

MASARYKOVA UNIVERZITA



Filozofická fakulta

Ústav českého jazyka

Český jazyk se specializací počítačová lingvistika

David Klement

Vizualizace sémantické sítě

Magisterská diplomová práce

Vedoucí práce: RNDr. Adam Rambousek, Ph.D.

2017

Obsah

Úvod	3
I Sémantické sítě	4
1 Princetonský WordNet	6
1.1 Motivace vzniku	6
1.2 K vlivu psycholingvistiky na organizaci WordNetu	9
1.3 Organizace WordNetu	11
1.4 Sémantické vztahy WordNetu	13
1.5 Lexikální vztahy ve Wordnetu	15
2 Další wordnety	19
2.1 EuroWordNet	19
II Přehled a porovnání existujících vizualizací sémantických sítí	23
3 Metodologie porovnání	25
3.1 Výběr rozhraní	25
3.2 Strukturalizace přehledu a kritéria hodnocení	27
3.3 Podmínky testování	28
3.4 WordNet Search jako základ porovnání	28
4 Vizualizace s webovým rozhraním	31
4.1 An interactive visualization of the Princeton WordNet database	31
4.2 WordNET Editor	33
4.3 Cornetto Demo	34
4.4 WordVis	38
4.5 sloWTool	40
4.6 BabelNetXplorer	43

OBSAH	2
5 Vizualizace v aplikačním prostředí Java	46
5.1 wnbrowser	47
5.2 Treebolic	48
5.3 VisualBrowser	49
6 Desktopové aplikace	51
6.1 Artha	52
6.2 GoldenDict	53
7 Shrnutí přehledu rozhraní a vyplývající závěry o vhodném rozhraní k sémantickým sítím	55
 III Praktická část: rozhraní k sémantické síti	 57
8 Návrh rozhraní	58
8.1 Východiska	58
8.2 Cíle nového rozhraní	59
9 Prezentace vytvořeného rozhraní	60
9.1 Základní představení	60
9.2 Textová reprezentace	61
9.3 Grafická reprezentace	62
10 Použité technologie	64
10.1 HTML	64
10.2 JavaScript	65
10.3 CSS	68
11 Implementace rozhraní	70
11.1 Získávání dat	71
11.2 Zpracování odpovědi	73
11.3 Zobrazení dat	73
11.4 Uchovávání stavu v URL	75
12 Možnosti dalšího vývoje	77
Závěr	79
Bibliografie	86

Úvod

Tato práce se zabývá sémantickými sítěmi typu wordnet především z pohledu lidského uživatele. Sémantická síť princetonský WordNet a další sítě z ní vycházející byly projektovány se záměrem napomoci zejména počítačovému zpracování přirozeného jazyka. Díky své struktuře založené na psycholingvistických poznatcích jsou však potenciálně užitečné i pro lidského uživatele. Tento potenciál společně s vědními základy, které se staly formanty struktury wordnetů, práce rozebere ve své části I.

Aby však sémantická síť byla pro lidského uživatele použitelná, je nutné umožnit k ní přístup pomocí rozhraní, jež zájemci zobrazí relevantní data. Takových rozhraní bylo vytvořeno za dobu vývoje sémantických sítí značné množství; vytvořená rozhraní se liší uživateli, na něž jsou zaměřena, platformami, které podporují či kupříkladu způsoby prezentace dat wordnetů. Jejich společným rysem však bývá jistá neaktuálnost návrhu uživatelského rozhraní, a to především v kontextu jednoho z trendů poslední dekády, kterým je přesun uživatelů z klasických stolních počítačů a laptopů na mobilní zařízení [37].

V části II se práce bude věnovat právě těmto existujícím rozhraním, rozebere jejich funkcionalitu, positiva a negativa a zhodnotí jejich přínos a použitelnost pro koncového uživatele. Hodnocená rozhraní je nutno brát pouze jako zástupce většího množství dostupných rozhraní (a možná ještě většího těch nedostupných), jelikož cílem této práce není představit co nejúplnější přehled existujících rozhraní, ale spíše ukázat menší množství těch relativně snadno dostupných a podrobněji je rozebrat. Cílem této části je identifikovat nejvýraznější nedostatky těchto rozhraní a vyvodit z jejich zhodnocení některé vlastnosti, kterými by se mělo rozhraní z pohledu současnosti moderní vyznačovat.

