdrojový prvek nahrazen pořadím své hodnoty vzhledem k hodnotám ostatních prvků. Díky tomuto postupu tak nejsou důležité samotné hodnoty jednotlivých prvků, ale pořadí jejich hodnot. Jedná se tedy o další způsob vyhlazování.

   Spearmanův koeficient ρ pro vektory pořadí $\vec{x_{rank}}$ a $\vec{y_{rank}}$ se počítá podle vzorce níže [34].

$$\rho(\vec{x_{rank}}, \vec{y_{rank}}) = \frac{\sum_{i=1}^{n} (x_{rank_i} - \overline{x_{rank}}) * (y_{rank_i} - \overline{y_{rank}})}{\sqrt{(x_{rank_i} - \overline{x_{rank}})^2 * (y_{rank_i} - \overline{y_{rank}})^2}}$$

Č.	Bool.	1. krok	2. krok	Typ	Název
1	Ano	Cosine	Cosine	sym.	a_ffa_bool_cos_cos_scal.xsymm
2	Ano	Cosine	-	sym.	a_ffa_bool_cos_scal.xsymm
3	Ano	Coverage	Cosine	sym.	a_ffa_bool_cov_cos_scal.xsymm
4	Ano	Coverage	-	nesym.	a_ffa_bool_cov_scal.xdense
5	Ano	Dice	Cosine	sym.	a_ffa_bool_dice_cos_scal.xsymm
6	Ano	Dice	-	sym.	a_ffa_bool_dice_scal.xsymm
7	Ne	Cosine	Cosine	sym.	a_ffa_cos_cos_scal.xsymm
8	Ne	Cosine	-	sym.	a_ffa_cos_scal.xsymm
9	Ne	Dice	Cosine	sym.	a_ffa_dice_cos_scal.xsymm
10	Ne	Dice	-	sym.	a_ffa_dice_scal.xsymm
11	Ne	Distance	Cosine	sym.	a_ffa_dist_cos_scal.xsymm
12	Ne	Distance	-	sym.	a_ffa_dist_scal.xsymm
13	Ne	Jaccard	Cosine	sym.	a_ffa_jac_cos_scal.xsymm
14	Ne	Jaccard	-	sym.	a_ffa_jac_scal.xsymm
15	Ne	Lin	Cosine	sym.	a_ffa_lin_cos_scal.xsymm
16	Ne	Lin	-	sym.	a_ffa_lin_scal.xsymm
17	Ne	Spearman	Cosine	sym.	a_ffa_spear_cos_scal.xsymm
18	Ne	Spearman	-	sym.	a_ffa_spear_scal.xsymm

Tabulka 3.4: Rysy získané ze syntaktického modelu kontextu.

Č.	Bool.	1. krok	2. krok	Typ	Název
19	Ano	Cosine	-	sym.	ms0_ffms0_bool_cos_scal.xsymm
20	Ano	Coverage	-	sym.	ms0_ffms0_bool_cov_scal.xdense
21	Ano	Dice	-	sym.	ms0_ffms0_bool_dice_scal.xsymm
22	Ne	Cosine	-	sym.	ms0_ffms0_cos_scal.xsymm
23	Ne	Dice	-	sym.	ms0_ffms0_dice_scal.xsymm
24	Ne	Distance	-	sym.	ms0_ffms0_dist_scal.xsymm
25	Ne	Jaccard	-	sym.	ms0_ffms0_jac_scal.xsymm
26	Ne	Lin	-	sym.	ms0_ffms0_lin_scal.xsymm
27	Ne	Spearman	-	sym.	ms0_ffms0_spear_scal.xsymm

Tabulka 3.5: Rysy získané z celé věty jako model kontextu.

Č.	Bool.	1. krok	2. krok	Typ	Název
28	Ano	Cosine	-	sym.	ms1_ffms1_bool_cos_scal.xsymm
29	Ano	Coverage	Cosine	sym.	ms1_ffms1_bool_cov_cos_scal.xsymm
30	Ano	Coverage	-	nesym.	ms1_ffms1_bool_cov_scal.xdense
31	Ano	Dice	Cosine	sym.	ms1_ffms1_bool_dice_cos_scal.xsymm
32	Ano	Dice	-	sym.	ms1_ffms1_bool_dice_scal.xsymm
33	Ne	Cosine	Cosine	sym.	ms1_ffms1_cos_cos_scal.xsymm
34	Ne	Cosine	-	sym.	ms1_ffms1_cos_scal.xsymm
35	Ne	Dice	-	sym.	ms1_ffms1_dice_scal.xsymm
36	Ne	Distance	Cosine	sym.	ms1_ffms1_dist_cos_scal.xsymm
37	Ne	Distance	-	sym.	ms1_ffms1_dist_scal.xsymm
38	Ne	Jaccard	-	sym.	ms1_ffms1_jac_scal.xsymm
39	Ne	Lin	-	sym.	ms1_ffms1_lin_scal.xsymm
40	Ne	Spearman	-	sym.	ms1_ffms1_spear_scal.xsymm

Tabulka 3.6: Rysy získané z funkce vzdáleností v celé větě.

Č.	Bool.	1. krok	2. krok	Typ	Název
41	Ano	Cosine	Cosine	sym.	mw4_ffmw4_bool_cos_cos_scal.xsymm
42	Ano	Cosine	-	sym.	mw4_ffmw4_bool_cos_scal.xsymm
43	Ano	Coverage	Cosine	sym.	mw4_ffmw4_bool_cov_cos_scal.xsymm
44	Ano	Coverage	-	sym.	mw4_ffmw4_bool_cov_scal.xsymm
45	Ano	Dice	Cosine	sym.	mw4_ffmw4_bool_dice_cos_scal.xsymm
46	Ano	Dice	-	sym.	mw4_ffmw4_bool_dice_scal.xsymm
47	Ne	Cosine	-	sym.	mw4_ffmw4_cos_scal.xsymm
48	Ne	Dice	Cosine	sym.	mw4_ffmw4_dice_cos_scal.xsymm
49	Ne	Dice	-	sym.	mw4_ffmw4_dice_scal.xsymm
50	Ne	Distance	Cosine	sym.	mw4_ffmw4_dist_cos_scal.xsymm
51	Ne	Distance	-	nesym.	mw4_ffmw4_dist_scal.xdense
52	Ne	Jaccard	Cosine	sym.	mw4_ffmw4_jac_cos_scal.xsymm
53	Ne	Jaccard	-	sym.	mw4_ffmw4_jac_scal.xsymm
54	Ne	Lin	Cosine	sym.	mw4_ffmw4_lin_cos_scal.xsymm
55	Ne	Lin	-	sym.	mw4_ffmw4_lin_scal.xsymm
56	Ne	Spearman	Cosine	sym.	mw4_ffmw4_spear_cos_scal.xsymm
57	Ne	Spearman	-	sym.	mw4_ffmw4_spear_scal.xsymm

Tabulka 3.7: Rysy získané metodou okénka urč. velikosti.

3.4.2 Použité transformace

Z předchozích postupů jsme získali osm různých základních matic. Na každou z nich bylo aplikováno několik transformací, včetně koncového *scalingu*.

Z důvodu technických omezení nebyly některé z plánovaných matic rysů dopočítány. I přesto je ale sada 57 matic rysů dostačující k tomu, aby bylo možné porovnat přínosy různých postupů jejich zisku. Jednotlivé rysy jsou přiblíženy v tabulkách 3.4, 3.5, 3.6 a 3.7. Každá tabulka obsahuje rysy získané z jiné zdrojové matice. Sloupec *Bool.* značí, jestli byla před aplikováním transformací použita booleanizace. Ve sloupcích *1. krok* a *2. krok* jsou uvedeny metriky, které byly postupně na danou matici použity. Sloupec *Typ* značí, jaký typ má výstupní matice - jestli se jedná o symetrickou, nebo nesymetrickou matici.V posledním sloupci je uveden název souboru, ve kterém je výsledná matice rysu uložená.

Jak bylo popsáno v předchozích sekcích, název každé matice je tvořen postupným zaznamenáváním prováděných operací, oddělených symbolem podtržítka. Za tečkou je dále doplněna informace o výstupním formátu matice. Přípona *.xdense* značí hustou čtvercovou matici, zatímco přípona *.xsymm* značí hustou čtvercovou a symetrickou matici. Tyto přípony jsou využívané při dalším zpracování matic.

Z odkazovaných tabulek je vidět, že na některé rysy byla znovu aplikována míra *cosinus*. Účelem tohoto postupu bylo přinést další informaci o podobnosti distribuce podobností modelů kontextu, tedy o abstrakci na ještě vyšší úroveň. V další kapitole je hypotéza přínosu tohoto postupu ověřena.

3.4.3 Další možné neimplementované postupy

Z postupů, které lze také využít pro vyhlazování nebo filtrování a nebyly implementovány v této práci zmiňujeme ještě následující dva, i s důvody jejich vypuštění. Kromě dalších kontextových metrik sem patří i metody provádějící jiné druhy vyhlazování než prosté filtrování.

Latentní sémantická analýza (LSA)

Metoda založená na *Singular Value Decomposition* (SVD), jejíž použití na kontextové vektory provádí například Landauer [19], je vyhlazování na ještě vyšší úrovni, než globální (viz předchozí citace), proto by bylo dobré ji také implementovat. Použití LSA by však dobu výpočtů násobně prodloužilo a je proto vhodné spíše do případných rozšiřujících prací.

TF·IDF

Míra *Term Frequency by Inversed Document Frequency* je indikátorem specifičnosti slova vzhledem k *"dokumentu"*, v našem případě k prvku modelu kontextu. Konkrétně se hodnota počítá jako násobek tf, tedy *term frequency*, což je součet hodnot modelu kontextu přes celý jeho řádek v matici kontextu, příslušící počítanému políčku, a idf, který se počítá podle následujícího vzorce, ve kterém *number of documents* reprezentuje počet různých slov z modelu kontextu, se kterými se slovo vyskytuje (ve výsledku počet nenulových hodnot) a *document frequency* reprezentuje součet hodnot přes celý sloupec v matici kontextu, příslušící danému políčku.

$$idf = \log \frac{NumberofDocuments}{DocumentFrequency}$$

Tento postup je používán například v práci Patwardhana a Pedersena [25]. Využitím tohoto postupu by byla získána stejně velká sada matic, jako Booleanizací, která by opět přinesla mezi ostatní míry novou perspektivu, nicméně takové navýšení by též zvýšilo počet zdrojových matic na dvojnásobek. Obdobným způsobem by byla zvýšena i doba výpočtu všech matic z výchozích vycházející, proto byla tato technika vynechána.

3.5 Konstrukce datasetu

Výše popsaným způsobem jsme získali množství matic rysů. Z nich a za pomoci Czech WordNetu zkonstruujeme dataset, který bude dále použit ke trénování, testování a vyhodnocování metody extrakce sémantických informací, popsané v následující kapitole.

Každá instance konstruovaného datasetu odpovídá vektoru rysů pro dvojici lemmat z kartézského součinu průniků domén všech rysů. Pro každou instanci tak máme hodnoty všech rysů odpovídajících dvojici, určující danou instanci. Tyto hodnoty tvoří vektor rysů.

3.5.1 Využití Czech WordNetu

Czech WordNet je zřejmě jediný dostupný strojově čitelný strukturovaný sémantický zdroj pro český jazyk. Je součástí sítě WordNetů evropských jazyků Euro WordNet (EWN) [33]. V této práci byla použita data získaná z CWN ve verzi 1.5.

Z českého WordNetu extrahujeme již anotované sémantické vztahy a použijeme je jako cílovou třídu v našem datasetu. Z instancí, jejichž obě slova patří do domény CWN, vytvoříme data pro trénování a testování. Z instancí odpovídajících ostatním dvojicím, pro které máme všechny rysy, ale ne cílovou třídu, budeme pomocí metody trénované na prvním datasetu získávat nové relace.

Struktura WordNetu

Stejně jako anglickojazyčný Princeton WordNet, i členské projekty EWN, tedy i CWN, mají lemmata strukturována do *synsetů*, které jsou propojeny relacemi, jako jsou *HAS_HYPONYM* nebo *HAS_MERONYM*. Všechna lemmata v jednom synsetu označují jednu entitu nebo objekt, jsou to v tom smyslu tedy synonyma. Strukturu EWN blíže rozebírá Vossen [33].

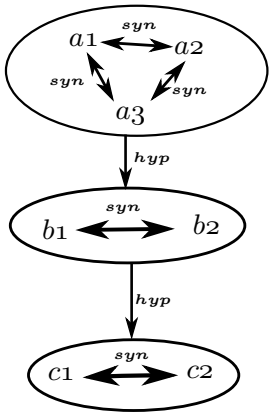
3.5.2 Získávání relací

Pro potřebu strojového učení potřebujeme z CWN získat dvojice lemmat, společně s relací, kterou mezi nimi CWN ukládá. To děláme různě v případu synonym a ostatních relací.

- **Synonyma** — Synonyma jsou získávána přímo ze synsetů. Každá dvojice, kde jsou obě lemmata členy stejného synsetu je do výstupního seznamu uložená jako dvojice synonym.
- **Ostatní relace** — Ostatní relace jsou získávány z CWN z jeho relací mezi synsety. Každá dvojice, kde je jedno lemma v prvním synsetu a druhé lemma ve druhém synsetu, propojeném s prvním určitou relací, je uložena do výstupního seznamu, jako dvojice propojená touto relací.

Kromě relací v CWN, zavedených jako *INTERNAL_LINKS*, jsou jako relace přidány i jejich tranzitivní uzávěry (v algebraickém smyslu). Formálně vyjádřeno je to v následujícím vzorci, kde množina *Relations* obsahuje všechny druhy tranzitivních relací a množina *Synsets* obsahuje všechny synsety v CWN.

$$\forall r \in Relations; x, y, z \in Synsets : (x, y) \in r \wedge (y, z) \in r \implies (x, z) \in r$$



$a1 \leftrightarrow a3$ $a1 \leftrightarrow a2$ $a2 \leftrightarrow a3$ $b1 \leftrightarrow b2$ $c1 \leftrightarrow c2$

$a1 \rightarrow b1$ $a1 \rightarrow b2$ $a2 \rightarrow b1$ $a2 \rightarrow b2$ $a3 \rightarrow b1$ $a3 \rightarrow b2$ $a1 \rightarrow c1$ $a1 \rightarrow c2$ $a2 \rightarrow c1$ $a2 \rightarrow c2$ $a3 \rightarrow c1$ $a3 \rightarrow c2$

Obrázek 3.4: Extrakce relací z Czech WordNetu

Obrázek 3.4 ukazuje přehledně způsob získávání sémantických relací z Czech WordNetu. Z CWN byly tímto způsobem získány relace v množstvích popsaných v tabulce 3.8.

Kvůli nízkým počtům meronym/holonym a antonym byla v dalších postupech použita pouze synonyma, hyponyma a hyperonyma.

Relace	Počet
Synonyma (SYN)	37 196
Hypo/hyperonyma $(HYP)^{*}$	41 715
Mero/holonyma $(MERO)^{*+}$	222
Antonyma (ANT)	230
Celkem	79 363

Tabulka 3.8: Relace získané z Czech WordNetu.

Relací označených hvězdičkou je v součtu dvojnásobek - pro každý směr jedna.

$^{+}$V CWN se vyskytují relace jak MERO_PART, tak MERO_MEMBER (viz Pala [29]). Číslo v tabulce je součtem jejich četností.

NOT relace

Pro trénování metody strojového učení jsou potřeba i negativní příklady instancí, tedy dvojice slov, která spolu v relaci nejsou. Pro zkratku budeme označovat dvojici, ve které nejsou lemmata nijak sémanticky propojena, zkratkou *NOT*.

K tomuto účelu byla použita aproximace, která říká, že pro každou dvojici slov přítomných v CWN obsahuje tento zdroj i jejich relace, pokud existují. *NOT* relace jsou tedy vybírány z dvojic, kde obě slova jsou obsažena v nějakých synsetech CWN, ale nebyl extrahován žádný vztah mezi nimi. Přesněji, obě slova patří do domény WordNetu, ale neexistuje žádná relace z tranzitivního uzávěru množiny relací ve WN zapsaných, která by byla pro tuto množinu definovaná.

Nakolik je tato aproximace přesná jsme zjišťovali ruční anotací *NOT* relací, viz sekce 4.2.1 a sekce s výsledky 4.3.

Diskuse kvality

V této práci již byly zmíněny nedostatky týkající se pokrytí CWN běžného jazyka. Kromě toho registrujeme, že i přesnost může být pro jeho využití jakožto sémantické znalostní báze nedostatečná.

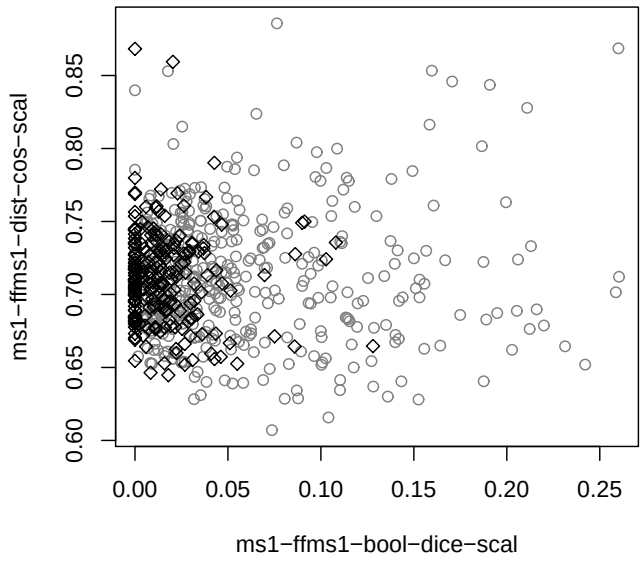
Vzhledem k tomu, že úspěšnost navrhované metody nemůže být lepší, než zdroj, na kterém byla trénovaná, je nezbytné tuto přesnost vyčíslit. Podrobný postup experimentu je popsán v sekci 4.2.1, jeho výsledky pak v 4.3.

4. Použitý postup extrakce sémantických relací

Definice. *Confusion matrix nazýváme tabulku, která shrnuje výsledky predikce metody strojového učení ve vztahu k hodnotám cílové třídy z testovací sady. Detailnější definici uvádějí například Kohavi a Provost [18].*

V našem případě bude cílová třída v trénovacích datech i v predikci nabývat tří hodnot, a to *HYP, SYN* a *NOT*. Hodnota *NOT* bude negativní instance cílové třídy. Buňky v *confusion matrix* budeme adresovat pomocí uspořádané dvojice (A,B), kde A odpovídá predikované hodnotě cílové třídy a B správné hodnotě.

V předchozí kapitole je popsán postup konstrukce datasetu. V této kapitole je tento dataset použit, společně se sémantickými relacemi extrahovanými z CWN, ke trénování a testování modelu strojového učení. Tento natrénovaný model je později využit i k získávání nových relací, které CWN neobsahuje. Úspěšnost tohoto procesu je hodnocena ručně.



Obrázek 4.1: Rozložení relací vzhledem ke dvěma nejlepším metrikám pro *SYN*. *Šedá kolečka označují rozložení synonym vzhledem k daným rysům. Černé diamantíky označují rozložení NOT relací.*

4.1 Automatické testování úspěšnosti na CWN

Nejprve bylo nutné vyladit optimální sadu rysů, která při relativně malé velikosti bude poskytovat metodě strojového učení nejlepší informace. Tato sada rysů byla vybrána automaticky způsobem popsaným v následující podkapitole.

Aby bylo možné využít metody strojového učení s učitelem, byl k relacím *HYP* a *SYN* v datasetu získaným z CWN přidán stejný počet negativních instancí, tedy *NOT* relací.

4.1.1 Použitá metoda strojového učení

Pro účely prezentované úlohy je potřeba vybrat metodu strojového učení, která bude aplikována jak při výběru nejlepších rysů, tak při konkrétní extrakci nových sémantických relací. Pro obě úlohy je nutné použít stejnou metodu, aby byly porovnatelné jak výsledky automatického, tak ručního testování. V první úloze je preferována rychlost učení a predikce, ve druhé naopak přesnost. Pro první úlohu je tak vhodný nějaký druh lineárního klasifikátoru, pro druhou nějaká sofistikovanější metoda. Nakonec byla zvolena metoda strojového učení *Support Vector Machines* (SVM) s lineárním jádrem, tedy lineární *margin classifier*. Tuto metodu blíže popisuje například Mitchell [22]. Byla použita implementace ve specializované lineární verzi knihovny *libsvm* [10], *liblinear* [11]. Nastavení jádra bylo zvoleno tak, aby byla výsledná metoda co nejrychlejší, protože z předběžných pokusů vychází požadavek na rychlost jako naprosto zásadní. Z toho důvodu bylo použito nastavení metody predikce *logistická regrese*, kde navíc knihovna *liblinear* podporuje extrakci pravděpodobností predikce výsledných tříd a predikci do více než dvou tříd, pomocí vícenásobné aplikace na podproblém se dvěma třídami. V úloze automatického testování je predikovaná třída určovaná výběrem nejvyšší rozhodovací hodnoty ze všech tříd.

Použitím lineární metody strojového učení byla splněna podmínka pro výběr nejlepších rysů, je ale možné použít tyto modely i k extrakci nových relací? Na obrázku 4.1 je zobrazena distribuce synonym a *NOT* relací vzhledem ke dvěma nejhodnotnějším rysům pro synonyma. Jak je vidět, shluky se překrývají a nejsou lineárně separabilní. Pravděpodobně by bylo možné nalézt konfiguraci nelineární metody strojového učení, která by vykazovala lepší výsledky, za cenu delší doby trénování, rozdíl oproti lineární metodě ale nebude tak velký, aby znehodnotil získané závěry, které se týkají obecných tendencí v této úloze.

4.1.2 Metodika vyhodnocování

Vzhledem k tomu, že bude prováděna klasifikace do více tříd, musí být aplikovány i míry hodnotící výsledky takové klasifikace. První se nabízí široce využívaná míra *accuracy*, tedy součet všech správně určených prvků v *confusion matrix*, podělený součtem všech hodnot.

Pokud bude tato míra použita jako optimalizace pro výběr rysů, bude maximalizován výkon klasifikace všech tříd. Pro tuto úlohu to ale není nutné. Při extrakci nových relací není potřeba rozpoznávat *NOT* relace. Zde se nabízí prostor pro vylepšení.

Použité metriky

Pro lepší hodnocení, s ohledem na požadované cíle, zaveďme metriku *accuracy without NOT*. Ta bude počítaná obdobně jako *accuracy*, jen do součtů nebude započten prvek matice odpovídající řádku a sloupci relace *NOT*. Její hodnota tedy bude součet všech prvků na diagonále *confusion matrix*, kromě prvku (*NOT*,*NOT*), podělený součtem všech hodnot. Obdobným způsobem je možné zavést i tradiční míry *precision* a *recall* a tedy i *F-measure* [22]. Tyto míry hodnotí úspěšnost komplexněji, než *accuracy*.

	NOT	*SYN*	*HYP*
NOT		FN_{SYN}	
SYN	FP_{SYN}	TP_{SYN}	FP_{SYN}
HYP		FN_{SYN}	

Tabulka 4.1: *TP*, *FP* a *FN* pro relaci *SYN* v *confusion matrix*

Sloupce obsahují hodnoty anotace, řádky obsahují hodnoty predikce.

K zavedení *precision* a *recall* tímto způsobem je třeba určit hodnoty *confusion matrix*, které reprezentují *true positives (TP)*, *false positives (FP)*, *true negatives (TN)* a *false negatives (FN)*. *TP* budou hodnoty správně určených prvků, kromě prvku (*NOT*,*NOT*) a *TN* bude reprezentovat právě prvek (*NOT*,*NOT*). *FP* zase odpovídají součtu hodnot řádku ne-*NOT* relací, kromě správně určeného prvku a *FN* obdobně součtu hodnot sloupce ne-*NOT* relací, opět kromě správně určeného prvku. Pro jeden prvek to ukazuje tabulka 4.1.

Výsledné hodnoty *precision* a *recall* z *confusion matrix M* pro relace *NOT*, *SYN* a *HYP* budou na základě takto určených polí počítány způsobem, který ukazují následující vzorce.

$$Precision(M) = \frac{M_{(SYN,SYN)} + M_{(HYP,HYP)}}{\sum_{r \in Relations} M_{(r,SYN)} + \sum_{r \in Relations} M_{(r,HYP)}}$$

$$Recall(M) = \frac{M_{(SYN,SYN)} + M_{(HYP,HYP)}}{\sum_{r \in Relations} M_{(SYN,r)} + \sum_{r \in Relations} M_{(HYP,r)}}$$

Hodnota *F-measure* se pak bude počítat stejně, jak je obvyklé. Pro $\beta \in [0, 1)$ má větší váhu *precision*. Pro $\beta = 1$ mají *precision* a *recall* stejnou váhu. Jedná se o jejich harmonický průměr [22]. Pro $\beta > 1$ má ve výsledku míry větší váhu *recall*. Vzorec počítání *F-measure* se používá následující vzorec.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

Pro účely získávání relací je důležitější přesnost vydaných výsledků, proto se jako nejvhodnější míra hodnocení automatického získávání relací jeví míra $F_{0,5} - measure$, tedy míra F_β, kde je za parametr β dosazená hodnota 0,5. Tím dostane větší váhu *precision*, tedy přesnost získaných relací. Pouze *precision* není možné použít, protože takto hodnocená metoda extrahuje jen minimální množství relací. Tomuto problému zabraňuje právě přidání vlivu míry *recall*.

Testování

Úspěšnost predikce lineárního SVM na relacích extrahovaných z CWN byla určována pomocí techniky křížové validace (*cross validation* [22]) s deseti rozděleními dat.

Technika křížové validace spočívá v opakovaném rozdělení datasetu na dvě části tak, že testovací část má při n rozděleních velikost $\frac{|Dataset|}{n}$ a trénovací část zbytek. Dataset je takto postupně rozdělen n–krát tak, že všechny vzniklé testovací sady jsou vzájemně disjunktní. Tímto způsobem získáme přesnější odhad úspěšnosti modelu, než při použití pevné trénovací a testovací sady.

Rozdělení datasetu na deset párů trénovacích a vzájemně disjunktních testovacích sad probíhá náhodným způsobem. Jsou připraveny seznamy relací z CWN a NOT relací, oba stejné velikosti, viz sekce 3.5.2. První je promíchán opakovanou výměnou jeho dvou náhodně vybraných prvků mezi sebou. Opakování takové výměny je proveden dvojnásobný počet, než je počet relací v seznamu. Seznam NOT relací je z mnoha dostupných NOT relací vybraný náhodně následujícím způsobem. Opakovaně jsou vybírány náhodné dvojice lemmat z domény CWN a provádí se testování, jestli není obsažena v seznamu relací CVN, nebo již vybraných NOT relacích. Pokud této podmínce dvojice lemmat vyhovuje, je do seznamu NOT relací přidán. Výběr končí dosažením požadované velikosti seznamu.