Praktickým cílem této práce pak bylo vytvořit rozhraní nové, jehož návrh by vycházel z empiricky získaných poznatků při hodnocení existujících rozhraní v části II. V rámci práce bylo takové nové rozhraní vytvořeno, přičemž podrobnějšímu popisu výsledku je věnována III. část tohoto textu.

Část I

Sémantické sítě

Sémantické sítě, neboli wordnety, jsou lexikální databáze vytvořené s rozličnými zá-
měry, mezi něž patří například i strojová inference informací v počítačovém zpracování
přirozeného jazyka. Ve wordnetech jsou slova slučována podle významů do synonymic-
kých okruhů a tyto okruhy jsou propojovány sémantickými vztahy [46, 49], čímž celek do-
stává svému označení sémantická síť.

Kapitola I

Princetonský WordNet

Princetonský WordNet je prvním wordnet vůbec. Vznikal na univerzitě v Princetonu pod G. A. Millerem od poloviny 80. let 20. století. Vzhledem k tomu, že byl prvním wordnetem, bylo k němu referováno jako k WordNetu, bez přívlastku. Ačkoliv tento stav v podstatě přetrvává dodnes, oproti době jeho vzniku se situace změnila, vzniklo několik dalších wordnetů a nastala tudíž potřeba je rozlišit. V anglickém prostředí se obvykle pojmem wordnet míní ten princetonský a všechny ostatní wordnety mají přívlastek či vlastní jméno. Příkladem necht' jsou evropské wordnety BalkaNet či EuroWordNet¹. Ačkoliv v mezinárodním prostředí je obvyklé přívlastek „princetonský“ používat, bude tato práce pracovat s následujícím rozlišením:

- *WordNet* (ve významu princetonského WordNetu)
- *wordnet* (v obecném významu sémantické sítě založené na WordNetu)
- *BalkaNet* (ve významu konkrétního wordnetu, např. *BalkaNet* či *wordnet EuroWordNet*)

Motivace vzniku

Od počátků snah o zpracování přirozeného jazyka² bylo nutné poskytnout programu data o lexiku ve zpracovávaném textu, ať už ona data byla jakákoliv. Kupříkladu v začátcích strojového překladu se mělo za to, že stačí ekvivalentní dvojice slov ve zdrojovém a cílovém jazyce, později byla implementována i pravidla, jež měla eliminovat problémy s pořádkem slov ve větách, ale výsledné překlady stále nebyly po kvalitativní stránce uspokojivé, neboť vyžadovaly značné množství editací po zpracování automatickým systémem. [62]

Tradičně se lexikální materiál ukládal způsobem nikoliv diametrálně odlišným od páirových slovníků určených pro lidské uživatele. Ty obvykle obsahují lexikální záznamy

¹O jiných wordnetech než tom princetonském více v dalších kapitolách

²angl. natural language processing, zkráceně NLP

seřazené podle abecedy a s potřebnými informacemi o daných slovech, z nichž pak program může čerpat při zpracování textu.