Každý z těchto seznamů je pak rozdělen na deset stejných dílů, ze kterých jsou skládány trénovací a testovací data pro křížovou validaci. Seznam pro první sadu testovacích dat tak vznikne výběrem dvojic od první až po jednu desetinu datasetu CWN relací a stejné části NOT relací a první sada trénovacích dat křížové validace budou zbylé dvojice v obou seznamech.

```
SelectBestFeatures(Features, measure)
begin
  CurrentSet = {}
  BestSet = {}
  best_performance = 0

  for i in 1 .. |Features| do
    local_best_feature = NULL
    local_best_performance = 0

    for f in Features do
      performance = measure(CurrentSet U {f})

      if performance > local_best_performance then
        local_best_performance := performance
        local_best_feature := f
      endif
    done

    CurrentSet = CurrentSet U {best_feature}

    if local_best_performance > best_performance then
      best_performance := local_best_performance
      BestSet := CurrentSet
    endif
  done

  return BestSet
end
```

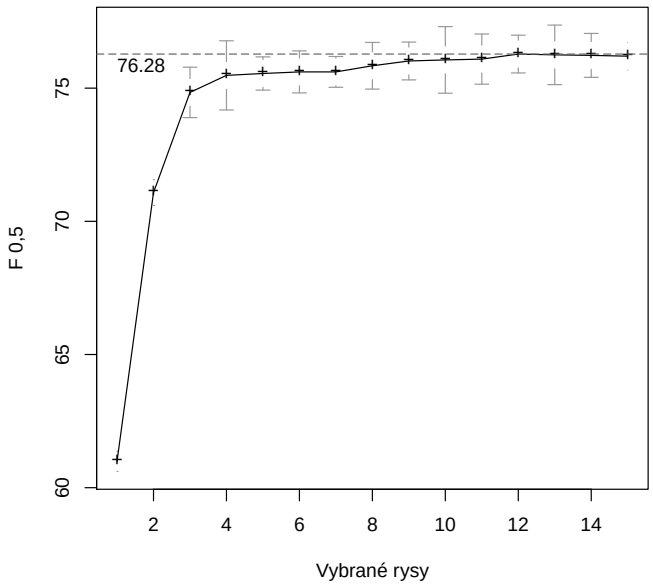
Obrázek 4.2: Algoritmus hladového výběru rysů

Measure je v tomto algoritmu funkce, která provede predikci pomocí modelu trénovaného na vstupní sadě rysů a vrátí hodnotu zvolené míry, aplikované na její výsledek.

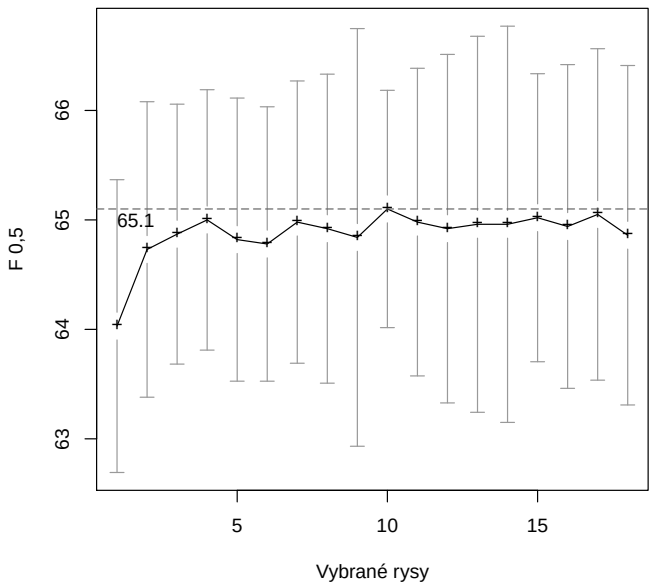
4.1.3 Výběr kvalitních rysů

Postupem popsaným v předchozích částech textu jsme získali celkem 57 rysů. Některé z nich byly získané z podobných matic stejným způsobem, proto je pravděpodobné, že některé dvojice rysů nebudou nezávislé. Z množiny získaných rysů je tak potřeba vybrat ty, se kterými má extrakce relací nejlepší výsledky. Míry hodnotící úspěšnost získaných relací a jejich atributy byly popsány výše. K hodnocení kvality rysů byla na základě předchozího rozboru použita míra $F_{0,5}$.

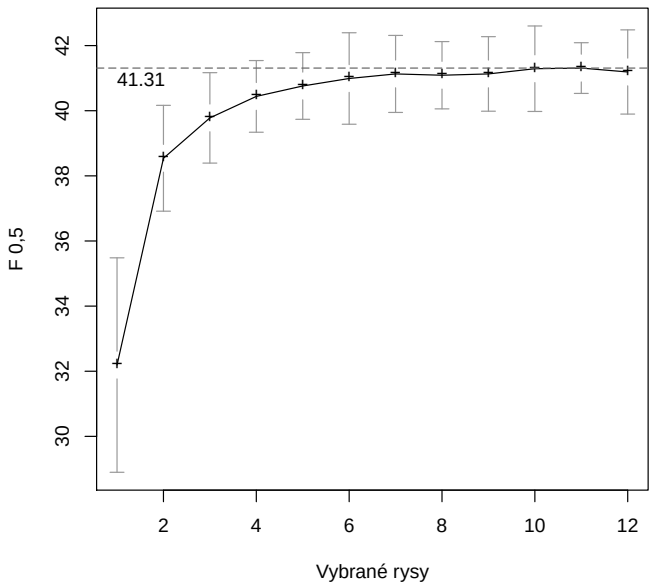
Ideální metoda výběru rysů by zahrnovala postupné hodnocení všech podmnožin množiny rysů, to je ale výpočetně, tedy i časově, příliš náročné, proto byla použita hladová heuristika. Hladovou ji nazýváme proto, že začíná s prázdnou množinou, v každém kole ji rozšíří o rys, který maximalizuje její výkon (má nejvyšší hodnotu použité míry na vydaných výsledcích), a rysy nikdy neodebírá. Konkrétně pracuje podle algoritmu popsaném v pseudokódu na obrázku 4.2.



Obrázek 4.3: Vývoj úspěšnosti automatického testování pro *SYN*, *NOT*. *Vodorovná osa grafu je popsaná pořadím vybraných rysů. Mapování tohoto pořadí na konkrétní rysy je zobrazené v tabulce 4.4. Stejně u následujících dvou obrázků.*



Obrázek 4.4: Vývoj úspěšnosti automatického testování pro *HYP*, *NOT*.



Obrázek 4.5: Vývoj úspěšnosti automatického testování pro *SYN*, *HYP*, *NOT*.

Predikce mezi relacemi	$F_{0,5}$	Accuracy
SYN, NOT	**76,28% ± 0,71**	**74,37%**
HYP, NOT	65,10% ± 1,08	64,05%
SYN, HYP, NOT	41,31% ± 0,78	59,41%
Z toho SYN	46,10% ± 0,71	-
Z toho HYP	22,84% ± 2,22	-

Tabulka 4.2: Výsledky automatického testování

Vyznačený konfidenční interval je počítán na 95% hladině spolehlivosti.

4.1.4 Výsledky

Postupně byly provedeny tři experimenty, které měly za cíl určit nejlepší způsob extrakce sémantických informací a vybrat nejvhodnější rysy. V prvním experimentu byly sady trénovacích i testovacích dat tvořeny pouze dvojicemi lemmat ve vztahu *HYP* a *NOT*, ve druhém *HYP* a *SYN* a ve třetím *SYN*, *HYP* i *NOT*. V každém experimentu byl poměr *NOT* relací ku ostatním relacím 1:1.

Průběh výběru nejlepší množiny rysů pro všechny tři experimenty ukazují obrázky 4.3, 4.4 a 4.5. Jak je vidět, byly experimenty přerušeny vždy ve chvíli, kdy už bylo možné vybrat sadu rysů, dávající dobré výsledky a mající v porovnání s ostatními nízký rozptyl.

Nejvyšší hodnoty dosažené aplikováním popsaného postupu přehledně znázorňuje tabulka 4.2. Výsledky a průběh všech provedených výpočtů jsou uvedeny v adresáři */calculated_data/feature_evaluation* na přiloženém CD.

Hyponymické a hyperonymické relace byly při pokusech sloučeny do jedné cílové třídy, byla tedy predikována pouze tato relace, nikoliv její směr. Při pokusech se samostatnou extrakcí hyponym a hyperonym bylo při automatickém vyhodnocení dosaženo výsledků s úspěšností v řádu pouhých jednotek procent, proto zde nejsou uvedeny. Předpokládáme, že důvodem nízké úspěšnosti je použití pouze jedné nesymetrické metriky pro získávání rysů. Pro lepší výsledky by bylo nutné přidat další nesymetrické rysy.

Jak je vidět z tabulky 4.2, lepší výsledky jsou dosahovány při klasifikaci do dvou tříd odděleně, než při klasifikaci do tří tříd zároveň, proto bude další predikce prováděna odděleně do dvou tříd, tedy do *SYN, NOT* a *HYP, NOT*.

Míra	a	mw4	ms0	ms1
Cosine	H	S	S, HS	S, HS
Dice	H	S, HS	S	S
Jaccard	-	S	-	HS
Lin	H	-	-	-
Distance	-	H	H	-
Coverage	H	HS	-	HS
Spearman	-	-	S	S
ˆCosine	-	HS	-	S, HS

Tabulka 4.3: Výběr rysů v závislosti na metrikách a zdrojových maticích.
Poslední řádek odpovídá rysům získaným aplikací cosinu na matici jiného rysu. 'H' v políčku značí výběr rysu pro extrakci hyperonym, 'S' pro extrakci synonym a 'HS' pro extrakci hyperonym a synonym naráz.

Pořadí	HYP		SYN		HYP, SYN	
1	**16**	*(a_lin)*	**32**	*(ms1_b_dice)*	28	*(ms1_b_cos)*
2	**51**	*(mw4_dist)*	36	*(ms1_dist_cos)*	54	*(mw4_lin_cos)*
3	2	*(a_b_cos)*	**28**	*(ms1_b_cos)*	**52**	*(mw4_jac_cos)*
4	**51**	*(mw4_dist)*	49	*(mw4_dice)*	22	*(ms0_cos)*
5	**4**	*(a_b_cov)*	**32**	*(ms1_b_dice)*	19	*(ms0_b_cos)*
6	10	*(a_dice)*	53	*(mw4_jac)*	49	*(mw4_dice)*
7	24	*(ms0_dist)*	47	*(mw4_cos)*	44	*(mw4_b_cov)*
8	6	*(a_b_dice)*	22	*(ms0_cos)*	30	*(ms1_b_cov)*
9	**4**	*(a_b_cov)*	40	*(ms1_spear)*	33	*(ms1_cos_cos)*
10	**16**	*(a_lin)*	27	*(ms0_spear)*	38	*(ms1_jac)*
11	-	-	21	*(ms0_b_dice)*	**52**	*(mw4_jac_cos)*
12	-	-	**28**	*(ms1_b_cos)*	-	-

Tabulka 4.4: Pořadí výběru rysů při jednotlivých experimentech.
Tučně označené rysy byly vybrány dvakrát.

Diskuze vybraných rysů

Jako nejlepší rysy pro extrakci synonym byly hladovým algoritmem vybrány položky uvedené v tabulce 4.4. Algoritmus výběru rysů umožňuje výběr jednoho rysu vícekrát a jak je vidět ze zmíněné tabulky, opravdu se tak stalo. Při použití ideálního klasifikátoru by stačilo vybrat každý rys v každém experimentu pouze jednou. Při použití lineárního klasifikátoru může ale několikanásobné využití jednoho rysu pomoci v případě, kdy data nejsou lineárně separabilní, což v tomto případě určitě nejsou (viz obrázek 4.1).

Sada rysů vybraná jako nejlepší pro extrakci hyperonym vychází disjunktní s oběma zbylými sadami. Kromě metrik *Distance* a *Lin*, které byly přímo vybrány jen do sady rysů pro hyperonyma, se rysy liší pouze zdrojovou maticí (viz tabulka 4.3). Převážná většina rysů pro extrakci hyperonym vychází ze zdrojové matice typu *a*, tedy z matice, vzniklé z modelu kontextu, načteného ze syntaktické struktury věty, zatímco v ostatních sadách žádný rys vycházející z této matice vybrán nebyl. Z toho lze usuzovat, že tento způsob konstrukce zdrojové matice je pro extrakci hyperonym obzvláště vhodný.

Výběr rysů pro extrakci synonym a i pro sloučený případ má více styčných bodů. Konkrétně rysy 28,49 a 22. Zřejmě i proto jsou výsledky synonym ve třetím experimentu o tolik vyšší, než výsledky hyperonym. Jak je vidět, extrakce hyperonym a synonym jsou dvě značně odlišné úlohy. Pro lepší výsledky tak bude důležité provádět obě úlohy zvlášť, různými predikčními modely.

Dalším zjištěním je, že metriky jsou postupně využity všechny. Nově navržený postup *"umocnění"* matice rysu metrikou cosinus se osvědčil, rysy vzniklé touto metodou patřily k nejdůležitějším při extrakci synonym i ve sloučeném případě.

4.2 Extrakce nových relací

Nové relace jsou vybírány ze seznamu kandidátů, který se skládá z dvojic lemmat, které jsou pak dále hodnoceny. Při selekci kandidátů jsou vybírána lemmata w_1 a w_2 taková, že platí následující.

$$w_1 \in \cap_{r \in Features} Dom_f(r) \backslash Dom_{ss}(CWN) \vee w_2 \in \cap_{r \in Features} Dom_f(r) \backslash Dom_{ss}(CWN)$$

Seznam kandidátů je tak tvořen dvojicemi lemmat, kde alespoň jedno není obsaženo v doméně původního sémantického zdroje. Pro každou dvojici kandidátů je sestaven odpovídající vektor rysů, vybraných v sekci 4.1.3, a tím je zkonstruována kandidátská sada dat. Na tuto sadu dat jsou aplikovány modely, predikující do tříd *SYN*, *NOT* a *HYP*, *NOT*, natrénované na datasetu získaném z celého dodaného sémantického zdroje, opět s vybranými kvalitními rysy. Aplikací modelů získáme rozhodovací hodnoty pro obě třídy. Tím vzniknou dva seznamy, které jsou dále seřazeny podle rozhodovací hodnoty přiřazené k pozitivní hodnotě cílové třídy (k *SYN* nebo *HYP*).

Páry lemmat s nejvyššími rozhodovacími hodnotami z výsledných seznamů jsou prezentovány jako nově získané relace. Úspěšnost popsaného procesu je dále hodnocena ručně, postupem popsaným v následující sekci.

4.2.1 Postup ručního hodnocení

Ruční hodnocení bylo prováděno třemi nezávislými anotátory, kteří hodnotili sadu dat, obsahující 900 dvojic slov. U každé dvojice měli vybrat jednu z následujících možností (příklady *similar* a *related* jsou převzaty z Budanitsky a Hirst [8]).

a) *Tato dvě slova v některém ze svých významů jsou hypo/hyperonyma.*
Například *hruška – ovoce* nebo *hruška – malvice*, ale už ne *hruška – jablko*.

b) *Tato dvě slova v některém ze svých významů jsou synonyma.*

c) *Tato dvě slova v některém ze svých významů jsou významově vztažená (related).*
Například *auto – benzín* nebo *rychlost – úzkost*, ale už ne *benzín – nafta*. Pár *benzín – nafta* sice je významově vztažený, ale zároveň je i podobný, proto má být tato dvojice označena odpovědí *d*).

d) *Tato dvě slova patří v některém ze svých významů do stejného oboru (similar).*
Například *auto – motorka*, *benzín – nafta* nebo *kůň – antilopa*, ale už ne *benzín – auto*.

e) *Tato dvě slova nejsou v žádném sémantickém vztahu, ani si nejsou sémanticky podobná.*

Pokud dvojice anotovaných lemmat vyhovovala více třídám najednou, měli anotátoři instrukce vybírat vždy tu nejspecifičtější z nich. V tom smyslu, že hyponyma, synonyma i slova podobná jsou specifičtější, než slova pouze významově vztažená. Synonymie je zase specifičtější než pouhá podobnost. V případech, kdy byla dvě lemmata vzájemně jak podobná, tak synonymická, byla tato dvojice anotována jako synonyma, ze synonymie totiž podobnost přímo vyplývá. Stejně jako z podobnosti a hyperonymie významová vztaženost.

Před samotnou anotací prošli anotátoři školením na ukázkovém datasetu velikosti 60 relací. Tento dataset i finální hodnocení jsou uloženy v adresáři */calculated_data/annotation* na přiloženém CD. Anotátoři neznali ani zdroj dvojice slov, ani předpovídaný typ relace. Anotátoři během práce také neměli možnost spolu vzájemně komunikovat, tedy nemohli konzultovat sporné dvojice. Instance, u kterých se shodli alespoň dva anotátoři, byly vnímány jako správně určené, ostatní nebyly při hodnocení brány v potaz.

Anotovaná sada 900 dvojic lemmat se skládala z jedné třetiny (tedy 300 dvojic) z *NOT* relací, tedy ze slov, mezi kterými nebyla očekávána žádná relace. Výběr těchto relací je založen na hypotéze popsané v podsekci 3.5.2. *NOT* relace tedy přidáváme nejen kvůli vyvážení počtu dvojic v relaci a počtu nevztažených dvojic (aby nebyli anotátoři ovlivněni častějším výskytem jedné možnosti), ale i kvůli ověření zmíněné hypotézy.

Dalších 300 dvojic tvořily páry, mezi nimiž existuje v CWN přímý nebo nepřímý vztah, a to buď hypo/hyperonymický nebo synonymický. V tomto pokusu nebyly vzájemně rozlišovány hyponyma a hyperonyma. Výsledek ohodnocení této části ověří spolehlivost relací, získaných z CWN.

Posledních 300 dvojic bylo tvořeno páry, které podle našeho modelu jsou v nějaké sémantické relaci. Konkrétně byl z každého ze seznamů popsaných v sekci 4.2 vybrán stejný počet dvojic lemmat s nejvyšší rozhodovací hodnotou přiřazenou cílovému ohodnocení. Sada dvojic lemmat pro anotaci tedy obsahovala 150 nejlépe hodnocených dvojic pro třídu *SYN* a 150 pro třídu *HYP*.

Shrnutí a analýza ručního hodnocení výsledků extrakce nových sémantických relací jsou provedeny v sekci 4.3.

4.3 Výsledky získávání nových relací

V této sekci jsou uvedeny získané výsledky a ověřena platnost hypotéz, vyslovených ohledně kvality relací, získaných z CWN, a ohledně metody výběru *NOT* relací. Dále jsou zde uvedeny detailní přehledy anotace a příklady získaných relací. Seznam predikovaných dvojic lemmat, na kterých se shodli alespoň dva anotátoři, je uvedený v příloze C.

4.3.1 Shoda mezi anotátory

Při značkování sémantických vztahů postupují anotátoři podle svých vědomostí a dosavadních životních zkušeností. Je tedy běžné, že se v mnoha případech neshodnou na stejném způsobu anotace páru lemmat. Čím více anotátorů značkuje stejnou sadu párů lemmat, tím je získán přesnější výsledek kvality relací. V našem případě byla anotace provedena třemi nezávislými anotátory. Za platné ohodnocení dvojice lemmat byly považovány pouze situace, kdy se shodli alespoň dva z nich.

Shodu anotátorů na hodnocení jednotlivých kandidátů na relace přehledně znázorňuje tabulka 4.5. Detailnější rozbor shody anotátorů na jednotlivých odpovědích vzhledem k označení dvojic lemmat v jednotlivých zdrojích uvádějí tabulky 4.6, 4.7 a 4.8.

Úroveň shody:	Shoda tří		Shoda dvou			žádná shoda	
Odpověď	Abs.	Rel.	Abs.	Rel.	Právě 2	Abs.	Rel.
a) Hyphonyma	46	5,1%	93	10,3%	(47)	32	3,6%
b) Synonyma	35	3,9%	77	8,6%	(42)	24	2,7%
c) Related	14	1,6%	94	10,4%	(80)	0	0%
d) Similar	28	3,1%	80	8,9%	(52)	0	0%
e) Not	399	44,3%	498	55,3%	(99)	2	0,3%
Celkem	522	58%	842	93,6%	(320)	58	6,4%

Tabulka 4.5: Shoda anotátorů na hodnocení jednotlivých dvojic.

V řádcích jsou uvedeny počty shodných odpovědí na anotační otázky.

Zdroj	Relace	Hodnota shody alespoň dvou					Celkem
Not		a) Hyp.	b) Syn.	c) Rel.	d) Sim.	e) Not.	
	not	1	1	9	1	**286**	298
Celkem z not		1	1	9	1	286	298
Predicted		a) Hyp.	b) Syn.	c) Rel.	d) Sim.	e)Not.	
	hyp	**4**	23	18	31	59	135
	syn	11	**17**	50	36	25	139
Celkem z predicted		15	40	68	67	84	274
Wn		a) Hyp.	b) Syn.	c) Rel.	d) Sim.	e) Not.	
	hyp	**28**	17	11	8	69	133
	syn	49	**19**	6	4	59	137
Celkem z wn		77	36	17	12	128	270
Celkový součet		93	77	94	80	498	842

Tabulka 4.6: Shoda dvou anotátorů s předpovědí

Zdroj	Relace	Hodnota shody všech tří					Celkem
Not		a) Hyp.	b) Syn.	c) Rel.	d) Sim.	e) Not.	
	not	0	1	1	0	**256**	258
Celkem z not		0	1	1	0	256	258
Predicted		a) Hyp.	b) Syn.	c) Rel.	d) Sim.	e)Not.	
	hyp	**1**	11	5	10	51	78
	syn	3	**5**	4	16	10	38
Celkem z predicted		4	16	9	26	61	116
Wn		a) Hyp.	b) Syn.	c) Rel.	d) Sim.	e) Not.	
	hyp	**17**	12	2	1	40	72
	syn	25	**6**	2	1	42	76
Celkem z wn		42	18	4	2	82	148
Celkový součet		46	35	14	28	399	522

Tabulka 4.7: Shoda všech tří anotátorů s předpovědí

Relace	Shodovaná hodnota	Shoda A-B	Shoda A-C	Shoda B-C	Průměr
Hyp.	**hyp**	20	23	25	22,7
	syn	24	26	36	28,7
	rel	14	13	16	14,3
	sim	13	14	34	20,3
	not	105	107	98	103,3
Správně z hyp		20/ 176	23/183	25/209	
Syn.	hyp	39	40	37	38,7
	syn	16	17	25	19,3
	rel	23	30	15	22,7
	sim	21	19	34	24,7
	not	77	57	54	62,7
Správně ze syn		16/176	17/163	25/165	
Not.	hyp	0	1	0	0,3
	syn	1	1	1	1
	rel	3	4	4	3,7
	sim	0	0	1	0,3
	not	272	267	259	266
Správně z not		272/276	267/273	259/265	
Celková shoda		308/628	307/619	309/639	
V procentech		49,0%	49,6%	48,4%	

Tabulka 4.8: Shoda anotátorů po dvojicích

4.3.2 Hodnoty metrik úspěšnosti

Tabulka 4.9 ukazuje procenta správně určených relací *HYP*, *SYN* a *NOT* a procenta správně určených relací podobných a vztažených. Všechna čísla jsou získána z tabulky 4.6, která uvádí počty shod alespoň dvou anotátorů po jednotlivých odpovědích ve vztahu s cílovými třídami, určenými jednotlivými zdroji.

Hodnoty správně určených *HYP + SYN* odpovídají součtu přesné shody anotace s cílovými třídami jednotlivých zdrojů dvojic, vyděleném celkovým počtem všech dvojic z daného zdroje, u kterých se na nějaké odpovědi shodli alespoň dva anotátoři. Pro dvojice nově extrahované metodou popisovanou v této práci je to tedy $(4 + 17)/274 = 0,0766$ a pro dvojice získané z CWN je to $(28 + 19)/270 = 0.1741$.

Procenta dvojic zahrnutých do sloupce *Similar* odpovídají součtu shod počtu dvojice, anotované jako synonyma, s cílovými třídami a počtu synonym ve zdroji označených jako *similar*, vyděleném celkovým počtem všech synonym z daného zdroje, u kterých se na nějaké odpovědi shodli alespoň dva anotátoři. Nejedná se zde tedy o přesnou shodu anotace se zdrojem dvojic, ale o volnější určení slov stejného nebo podobného významu.

Procenta dvojic ve sloupci *Related* jsou získána obdobně jako v případě sloupce *Similar*. Jedná se o součet všech ne-*NOT* políček (tedy všech kromě odpovědi *e*) daného zdroje, vydělený celkovým počtem všech dvojic zdroje, u kterých se na nějaké odpovědi shodli alespoň dva anotátoři. Tento počet odpovídá poměru dvojic v jakékoliv sémantické relaci (jak je chápali anotátoři), získaných z daného zdroje.

Sloupec *NOT* ukazuje poměr shody anotátorů s hypotézou extrakce *NOT* relací. Výsledná hodnota je uvedena v řádku CWN, protože *NOT* relace byly získány také z CWN. V tabulce 4.6 jsou ale počty pro výpočet této hodnoty uvedeny v řádku odpovídajícím zdroji *Not*.

Získané výsledky jsou detailně rozebrány dále, v sekci 4.4.

	Správně určených HYP + SYN	Similar	Related	NOT
CWN	17,41%	16,79%	48,12%	95,97%
Predikce	7,66%	38,13%	56,3%	—

Tabulka 4.9: Shodně označené relace z WN a predikované při shodě dvou anot.

4.3.3 Příklady nově získaných relací

Seznam predikovaných relací, na kterých se shodli alespoň dva anotátoři, je uveden v příloze C. V této podsekci se podíváme blíže na některé skupiny dvojic lemmat.

Tabulka 4.10 uvádí seznam dvojic, na kterých se shodli jak všichni anotátoři, tak model predikce. Je jich celkem šest, ze 116 predikovaných relací, na kterých se shodli tři anotátoři viz tabulka 4.7.

Tabulka 4.11 naopak uvádí příklady dvojic lemmat, mezi kterými byl predikovaný nějaký vztah, ale žádní dva anotátoři se na nich neshodli. Z tohoto přehledu je vidět jak rozdílné vnímání sémantiky předložených slov anotátory, tak specifika ruční anotace, tedy že i dvojice slov, které analogicky téměř odpovídají příkladům v zadání (např. *rok – měsíc = Similar*) nejsou všemi anotátory shodně ohodnoceny.

Relace	První lemma	Druhé lemma
syn	obličej	tvář
syn	paragraf	ustanovení
syn	předpis	zákon
syn	skvrnka	skvrna
syn	zákon	ustanovení
hyp	výrobek	zboží

Tabulka 4.10: Predikované relace, na kterých se všichni anotátoři shodli.

První lemma	Druhé lemma	Anot. A	Anot. B	Anot. C	Predikce
pomluva	polopravda	not	rel	sim	hyp
rok	měsíc	not	rel	sim	hyp
šlechtic	spisovatelka	not	rel	sim	hyp
uzenina	obilovina	not	rel	sim	hyp
vnitro	finance	not	rel	sim	hyp
člen	předseda	rel	hyp	sim	hyp
pole	indukce	rel	sim	not	syn
společnost	podíl	rel	sim	not	syn

Tabulka 4.11: Ukázky dvojic lemmat, na kterých se žádní dva anotátoři neshodli.