Jak uvádí Pala a Ševeček [46], uspořádání lexikálního materiálu v formě, v jaké je v tradičních slovnících, je sice vhodné pro člověka, jelikož je to v podstatě jediný způsob, jak lidskému uživateli umožnit vyhledání konkrétního slova, zvláště nezná-li jeho význam, pro strojové zpracování ale je ale takto zpracovaný slovník slovník nedostatečný. Seřazení indexu slovních forem je sice pro efektivní vyhledávání slov důležité, struktura tradičního slovníku však neposkytuje vhodně zakódované informace o příbuznosti slov. Jednak kvůli onomu abecednímu řazení inherentně vzdaluje slova, jež člověk chápe jako nějakým způsobem blízká [46], jednak neobsahuje dodatečné vazby mezi příbuznými slovy (nebo je obsahuje ve formě nevhodné pro strojové zpracování). Významová blízkost může vyplývat ze vztahu volné synonymie, antonymie, podřazenosti, nadřazenosti, etc. Pokud si tedy například uživatel výkladového slovníku nepříliš obeznámený s daným jazykem vyhledá určité heslo, dozví se sice pravděpodobně jeho význam, ale nebude schopen své znalosti prohlubovat dále kupříkladu zjištěním, jaké slovo odpovídá opačnému významu.

Dalším všeobecným problémem při využití tradičních slovníků k počítačovému zpracování jazyka je fakt, že lexikografové předpokládají u uživatele slovníku značné encyklopedické znalosti. Zařazují tak do slovníku jen informace dle jejich názoru důležité pro zařazení dané entity do příslušné nadřazené třídy (genus proximum) a rozlišení dané entity v nadřazené třídě (differentia specifica). Vyhledá-li si tedy člověk ve Slovníku spisovného jazyka českého heslo vlk, zjistí následující:

vlk: psovitá šelma šedě (n. šedožlutě) zbarvená, žijící v Evropě, Asii a v Sev. Americe

Definice a priori předpokládá, že uživatel je obeznámen s tím, co je šelma a co je pes. Pokud takovou znalostí neslyne (což je vcelku představitelné například u cizince), je nucen si tato slova ve slovníku najít a podívat se na jejich definice (pomiňme nyní netriviální úkol převést slovo psovitá na základní tvar pes). Pokud nerozumí definicím ani nadřazených slov, musí pokračovat v hierarchii dále a dále.

Z uvedeného případu plyne, že jakkoliv je možné správným vyhledáváním hyperonym³ dospět k tomu, že vlk je konkrétní entita našeho vesmíru, živá bytost o čtyřech končetinách, savec nějakým způsobem příbuzný se psovi, má šedou srst etc., je takový proces dosti komplikovaný. Případ s cizincem se sice nemusí zdát zcela relevantní, protože se dá předpokládat, že daný člověk má, byť v jiném jazyce, stejné základní znalosti předpokládané lexikografy jako člověk, jehož mateřštinou je čeština. Situace je však dramaticky jiná u počítače

kurva fix,
si snad
budu
muset
dojit do
knihovny,
abych to
ocitoval

³nadřazené slovo

(přesněji u počítačového programu). Na rozdíl od člověka totiž počítač nemá žádné předchozí znalosti, tudíž musí projít celým procesem popsaným výše, aby byl schopen kupříkladu určit, že vlk může umřít (ježto je živá bytost). Protože však tradiční slovníky typu SSJČ byly vytvářené pro papírové médium, neobsahují žádné propojení ve stylu *toto je odkaz na hyperonymum*, a počítač tudíž jen těžko může zjišťovat, na které vlastně slovo se to má podívat, aby se dobral podstaty pojmu vlk.

Strojově čitelné slovníky

V zájmu automatizace vyhledávání ve slovníku vznikaly tzv. strojově čitelné slovníky⁴, což je pojem souhrnně označující lexikální databáze. Podle množství informací, které taková databáze obsahuje, pak lze tyto dělit na slovníky, taxonomie a ontologie. Běžně obyčejný slovník neobsahuje oproti tradičnímu papírovému slovníku navíc žádné metainformace, takže je počítač při jeho užívání v podstatě omezen na elektronický listovač [38].

Míru, jakou se strojově čitelný slovník odlišuje od pouhé zdigitalizované formy papírového slovníku a přibližuje se k pokročilé lexikální databázi, lze vyjádřit v několika stupních. V případě, že slovník má jednotlivé významy uspořádány v hierarchii dle nadřazenosti–podřazenosti, lze jej označit za taxonomii, tedy systém s hlubší strukturou než pouze abecedním řazením hesel.