4.4 Shrnutí výsledků

Z pohledu na výsledky shody anotátorů je zřejmé, že jejich odpovědi měly značný rozptyl. Proto je k vyhodnocení použita tabulka relací, na kterých se shodli alespoň dva anotátoři.

4.4.1 Diskuse

Jak ukazuje tabulka shody anotátorů po dvojicích 4.8, i mezi sebou se na správné hodnotě lidští hodnotitelé shodli v méně než polovině případů. I z toho důvodu je nutné dívat se na výsledky s určitým odstupem. Přesto v nich můžeme pozorovat určité tendence.

Výběr *NOT* relací

V sekci 3.5.2 byla vyslovena hypotéza o způsobu výběru *NOT* relací. Z posledního sloupce tabulky 4.9 je vidět, že úspěšnost metody výběru *NOT* relací byla anotátory ohodnocena na téměř 96%. Takto vysoké číslo značí, že prezentovaný postup (výběr dvojic lemmat, kde alespoň jedno není z domény vstupního sémantického zdroje) lze využít, a opravdu vydává dvojice lemmat, které spolu v relaci nejsou.

Využití Czech WordNetu

V sekci 3.5.2 byla vyslovena hypotéza, že mnoho relací extrahovaných z Czech WordNetu není jednoznačných. Po prozkoumání výsledků můžeme říct, že pro lidské anotátory je obtížné správně klasifikovat relace ze vzorku, pokud lemmata nejsou vztažena ke svým synsetům, tedy jsou-li vytržená z kontextu. Anotátor se pak zaměřuje na nejběžnější význam daného slova a předkládaná dvojice lemmat je anotována jakožto nevztažená. Toto se děje i po důkladné vstupní instruktáži, říkající mimo jiné, že dvojice lemmat mají být anotovány v určité relaci, pokud jí slova vyhovují alespoň v nějakém ze svých významů. Zmíněný jev je doložen vysokými čísly ve sloupci odpovědí e v tabulce 4.6. Tyto výsledky neznamenají, že CWN nelze pro tento účel využít, jiný strojově čitelný sémantický zdroj pro češtinu totiž nemáme. Je ale možné tvrdit, že metodika konstrukce synsetů (často příliš velkých) není pro tento účel nejvhodnější.

Výkon metody

Souhrnné úspěšnosti ruční anotace jsou uvedené v tabulce 4.9. Při porovnání hodnot v obou řádcích tabulky, tedy výsledků predikce a výsledků hodnocení relací z CWN, je možné prohlásit, že použitý postup extrakce relací je relevantní. Přesto výsledky naznačují, že se spíše než na extrakci konkrétních relací hodí k určování sémantické podobnosti a vztaženosti.

Stejně jako anotátoři, i prezentovaná metoda pracuje s většinovými významy předložených lemmat, proto jsou výsledky *similarity* a *relatedness* dokonce lepší, než u dvojic lemmat, mezi kterými v CWN je nějaký vztah. Tato vlastnost je hlavním přínosem, který by výsledná sada nástrojů mohla mít při poloautomatické konstrukci sémantického zdroje.

Dále nebyla provedena samotná konstrukce desambiguované sémantické sítě, pouze extrakce sémantických relací. Důvodem je nízká přesnost extrakce konkrétních sémantických relací (hyperonym a synonym). Uvedené výsledky v tabulce 4.9 jsou horním odhadem úspěšnosti všech nově extrahovaných relací. Soubory nově extrahovaných synonym a hyperonym spolu s pravděpodobností jejich úspěšného určení, vzniklé aplikací software na celá data, jsou uloženy na přiloženém CD v adresáři */calculated_data*.

Z údajů o sémanticky vztažených dvojicích slov by bylo možné pomocí shlukování (neboli *clusteringu*) synonym, jak provádí Lin [21], sestavit synsety a ty pomocí hledání maximální kostry hyperonym sestavit do sítě, jak popisuje například Andrews [5]. Takto vytvořená síť by ale zřejmě měla nízkou úspěšnost jednotlivých relací, nebyla by tedy zřejmě prakticky použitelná.

Tato práce se specializovala spíše na zisk sémantických informací za pomoci aplikace nových postupů, než na opakování již popsaných procedur.

4.4.2 Další možná vylepšení

Implementací dalších technik je jistě možné dosažené výsledky ještě vylepšit. Například nebylo provedeno žádné ladění parametrů modelu SVM. Je možné že i využití nějaké jiné metody strojového učení by přineslo signifikantní zlepšení.

Jak je zjištěno v Agirre et al. [4], použití většího objemu nestrukturovaných zdrojových dat může přinést také značné zlepšení výsledků. Provedení postupu pro jazyk, pro který existuje více zdrojů (např. angličtinu) by tedy mohlo přinést ještě lepší výsledky experimentů.

5. Uživatelská dokumentace

Tato kapitola obsahuje uživatelský návod na instalaci a použití programů a skriptů, které jsou součástí práce:

- **FeatureRetriever (FR)** — program pro transformace matic kontextu.
- **EvaluateFeatures(EF)** — program pro vyhodnocení úspěšnosti sady rysů a extrakci nových rysů
- **WN-Transformer (WNT)** — program pro zpracování WordNetu v jeho textovém formátu.

5.1 Příprava prostředí a instalace

Programy byly vytvářeny a testovány v prostředí operačního systému Linux, konkrétně na distribuci Ubuntu 11.04. Náročné výpočty byly prováděny na Linguistic Research Clusteru (LRC), laskavě poskytnutém Ústavem Formální a Aplikované Lingvistiky na MFF UK. Na tomto clusteru je zavedené prostředí Sun Grid Engine, přiblížené v dále v sekci 5.1.1.

Součástí instalace programů je i jejich sestavení ze zdrojových kódů, proto je jakožto prerekvizita vyžadován kompilátor jazyka C++, nejlépe na Linuxu široce rozšířený *g++* (nejlépe verzi 4.4 nebo pozdější), a program *make*, oba jsou standardně k dispozici v repozitářích balíčků.

Dále je potřeba mít korektně nainstalovanou sadu knihoven *Boost* (testováno s verzí 1.41), která je také přítomná v repozitářích Ubuntu. Pro ostatní distribuce jsou dostupné ke stažení a ruční instalaci na *http://www.boost.org/users/download/*.

Před dalším postupem instalace by měly být všechny výše zmíněné programy a knihovny nainstalovány a konfigurovány.

V následujících podsekcích popíšeme ještě přípravu dalších tří prerekvizit, které nejsou standardní součástí distribucí Linuxu. Jedná se o sadu nástrojů a knihoven *SuperMatrix*, vyvíjenou na univerzitě ve Vratislavi, nástroj pro práci s anotovaným korpusem, *Tred*, vyvíjený v Ústavu Formální a Aplikované Lingvistiky na MFF UK a *LibLinear*, knihovnu lineárního SVM klasifikátoru.

5.1.1 Sun Grid Engine

Sun Grid Engine (SGE) [13] je prostředí umožňující snadno distribuovat výpočetní úlohy mezi stroje zapojené do *gridu*. Protože byly výpočty prováděny na LRC, obsahují skripty provádějící náročnější výpočty odkazy na nástroje *SGE*. Toto prostředí je tak pro využití sestavených skriptů nezbytné.

5.1.2 SuperMatrix

Projekt *SuperMatrix*, vyvíjený na univerzitě ve Vratislavi byl vytvářen jako sada nástrojů pro usnadnění ruční konstrukce polské mutace EWN, Polish WordNetu (PlWN).

SuperMatrix obsahuje nástroje pro vytváření, načítání a ukládání řídkých matic kontextu s přidanými předpočítanými hodnotami pro řádky a sloupce (například entropii). Dále obsahuje framework pro výpočet různých metrik *similarity* a *relatedness* pro dvojice vektorů kontextu, vážení kontextových hodnot a jejich filtrování. Dokumentace k projektu, obsahující i instalační instrukce je obsažená v repozitáři zdrojových kódů v sekci *supermatrix/doc/manual*. Pro přístup k repozitáři je nutné si nejprve zajistit akademickou licenci přímo z univerzity ve Vratislavi.

Kromě knihoven *matrices* a *comparator*, připravených při instalaci *SuperMatrixu*, využijeme také nástroj na konstrukci matice z kontextu. Tento nástroj se nazývá *tuplesproc* a je umístěn v adresáři *supermatrix/tools/architect2/tuplesproc* repozitáře. Při instalaci *SuperMatrixu* by mělo proběhnout jeho sestavení. Pokud ale nastanou chyby a po instalaci není nástroj *tuplesproc* sestavený, je možné jej dodatečně sestavit pomocí souboru programu *make* (tzv. *makefile*), umístěného na CD přiloženém k této práci jako */software/misc/tuplesproc.make*. Tento soubor stačí přesunout do adresáře se zdrojovými souboru nástroje *tuplesproc* a použít s ním program *make*.

Volání a funkčnost programu *tuplesproc* jsou dále rozepsány v sekci 5.2.

5.1.3 Tred

Používaný korpus PDT, vytvářený v Ústavu Formální a Aplikované Lingvistiky na MFF UK je doplněný sadou nástrojů, které korpus zpracovávají. V naší práci využíváme nástroj *btred*, který je součástí programu pro vytváření, prohlížení a editaci větných stromů, *Tredu* [23]. Nástroj *btred* slouží k provedení operací, definovaných v dávkovém souboru jazyka perl, na větách z korpusu.

Tred lze získat na adrese *http://ufal.mff.cuni.cz/~pajas/tred/*, kde je i jeho instalační návod a rozsáhlý manuál.

5.1.4 Instalace programů

Programy příslušné k této práci jsou na CD uložené ve formě zdrojových souborů, které stačí jen sestavit. Nejprve je třeba celý adresář */software* z CD zkopírovat na vybrané místo na disku, například adresáře *~/kirschner_thesis*, který si uživatel sám vytvoří. V následujícím textu budeme toto umístění používat jako instalační adresář. Instalovat programy lze samozřejmě do libovolného jiného adresáře, jehož názvem v instalačním postupu nahradíte tento ukázkový.

Předpokladem pro sestavení je splnění výše popsaných prerekvizit a přítomnost adresáře *liblinear* s přeloženou knihovnou v adresáři se zdrojovými kódy, viz níže.

Sestavení programů lze nejjednodušeji provést spuštěním instalačního skriptu, lokalizovaného v *~/kirschner_thesis/software/install.sh*. Během instalace jsou soubory přeloženy a sestaveny. Na závěr jsou výsledné sestavené programy zkopírovány do adresáře *~/kirschner_thesis/software/bin*, odkud je pak lze spouštět.

Volání a funkčnosti jednotlivých programů jsou dále rozepsány v následujících sekcích.

5.1.5 LibLinear

Knihovna *LibLinear* poskytuje funkčnosti trénování modelu lineárního SVM a predikci jeho pomocí. Bližší informace o funkcích využívaných z této knihovny jsou uvedeny v kapitole 6.

LibLinear lze stáhnout ze stránek Machine Learning Group at National Taiwan University na adrese *http://www.csie.ntu.edu.tw/~cjlin/liblinear/*. K dispozici je varianta pro MS Windows a pro Linux. Ze zřejmých důvodů je třeba stáhnout archiv zdrojových souborů pro Linux.

Stažený archiv je následně třeba rozbalit a přesunout do adresáře *~/kirschner_thesis/software/src/* pod jménem *liblinear*, viz předchozí podsekce. Pro sestavení pak stačí v adresáři *~/kirschner_thesis/software/src/liblinear* spustit příkaz *make all*. Pro verzi knihovny liblinear-1.8 provedeme instalaci následující sekvencí příkazů v terminálu. Předpokládáme stažený archiv umístěný relativně v aktuálním adresáři.

```
#Rozbalení archivu
tar -zxvf liblinear-1.8.tar.gz

#Přesunutí a přejmenování adresáře
mv liblinear-1.8 ~/kirschner_thesis/software/src/liblinear

#Změna aktuálního adresáře do adresáře liblinear
cd ~/kirschner_thesis/software/src/liblinear

#Sestavení knihovny liblinear
make all
```

5.2 Příprava dat

Systém je nastaven na vstupní data ve formátu PML [24], tedy ve formátu, ve kterém je uložený korpus PDT 2.0. Takováto vstupní data jsou dále zpracovávána pomocí *btredu* (viz výše) do čtveřic v následujícím formátu:

```
FIRST_LEMMA;RELATION;SECOND_LEMMA;COUNT
```

FIRST_LEMMA a *SECOND_LEMMA* jsou oba členové dvojice slov, relation je relace, která je přidána k *SECOND_LEMMA* jako popisek sloupce. *COUNT* je pak hodnota modelu kontextu pro tuto dvojici, která musí být z oboru kladných celých čísel.

5.2.1 Získávání hodnot modelu kontextu

Výše uvedené čtveřice jsou získávány podle požadovaného typu kontextu, jak je uvedeno v sekci 3.2, skripty uloženými na CD v adresáři */software/scripts*. Jedná se o skripty spouštějící program *btred* v prostředí *SGE*, které mají jako vstup soubory s daty a jako výstup výše popsané čtveřice. Konkrétní určení který skript prování jaké operace naleznete v příloze A – obsahu přiloženého CD.

Pokud uživatel bude chtít využít vlastní metody získávání modelu kontextu a vlastní nástroje transformující data, je to velice snadno možné, jen musí být dodržen výsledný formát čtveřic.

5.2.2 Konstrukce kontextových matice

Pro konstrukci matice je použit program ze sady nástrojů *SuperMatrix*, *tuplesproc*, jehož instalace je popsána v předchozí sekci. Na vstupu dostává cestu, kde má být výsledná matice uložena, její název a soubor obsahující seznam kontextových čtveřic. Konkrétní volání vypíše program *tuplesproc* při spuštění bez parametrů.

Program *tuplesproc* může být ovládán také dávkovým souborem */software/scripts/create_matrix.sh*. Před jeho spuštěním je ale třeba doplnit v něm správné cesty a názvy potřebných souborů.

5.3 Výpočet rysů

Výpočet matic rysů provádí program *FeatureRetriever*. Na vstupu dostává cestu, kde jsou uložené matice, název vstupní matice, název výstupní matice a konfigurační soubor, obsahující popis transformace, která má být programem provedena. Konfigurační soubor obsahuje záznamy v následujícím formátu.

```
PARAMETER_NAME=VALUE
```

Hodnoty, kterých může nabývat *PARAMETER_NAME* jsou *COMPARATION_METHOD_TYPE*, *PROCESS_ROWS*, *METHOD* a *FREE_THREADS*. První tři parametry ovládají výběr transformace matice, poslední parametr, *FREE_THREADS*, určuje počet procesorů, který má program při výpočtu nevyužít.

Parametr *COMPARATION_METHOD_TYPE* určuje, jestli bude využita metoda externí (hodnota parametru 'SuperMatrix'), nebo metoda implementovaná v rámci této práce (hodnota parametru 'Semantix').

Parametrem *PROCESS_ROWS* je určen proces výpočtu. Jeho hodnota 'AllAtOnce' značí, že matice bude transformovaná celá v kuse, bez dělení na jednotlivé bloky, počítané zvlášť v různých vláknech. Zbylé dva způsoby zpracovávání matice ji dělí na stejně objemné díly po řádcích a zpracovávají hodnoty buď pro bloky tvořené celými řádky (hodnota parametru *PROCESS_ROWS* 'AllxAll'), nebo jen pro díly horní trojúhelníkové matice, tedy jednu z identických polovin symetrické matice (hodnota parametru *PROCESS_ROWS* 'AllxAllSymmetric').

Parametr *METHOD* říká, která konkrétní transformace bude provedena. Jeho hodnota má jasně daný formát, Nejprve je uveden název metody, poté bez mezery následuje otevírací závorka. Po ní je uveden seznam argumentů metody v následujícím formátu.

```
KEY=VALUE
```

Jednotlivé argumenty s hodnotami jsou, opět bez mezer, od sebe odděleny čárkou, a seznam je uzavřen kulatou zavírací závorkou. Následující kód uvádí příklad konfiguračního souboru pro filtrování matice.

```
# Bude použita interně implementovaná metoda
COMPARATION_METHOD_TYPE=Semantix

# Matice nebude při transformaci rozdělována na části
# pro paralelní zpracování
PROCESS_ROWS=AllAtOnce

# Aplikace filtru, kde výsledná matice bude splňovat podmínky:
#  - minimální hodnota součtu sloupce bude 14
#  - minimální hodnota součtu řádku bude 30
#  - maximální hodnota součtu sloupce bude 2000
#  - minimální počet nenulových políček ve sloupci bude 7
#  - minimální počet nenulových políček ve řádku bude 15
#  - minimální entropie sloupce bude 2.21
#  - minimální entropie řádku bude 3.06
METHOD=Filter(minFCount=14,minRCount=30,maxFCount=2000,minFNZCount=7,
minRNZCount=15,minFEntropy=2.21,minREntropy=3.06)

# Hodnota tohoto parametru nemá při neparalelní metodě význam
FREE_THREADS=0
```

Řádky konfiguračního souboru začínající # jsou považovány za komentáře a ignorovány, stejně jako prázdné řádky. Řádky obsahující určení hodnoty parametru musí být bez mezer.

Sada konfiguračních souborů transformací matic je uložená v adresáři */software/cfg/* na CD. Posloupnosti transformací rysů mohou být provedeny dávkovými soubory shellu, připravenými v adresáři */software/scenarios/* na CD.

5.4 Transformace WordNetu a vyhodnocení rysů

K naučení modelu strojového učeni jsou potřeba trénovací relace, extrahované z již existujícího sémantického zdroje. K získání těchto relací slouží nástroj *WN-Transformer*. Na vstupu dostane instrukci, jaký má vydat druh výstupu, a kromě dalších konfiguračních parametrů i soubor s WordNetem v textové podobě (s příponou '.ewn'). Všechny parametry konfigurace programu jsou vypsány při jeho spuštění bez parametrů.

Druhy výstupů programu *WN-Transformer* podle prvního parametru jsou následující.

1. **lemmas** — S tímto parametrem program vypíše pouze seznam lemmat, které vstupní WordNet obsahuje.

2. **relations** — Tento parametr nastavuje program na získávání relací. Dále lze pomocí přepínačů nastavit, jestli mají být do výsledku zahrnuty i doplňkové *NOT* relace a jaký má být jejich poměr, jestli mají být vynechána víceslovná lemmata, jestli mají být extrahovány i relace přítomné až v tranzitivním uzávěru orientovaných relací a které všechny slovní druhy mají být do výstupu zahrnuty. Výstupní seznam obsahuje na každém řádku trojici *první_lemma*, *druhé_lemma* a *relace*, oddělené mezerou. Pár výstupních lemmat je u směrových relací seřazen tak, že první slovo je to obecnější, vzhledem k relaci.

Program WN-Transformer je možné spouštět také pomocí dávkových souborů, které vyhodnocují výkon získaných rysů. Tyto dávkové soubory jsou na CD umístěné v adresáři */software/scenarios/evaluate_features*.

Typický příklad spuštění je následující

```
wn_transformer relations -c 1 -m -p n \
   -r ~/kirschner_thesis/software/cfg/wn_transformer/EWNRels.cfg \
   -t wn_file.ewn > wn_relations.list
```

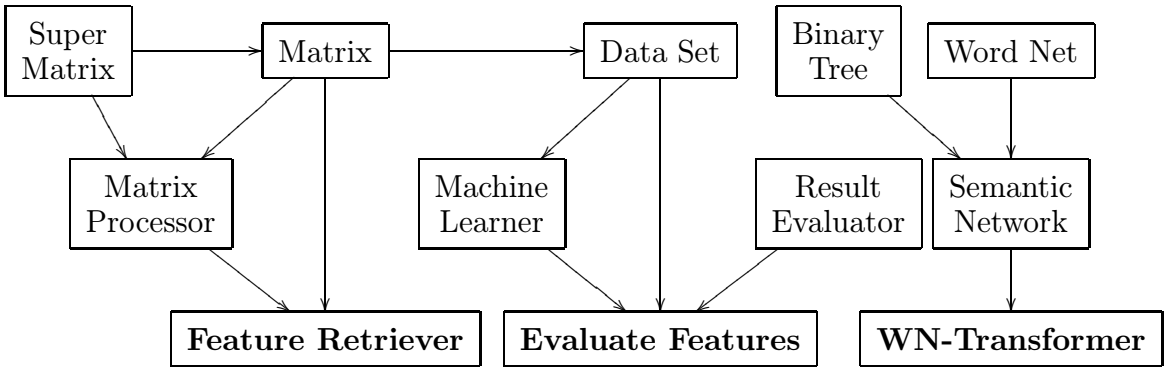
5.4.1 Vyhodnocování rysů

Posledním zbývajícím programem je *FeatureEvaluator*. Tento nástroj sestaví dataset ze vstupních matic rysů a relací s WordNetu a poté na něm buď provede křížovou validaci a ohodnotí tak kvalitu rysů, nebo rovnou extrahuje nové relace. Správný syntax spouštění program vypíše, pokud je spuštěn bez parametrů. Vstupem jsou: cesta k maticím rysů, příznak, jestli má provádět křížovou validaci, extrahovat nové relace, nebo jen vytisknout zkompilovaný dataset, soubor se seznamem relací s WordNetu a seznam názvů matic s rysy.

Stejně jako *WN-Transformer*, i *FeatureEvaluator* je připravený pro použití v dávkovém souboru v adresáři */software/scenarios/evaluate_features* na CD. Tento dávkový soubor za pomoci programů *WN-Transformer* a *FeatureEvaluator* provádí hladový výběr rysů, popsaný na obrázku 4.2.

6. Programová dokumentace

V této sekci je rozebrána jak struktura návrhu jednotlivých programů, tak i použité algoritmy a datové struktury. Nejprve jsou přiblíženy moduly a knihovny sdílené více programy a pak jednotlivé součásti programů vytvořených v rámci této práce. Jedná se o program pro transformaci matic kontextu *FeatureRetriever (FR)*, program pro vyhodnocení úspěšnosti sady rysů a extrakci nových rysů *EvaluateFeatures(EF)* a program pro zpracování WordNetu v jeho textovém formátu, *WN-Transformer (WNT)*. Schéma rozložení funkčností do modulů znázorňuje obrázek 6.1.



Obrázek 6.1: Schéma modulů programů vytvořených v rámci této práce.

Všechny tři zmíněné programy byly napsány v programovacím jazyce C++ s využitím pomocných knihoven *Standard Template Library (STL)* a Boost, viz 5.1. Vlastní zdrojový kód celkově čítá přes 4 500 řádků, rozdělených do 36ti souborů. Do těchto počtů nejsou započítány skripty, kterých práce obsahuje desítky.

6.1 Moduly a knihovny sdílené více programy

K dobrým postupům při návrhu programů patří rozdělení funkčností do modulů tak, aby byly tyto moduly samostatně využitelné v různých aplikacích. V případě předložených tří programů se vlastní funkčnosti překrývají jen minimálně. V programech FR a EF je shodně využit modul spravující matice kontextu a ve všech třech programech je použit zdrojový soubor obsahující hlavičky několika pomocných nástrojů *Tools.h*. Modul pro správu matic blíže rozebereme v další sekci. K souboru pomocných nástrojů *Tools.h* se již vracet nebudeme, protože funkce v něm obsažené nejsou z hlediska algoritmů a datových struktur zajímavé a tudíž postačí jejich popis v komentářích v souboru přítomných.

Z externích nástrojů jsou ve více programech využity také knihovny projektu SuperMatrix. Protože tento nástroj není široce znám, rozebereme jeho strukturu v samostatné podsekci.

6.1.1 Knihovna SuperMatrix (SM)

SuperMatrix nabízí dva hlavní jmenné prostory. Jsou to *Matrices*, který zastřešuje veškeré operace přímo z řídkými maticemi ve třídě *Matrices::SuperMatrix*, a *smartcomparator*, který zastřešuje porovnávací operace s dvojicemi řádků z objektu matice *Matrices::SuperMatrix*.

Třída *Matrices::SuperMatrix* ukládá kromě hodnot polí řídké matice i popisky řádků a sloupců a také u každého řádku a sloupce ukládá informační údaje počet nenulových hodnot vektoru (řádku nebo sloupce), součet vektoru, globální frekvenci labelu vektoru a entropii. Tyto vlastnosti se budou velmi hodit při počítání metrik kontextu a filtrování.

Jmenný prostor *smartcomparator* poskytuje tzv. komparátory, tedy metody operující s dvojicemi řádků (nebo i sloupců) SM. Tyto metody jsou identifikovány textovým řetězcem, takže je možné je snadno volitelně nastavovat například v konfiguračním souboru.

Podrobnější dokumentace knihovny se nachází v repozitáři SM v adresáři *supermatrix/doc/manual*.

6.1.2 Modul spravující matice kontextu

Matice zatím nijak netransformovaného kontextu se vyznačuje značnou velikostí, která může dosahovat stovek tisíc řádků i sloupců, ale zároveň je velmi řídká. Po provedení transformací se z ní naopak často stává hustá čtvercová, navíc často symetrická matice. Při tom všem je třeba mít u všech těchto druhů matic stejné funkce.

Popsaná situace byla v této práci vyřešena zavedením jedné abstraktní třídy *Matrix*, která poskytuje společné funkčnosti, a matic *SparseMatrix*, *DenseRectMatrix* a *SymmetricRectMatrix*, které z ní dědí a mají svoji vnitřní strukturu, hlavně pro práci s daty, rozdílnou.

Všechny poskytované funkce zde nebudeme rozebírat, jsou dobře zdokumentované ve zdrojovém kódu.

Třída SparseMatrix

V této třídě, dědící z třídy *Matrix*, je spravována řídká matice. Je zbytečné dělat vlastní implementaci, pokud již existuje vysoce kvalitní nástroj v podobě třídy *Matrices::SuperMatrix* z externí knihovny. Proto je třída *SparseMatrix* pouze pouzdrem, zajišťujícím funkcionalitu navazující na rodičovskou třídu.

Třída DenseRectMatrix

V této třídě, dědící z třídy *Matrix*, je spravována hustá čtvercová matice. Data jsou zde uložena v jednom poli typu *double*, jehož délka se rovná kvadrátu rozměru matice, a přistupuje se k nim následující funkcí, která kromě parametrů využívá i údaj o rozměru matice, *rows*.

```
double GetValue(size_t row, size_t col)
{
   return data[row * rows + col];
}
```

Třída SymmetricRectMatrix

V této třídě, dědící z třídy *Matrix*, je spravována hustá čtvercová symetrická matice. Data jsou zde uložena opět v jednom poli typu *double*, jehož délka odpovídá výrazu $rows * (rows + 1)/2$, kde *rows* je rozměr matice. K prvkům matice se přistupuje pomocí následujícího kódu.

```
size_t GetArrayIndex(size_t r, size_t c)
{
   size_t m = min(r,c);
   return (r*r + c*c - m*m + r + c + m) / 2;
}

double GetValue(size_t row, size_t col)
{
   return data[GetArrayIndex(row, col)];
}
```

6.2 Moduly tvořící program FeatureRetriever

FR využívá dva hlavní moduly. Jeden z nich, *Matrix* spravuje matice kontextu a je popsán výše. Druhý, *MatrixProcessor*, provádí transformace těchto matic a je popsán v následujících sekcích.

6.2.1 Metody transformující matice

Metody transformace matic se dělí do tří skupin. První skupinu lze na jedné matici počítat paralelně, navíc je výstupní matice symetrická. Druhou skupinu lze také na jedné matici počítat paralelně, ale výstupní matice symetrická není. Třetí skupinu tvoří metody, jejichž paralelizace na jedné matici není nutná, nebo jsou tak nenáročné, že není potřeba. První dvě skupiny získají z třídy *MatrixProcessingMethods* metodu, kterou pak aplikují buď na všechny kombinace dvojic řádků (nesymetrická metoda, výsledkem je nesymetrická matice), nebo pouze na množinu všech různých neuspořádaných dvojic, řádků, což je přibližně polovina předchozího počtu.

Metody, které jsou vydávané třídou *MatrixProcessingMethods* jsou buďto interní metody ze třídy *MatrixProcessingMethods*, nebo se jedná o zapouzdřené komparátory ze jmenného prostoru projektu *compare::smartcomparator* projektu *SuperMatrix*. Volání těchto metod je dobře popsané v dokumentaci tohoto projektu. Metody počítané přímo uvnitř třídy *MatrixProcessingMethods* jsou dokumentované ve zdrojovém kódu.

Pokud paralelizace není třeba, nebo není možná, jsou parametry metody spolu se vstupní maticí předány třídě *MatrixProcessingMethods*, ve které je vypočten výsledek a vrácena transformovaná matice.

6.3 Moduly tvořící program EvaluateFeatures

EF je tvořený modulem pro práci s daty, *DataSet*, modulem obsluhujícím strojové učení *MachineLearner* a modulem vyhodnocujícím úspěšnost predikce *ResultEvaluator*.

Modul *DataSet* na začátku dostane všechny vstupní rysy a seznam relací extrahovaný z WordNetu. Podle požadavku z nich vytvoří dataset buď s cílovou třídou z WN, nebo bez ní, kde se instance nebudou krýt s relacemi ve WN.

Dále pak na vyžádání vrací objekt typu *std::vector* z STL obsahující zvolené řádky datasetu. Například i–té trénovací a testovací sady pro křížovou validaci.

Modul *MachineLearner*, dostává datasety z modulu *DataSet* a podle toho, jestli běží v režimu predikce, nebo evaluace spouští proces predikce, nebo křížové validace výsledků. Knihovna, která tomuto modulu poskytuje metody strojového učení je *liblinear*, nicméně modul je navržen tak, aby bylo možné tuto knihovnu snadno nahradit jinou.

Poslední modul v řadě procesu EF je *ResultEvaluator*, který dostane seznam dvojic (predikce, cílová třída) a vypíše statistiky výsledků. Součástí tohoto modulu je i třída pracující s *confusion matrix*.

6.4 Moduly tvořící program WN-Transformer

WNT využívá pouze modul sémantické sítě, *semantic_network*, který zpracovává graf sítě, tvořený relacemi a lematy extrahovanými z WN modulem *word_net*. Struktura tříd modulu *word_net* odpovídá struktuře EWN, modul obsahuje navíc jen zastřešující třídu.

Z modulu *word_net* jsou do modulu *semantic_network* načítány relace a lemmata. Pro obojí jsou v modulu *semantic_network* připravené specializované třídy *lemma* a *relation*, dědící od nadtřídy *entity*.

Pro organizaci a unifikaci těchto entit, byl použit modul poskytující binární vyhledávácí strom AVL.

6.4.1 Modul binárního vyhledávacího stromu AVL

Z důvodů potřeby rychlého třídění a unifikace relací a lemmat, tedy entit z modulu WordNet, byl použit binární vyhledávací strom. Aby bylo využití stromu efektivní, byla zvolena variantu AVL stromu. AVL strom je dynamickou strukturou, která využívá rozdílu hloubky podstromů jednotlivých uzlů k vlastnímu vyvážení. Pro naše potřeby byla vytvořili vlastní generická implementaci. Protože množství ukládaných dat relací sahá do mnoha desítek tisíců, bylo třeba využít co nejjednodušší a zároveň nejefektivnější řešení. Nejsme schopni dosáhnout lepší úspěšnosti při vyhledávání než *log N*, proto je implementace v binárním stromě optimální.

Náš strom má v každém uzlu uloženu klíčovou hodnotu, odkaz na ukládaný objekt, odkazy na pravý a levý podstrom a hloubku. Při přidávání uzlů provádíme vyhledání místa pro uložení, a pokud jeden z upravených podstromů zvýší svou hloubku oproti druhému podstromu o více než jedna, provádíme rotaci. Využíváme dvou typů rotací, jednoduchou, která je buď pravá či levá, nebo dvojitou rotaci. Rotace srovná hloubky podstromů, čímž docílíme efektivní implementace našeho stromu.

7. Závěr

Postup extrakce sémantických informací prezentovaný v této práci je zřejmě první, který byl využit pro češtinu. Zároveň se jedná o první využití přístupu strojového učení s učitelem k extrakci sémantických informací vůbec.

Byly zkoumány různé možnosti načítání modelu kontextu, filtrování vzniklých matic, posloupnosti transformací a nakonec vyhodnocení i za pomoci ruční anotace třemi nezávislými anotátory. Byla navržena vysoce konfigurovatelná metoda získávání sémantických vztahů. Zároveň byla prezentována sada nástrojů pro automatizaci celého procesu extrakce.

Na konci obou kapitol popisujících použité postupy jsou uvedeny další možné cesty zlepšování výsledků. Práce tedy může být inspirací pro další komputační lingvisty. Možných způsobů vylepšení celého procesu bylo navrženo hned několik, což by mohlo naznačovat, že kvalita současných výsledků je nízká. Tak to ale není, jak dokládají hodnoty metriky automatického hodnocení kvality sad rysů i výsledky ručního hodnocení. Na jednu stranu jsou procenta úspěšnosti získaných konkrétních relací nízká, na druhou ale po zobecnění na měření míry *similarity* a *semantic relatedness* převyšují procenta ručního hodnocení relací v Czech WordNetu, která by měla být horní možnou hranicí dosažitelného.

V diskusi výsledků je vyslovena hypotéza proč by tomu tak mohlo být. Pravděpodobnou příčinou je, že slova v CWN jsou anotovaná často ve svých minoritních významech a po oproštění od značky významu (tj. vytržení z kontextu, ze synsetu) je pro anotátora obtížné jejich význam v relaci z CWN určit. Význam lemmatu z CWN by bylo možné upřesnit lemmaty, která se nacházejí ve stejném synsetu, takové upřesnění ale není možné použít u dvojic lemmat, získaných automaticky, proto nebyl tento údaj anotátorům dán k dispozici.

Automatická metoda extrakce naopak reaguje na distribuci modelu kontextu slova, tedy (podle distribuční hypotézy) na všechny významy slova současně a nejvíc je ovlivněná právě významem většinovým. To je důvod, proč byla úspěšnost metody, počítaná optikou *similarity* a *relatedness*, lepší, než relací z CWN. Tímto byla dokázána relevantnost použitých hypotéz a na nich založených postupů.

Kromě již zmíněných závěrů má tato práce ještě další přínosy, a to následujících několik zjištění (obrázky 4.3, 4.4 a 4.5).

1. Použití metody strojového učení s učitelem na dataset tvořený více kontextovými rysy přináší významné zlepšení.

2. Sekce 4.1.4 dokládá, že umocnění matice rysu metrikou podobnosti, aplikovanou na jeho řádky má smysl a dodává další informace.

3. V sekci 4.1.4 bylo také ukázáno, že vhodné metriky pro extrakci hyperonym jsou *Distance*, *Lin* a metriky aplikované na matici modelu kontextu, načteného ze syntaktické úrovně vět. Pro extrakci synonym se na druhou stranu hodí rysy získané z modelů kontextu z povrchové struktury věty.

4. Jen jedna použitá asymetrická míra (pokrytí) na rozpoznání směru relací nestačí. K vylepšení výsledků získávání orientovaných relací je třeba nalézt nějaké další.

5. Czech WordNet se v současné podobě s velkými synsety pro trénování popsané metody extrakce sémantických relací nezdá vhodný.

Součástí vyhotovené práce je CD, obsahující anotovaná data, záznamy jednotlivých pokusů a v neposlední řadě i již zmiňovaná a popisovaná sada programů a dávkových souborů, které mohou být využity pro další výzkum.

Literatura

[1] *Český národní korpus - SYN2000*. Praha, Czech Republic: Ústav Českého národního korpusu FF UK, 2000.

[2] *Český národní korpus - SYN2005*. Praha, Czech Republic: Ústav Českého národního korpusu FF UK, 2005.

[3] *Český národní korpus - SYN2006PUB*. Praha, Czech Republic: Ústav Českého národního korpusu FF UK, 2006.

[4] Agirre, E.; Alfonseca, E.; Hall, K.; aj.: A study on similarity and relatedness using distributional and WordNet-based approaches. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics, 2009, s. 19–27.

[5] Andrews, P.: *Semantic topic extraction and segmentation for efficient document visualization*. Lausanne, Switzerland: School of Computer and Communication Sciences, Swiss Federal Institute of Technology, 2004.

[6] Bejček, E.; Möllerová, P.; Straňák, P.: The Lexico-Semantic Annotation of PDT: Some Results, Problems and Solutions. In *TSD*, 2006, s. 21–28.

[7] Broda, B.; Piasecki, M.: SuperMatrix: a General tool for lexical semantic knowledge acquisition. In *IMCSIT*, 2008, s. 345–352.

[8] Budanitsky, A.; Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 2006: s. 13–47.

[9] Chandna, S.: *Comparative analysis of measures of similarity and semantic relatedness for text classification*. Patiala, Paňdžáb, Indie: Computer Science and Engineering Department, Thapar University, 2010.

[10] Chang, C.-C.; Lin, C.-J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[11] Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; aj.: LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 2008: s. 1871–1874.

[12] Fellbaum, C. (editor): *WordNet An Electronic Lexical Database*. Cambridge, MA ; London: The MIT Press, 1998.

[13] Gentzsch, W.: Sun Grid Engine: Towards Creating a Compute Power Grid. In *CCGRID*, IEEE Computer Society, 2001, s. 35–39.

[14] Hajič, J.; Panevová, J.; Hajičová, E.; aj.: *Prague Dependency Treebank 2.0*. Philadelphia, PA, USA: Linguistic Data Consortium, 2006.

[15] Harris, Z.: *Mathematical structures of language*. Interscience Publishers, 1968.

[16] Hearst, M. A.: Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, COLING '92, Nantes, France: Association for Computational Linguistics, 1992, s. 539–545.

[17] Hüllen, W.: *A History of Roget's Thesaurus: Origins, Development, and Design*. 2005.

[18] Kohavi, R.; Provost, F.: *Glossary of Terms*. 1998.

[19] Landauer, T. K.; Dutnais, S. T.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 1997: s. 211–240.

[20] Lenat, D.: CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 1995: s. 33–38.

[21] Lin, D.: Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, Morristown, NJ, USA: Association for Computational Linguistics, 1998, s. 768–774.

[22] Mitchell, T. M.: *Machine Learning*. New York: McGraw-Hill, 1997.

[23] Pajas, P.; Štěpánek, J.: Recent Advances in a Feature-Rich Framework for Treebank Annotation. In *The 22nd International Conference on Computational Linguistics - Proceedings of the Conference*, 2008, s. 673–680.

[24] Pajas, P.; Štěpánek, J.: Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, Manchester, United Kingdom: Association for Computational Linguistics, 2008, s. 673–680.

[25] Patwardhan, S.; Pedersen, T.: Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006 Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together*, Trento, Italy, 2006, s. 1–8.

[26] Pecina, P.: An Extensive Empirical Study of Collocation Extraction Methods. In *ACL*, 2005.

[27] Petkevič, V.: Reliable Morphological Disambiguation of Czech: Rule-Based Approach is Necessary. In *Insight into the Slovak and Czech Corpus Linguistics*, editace M. Šimková, Bratislava, Slovakia: Veda, 2006, s. 26–44.

[28] Piasecki, M.: Automated Extraction of Lexical Meanings from Corpus: A Case Study of Potentialities and Limitations. In *Representing Semantics in Digital Lexicography. Innovative Solutions for Lexical Entry Content in Slavic Lexicography. MONDILEX Fourth Open Workshop*, Warszawa, Poland: Institute of Slavic Studies, Polish Academy of Sciences, 2009.

[29] Pála, K.; Smrž, P.: Building Czech Wordnet. ročník 2004, 2004: s. 79–88.

[30] Richardson, S. D.; Dolan, W. B.; Vanderwende, L.: MindNet: acquiring and structuring semantic information from text. In *Proceedings of the 17th international conference on Computational linguistics*, Morristown, NJ, USA: Association for Computational Linguistics, 1998, s. 1098–1102.

[31] Schütze, H.: Automatic Word Sense Discrimination. *Computational Linguistics*, 1998: s. 97–123.

[32] Vanderwende, L.; Kacmarcik, G.; Suzuki, H.; aj.: MindNet: An Automatically-Created Lexical Resource. In *HLT/EMNLP*, 2005.

[33] Vossen, P. (editor): *EuroWordNet: a multilingual database with lexical semantic networks*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[34] Yule, G.; Kendall, M.: *An introduction to the theory of statistics*. London: Griffin, 1950.

Seznam tabulek

Seznam obrázků

A. Seznam použitých zkratek

- **CWN** (Czech WordNet) – Česká část EuroWordNetu, evropského WordNetu.
- **ČNK** (Český Národní Korpus) — Rozsáhlý korpus českého jazyka.
- **EF** (EvaluateFeatures) — Program pro predikci relací, součást práce.
- **FR** (FeatureRetriever) — Program pro získávání rysů, součást práce.
- **EWN** (Euro WordNet) — Vícejazyčný evropský WordNet.
- **LSA** (Latent Semantic Analysis) — Technika transformace matice kontextu.
- **NLP** (Natural Language Processing) – Obor počítačové zpracování přirozeného jazyka.
- **PDT** (Prague Dependency Treebank) – Pražský závislostní korpus.
- **PlWN** (Polish WordNet) — Polská část Euro WordNetu.
- **RMSD** (Root Mean Square Deviation) – Odmocněná střední kvadratická chyba.
- **SGE** (Sun Grid Engine) — Prostředí pro spoučtšní úloh na výpočetním clusteru.
- **SM** (SuperMatrix) — Nástroj pro práci s maticemi kontextu.
- **STL** (Standard Template Library) — Standatdní knihovna jazyka C++, která nabízí řadu užitečných datových struktur.
- **SVD** (Singular Value Decomposition) — Rozklad matice na tři určitých druhů, matematický základ LSA.
- **SVM** (Support Vector Machines) — Metoda strojového učení použitá v práci.
- **WN** (WordNet) — Anglicko-jazyčný sémantický zdroj.
- **WNT** (WN-Transformer) — Program pro získávání relací z WordNetu, součást práce.

B. Obsah přiloženého CD

```
+-/calculated_data:
| |   -Adresář obsahující mezivýpočty a logy výpočtů.
| |   -Soubory synonyms a hypernyms, obsahující všechny nové relace
| +-/calculated_data/annotation:
| |     -Adresář obsahující soubory s ruční anotací relací na
| |      kterých bylo provedeno vyhodnocení
| +-/calculated_data/feature_evaluation:
|       -Zde jsou uloženy záznamy z běhu výběru nejlepších
|        rysů i se záznamem úspěšností
+-/software:
| |   -Adresář obsahující softwarové nástroje a nastavení
| |   -Zde je také umístěný instalační skript install.sh
| +-/software/cfg:
| | |   -Zde se nachází soubory s nastavením pro program
| | |    FeatreRetriever
| | +-/software/cfg/wn_transformer:
| |     -Zde se nachází nastavení programu WN-Transformer.
| +-/software/lists:
| |   -Seznamy zakázaných lemmat a povolených tagů
| +-/software/misc:
| |   -Pomocné soubory
| +-/software/scenarios:
| | |   -Adresář, ve kterém jsou uložené skripty načítající
| | |    kontext z korpusů, konstruují z něj
| | |    matice a provádí s nimi naplánované operace
| | +-/software/scenarios/evaluate_features:
| | |   -Zde je uložený skript provádějící hladový výběr rysů
| | +-/software/scenarios/extract_relations:
| |     -Zde je uložený skript extrahující nové relace pomocí
| |      vybraných rysů
| +-/software/scripts:
| | | -Adresář, ve kterém jsou uložené pomocné a konverzní skripty
| | +-/software/scripts/btred:
| |     -Adresář obsahující dávkové soubory programu btred
| +-/software/src:
|     -Adresář obsahující zdrojové soubory EF, TWN a FR
+-/text:
      -Zde je uložen tento text ve formátu pdf
```

Pro začátek práce je třeba sestavit programy EF, TWN a FR pomocí

`/software/install.sh`

Poté upravit cesty ve všech dávkových souborech tak, aby vyhovovaly prostředí spouštění. Nastavení proměnných se vždy nachází na začátku skriptu.

C. Seznam úspěšně predikovaných relací

Na následujících relacích predikovaných EF se shodli alespoň dva anotátoři.

Pořadí	První lemma	Druhé lemma	Anot. A	Anot. B	Anot. C	Predikce
1	alkoholizmus	narkotikum	rel	rel	rel	hyp
2	angličtina	němčina	sim	sim	sim	hyp
3	b	c	rel	sim	sim	hyp
4	banka	spořitelna	sim	syn	syn	syn
5	barva	odstín	syn	sim	syn	syn
6	březen	září	sim	sim	sim	hyp
7	c	b	rel	sim	sim	syn
8	ČD	dráha	rel	syn	rel	syn
9	červen	duben	sim	sim	sim	syn
10	červen	srpen	sim	sim	sim	syn
11	člověk	bůh	rel	sim	sim	syn
12	člověk	bytost	hyp	syn	hyp	syn
13	člověk	duch	rel	sim	rel	syn
14	člověk	duše	rel	sim	rel	syn
15	člověk	láska	rel	rel	rel	syn
16	člověk	muž	hyp	hyp	hyp	syn
17	člověk	myšlenka	hyp	rel	rel	syn
18	člověk	příroda	rel	sim	rel	syn
19	člověk	vědomí	rel	sim	rel	syn
20	člověk	vůle	rel	sim	rel	syn
21	člověk	žena	hyp	hyp	hyp	syn
22	člověk	život	rel	rel	rel	syn
23	ČR	ČSFR	rel	sim	sim	hyp
24	den	březen	rel	rel	sim	syn
25	den	červenec	rel	sim	sim	syn
26	den	doba	rel	sim	sim	syn
27	den	duben	rel	rel	sim	syn
28	den	leden	rel	rel	sim	syn
29	den	listopad	rel	rel	sim	syn
30	den	měsíc	hyp	sim	sim	syn

Pořadí	První lemma	Druhé lemma	Anot. A	Anot. B	Anot. C	Predikce
41	dolar	měna	hyp	hyp	hyp	syn
31	den	pátek	sim	hyp	hyp	syn
32	den	prosinec	rel	rel	sim	syn
33	den	rok	hyp	sim	sim	syn
34	den	říjen	rel	rel	sim	syn
35	den	srpen	rel	rel	sim	syn
36	den	září	rel	rel	sim	syn
37	disk	gigabyte	rel	sim	rel	syn
38	doba	začátek	rel	sim	rel	syn
39	dolar	koruna	sim	sim	sim	hyp
40	dolar	marka	sim	sim	sim	syn
41	dolar	měna	hyp	hyp	hyp	syn
42	dům	byt	sim	sim	sim	syn
43	dur	moll	sim	sim	sim	syn
44	federace	ČSFR	rel	hyp	rel	hyp
45	finále	čtvrtfinále	rel	sim	sim	hyp
46	finále	semifinále	rel	rel	sim	hyp
47	firma	podnik	syn	syn	syn	hyp
48	firma	společnost	syn	syn	syn	hyp
49	foto	rámeček	rel	sim	rel	hyp
50	galerie	muzeum	rel	syn	syn	hyp
51	gól	branka	rel	syn	rel	hyp
52	infekce	skvrnitost	hyp	rel	rel	syn
53	jednání	zasedání	rel	syn	syn	hyp
54	komise	výbor	syn	sim	syn	hyp
55	konec	polovina	rel	sim	sim	syn
56	konec	začátek	sim	sim	sim	syn
57	koruna	marka	sim	sim	sim	hyp
58	koruna	miliarda	rel	rel	sim	syn
59	koruna	milión	hyp	rel	rel	syn
60	květen	březen	sim	sim	sim	syn
61	květen	červen	sim	sim	sim	syn
62	květen	červenec	sim	sim	sim	syn
63	látka	sloučenina	syn	sim	sim	syn
64	léčba	farmakoterapie	syn	sim	syn	syn
65	letadlo	letoun	syn	syn	syn	hyp
66	meniskus	vaz	rel	rel	hyp	hyp
67	metr	kilometr	sim	sim	sim	hyp
68	miliarda	milión	sim	sim	sim	syn
69	milión	miliarda	rel	rel	sim	hyp
70	milión	sto	rel	rel	sim	hyp

Pořadí	První lemma	Druhé lemma	Anot. A	Anot. B	Anot. C	Predikce
41	dolar	měna	hyp	hyp	hyp	syn
71	milión	tisíc	rel	sim	sim	hyp
72	ministerstvo	ministr	rel	rel	rel	hyp
73	ministr	ministerstvo	rel	sim	rel	syn
74	ministr	premiér	rel	sim	sim	hyp
75	místopředseda	předseda	sim	sim	sim	hyp
76	mnich	cisterciák	syn	hyp	hyp	hyp
77	móda	bižuterie	rel	rel	rel	hyp
78	monarchie	císař	rel	rel	rel	hyp
79	music	folk	rel	hyp	hyp	syn
80	muž	žena	sim	sim	sim	syn
81	mzda	plat	syn	syn	syn	hyp
82	mzda	výdělek	syn	syn	syn	hyp
83	navazování	navázání	sim	syn	syn	hyp
84	návrh	schválení	rel	rel	rel	syn
85	noha	ruka	rel	sim	sim	syn
86	období	rok	sim	hyp	hyp	syn
87	obličej	tvář	syn	syn	syn	syn
88	obligace	dluhopis	sim	sim	syn	syn
89	odumírání	vadnutí	hyp	syn	syn	syn
90	onemocnění	choroba	rel	syn	syn	hyp
91	ostrov	souostroví	sim	hyp	sim	syn
92	otec	matka	rel	sim	sim	syn
93	paragraf	ustanovení	syn	syn	syn	syn
94	pátek	čtvrtek	sim	sim	sim	hyp
95	pátek	středa	sim	sim	sim	syn
96	pivo	půllitr	rel	syn	rel	syn
97	plán	plánování	syn	syn	rel	syn
98	platforma	server	rel	sim	sim	syn
99	plyn	elektřina	rel	rel	sim	hyp
100	počátek	doba	rel	rel	sim	syn
101	počátek	konec	rel	sim	sim	syn
102	pojištění	pojišťovna	rel	rel	rel	hyp
103	pokles	vzestup	rel	rel	sim	syn
104	polopravda	nepravda	rel	sim	sim	hyp
105	prezident	ministr	rel	sim	sim	hyp
106	předpis	zákon	syn	syn	syn	syn
107	příjem	mzda	syn	syn	syn	hyp
108	půl	sto	rel	rel	sim	hyp
109	pupen	výhon	syn	sim	sim	syn
110	republika	ČR	sim	hyp	hyp	hyp

Pořadí	První lemma	Druhé lemma	Anot. A	Anot. B	Anot. C	Predikce
111	rodič	dítě	rel	sim	rel	syn
112	rok	doba	sim	hyp	sim	syn
113	rok	leden	rel	rel	sim	syn
114	rok	listopad	rel	rel	sim	syn
115	rok	prosinec	rel	rel	sim	syn
116	rok	týden	rel	rel	sim	syn
117	rok	únor	rel	rel	sim	syn
118	rok	září	rel	sim	sim	syn
119	ruka	rameno	rel	sim	rel	syn
120	růst	nárůst	sim	syn	syn	syn
121	růst	pokles	rel	sim	sim	syn
122	řešení	vyřešení	rel	syn	syn	hyp
123	řidič	řidička	syn	syn	sim	syn
124	říjen	září	sim	sim	sim	hyp
125	sklo	keramika	rel	sim	sim	hyp
126	skvrnka	skvrna	syn	syn	syn	syn
127	smlouva	dohoda	syn	syn	syn	hyp
128	snímek	film	syn	hyp	syn	syn
129	snížení	snižování	rel	syn	syn	syn
130	snížení	zvýšení	rel	sim	sim	hyp
131	sobota	neděle	sim	sim	sim	hyp
132	společnost	akcie	rel	sim	rel	syn
133	společnost	akcionář	rel	sim	rel	syn
134	společnost	firma	rel	syn	syn	syn
135	společnost	podnik	syn	syn	syn	hyp
136	společnost	společník	hyp	hyp	rel	syn
137	spravedlnost	vnitro	rel	sim	sim	hyp
138	systém	server	rel	sim	rel	syn
139	telefon	fax	rel	sim	sim	hyp
140	transakce	obchod	syn	syn	rel	syn
141	trenér	kapitán	rel	sim	sim	hyp
142	trh	investor	rel	sim	rel	syn
143	umění	umělec	rel	sim	rel	syn
144	univerzita	ČVUT	rel	hyp	hyp	syn
145	ustanovení	odstavec	rel	rel	rel	syn
146	utkání	střetnutí	rel	syn	syn	hyp
147	utkání	zápas	syn	syn	syn	hyp
148	vláda	parlament	rel	rel	hyp	hyp
149	vlak	rychlík	hyp	syn	hyp	syn

Pořadí	První lemma	Druhé lemma	Anot. A	Anot. B	Anot. C	Predikce
150	vodovod	kanalizace	rel	sim	rel	hyp
151	výrobek	zboží	hyp	hyp	hyp	hyp
152	výstava	veletrh	sim	syn	syn	syn
153	vývoz	dovoz	rel	sim	sim	hyp
154	vývoz	export	syn	syn	syn	hyp
155	zákon	ustanovení	syn	syn	syn	syn
156	zákon	zákoník	rel	hyp	rel	syn
157	zápas	střetnutí	rel	syn	syn	hyp
158	září	červen	sim	sim	sim	syn
159	září	červenec	sim	sim	sim	syn
160	září	srpen	sim	sim	sim	syn
161	zelenina	ovoce	rel	sim	sim	hyp
162	země	stát	rel	syn	syn	hyp
163	zranění	poranění	syn	syn	syn	hyp
164	zvýšení	snížení	rel	sim	sim	syn
165	zvýšení	zvyšování	sim	syn	syn	hyp