Dalším stupněm je již komplexní lexikální databáze, která má jednotlivé významy propojeny rozličnými vztahy, počínaje onou základní hyperonymií a hyponymií⁵ a pokračuje kupříkladu vztahy meronymie⁶ či antonymie⁷. Kromě vztahů mezi významy bude taková lexikální databáze obsahovat zřejmě i další informace, například o syntaktických kategoriích slov, definice jejich významů, etc. Databáze tak popsaných významů propojených sémantickými a případně i lexikálními vztahy⁸ může být nazývána ontologií. [18]

mam tady tu zminku vubec nechavat, nebo to mam jeste nejak rozvest.. je to duezite, ale nevim, jak moc pro tuhle praci

Od slovníků k WordNetu

Výše uvedená opozice papírového slovníku a ontologie ilustruje rozdíly tradičního slovníku a počítačově zpracovatelné lexikální databáze. Jedním z klíčových rozdílů je propojenost jednotek v lexikální databázi – tradiční slovníky, byvše v době svého vzniku většinou určeny pro distribuci v papírové formě určené pro lidského uživatele, neobsahují důsledné propojení sémanticky souvisejících slov. Příkladem budiž kostra a její části, např. lebka. V

⁴ angl. *machine readable dictionary*

⁵ podřazenosti

⁶ vztah je částí, tedy např. dveře je meronymem trolejbusu

⁷ protikladu

⁸ podrobněji v kapitole 1.5 na straně 15

SSČ⁹ i SSJČ se u lebký uvádí, že jde o kostru hlavy. Lze tedy s jistou rezervou tvrdit, že heslo obsahuje své holonymum¹⁰, opačný odkaz však již ani jeden z oněch dvou slovníků neobsahuje. Z celkem evidentních ekonomických důvodů nejsou u hesla kostra uvedeny všechny její části. Tento příklad příhodně ukazuje i jistou nesystémovost tradičních slovníků, která je pro počítačové zpracování fatální, jelikož, jak bylo zmíněno výše, znemožňuje systémové procházení hierarchie slovní zásoby a zjišťování podstaty jednotlivých významů.

Naznačeny tedy byly vlastnosti, jež by lexikální databáze měla oproti tradičnímu slovníku mít, aby byla použitelná pro počítačové zpracování přirozeného jazyka. Především jde o systémovost vztahů. Hypero-/hyponymie je vztah oboustranný, tudíž by mělo být možné se stejnou cestou dostat od nadřazeného slova k podřazenému a naopak. Dále je podstatné, aby sémantické vztahy mezi významy byly přesně definované, a tudíž algoritmicky zpracovatelné. Jedině tak je totiž možno jednoznačně určit, které slovo (či slova) je v takové databázi konkrétnímu slovu nadřazené, které je jeho specifikací, označením jeho částí, etc.

S touto myšlenkou vznikl WordNet – lexikální síť významů provázaných sémantickými vztahy a sovních forem provázaných lexikálními vztahy tak, aby to dle poznatků psycholingvistiky odráželo organizaci lexikálního materiálu v lidském mozku (více v kap. *K vlivu psycholingvistiky na organizaci WordNetu* na straně 9). [46]

K vlivu psycholingvistiky na organizaci WordNetu

Jelikož G. A. Miller, který byl koordinátorem projektu WordNet, byl svým zaměřením psycholog a přispěl k vzniku psycholingvistiky, ubíral se projekt Wordnetu podobným směrem. Společně s Johnson-Lairdem se Miller zaměřil na výzkum, jakým způsobem je lexikální materiál uložen v lidském mozku. Tento vědní směr je označován právě jako psycholingvistika a jeho počátky jsou spojeny s průzkumem asociací a způsobem modelování mentálního slovníku člověka. Výchozí myšlenka, jež se odráží i ve způsobu organizace WordNetu, spočívá v tom, že slovní zásoba je organizována konceptuálně a pro některé slovní druhy (zejména substantiva) hierarchicky. Konceptuálním uspořádáním je míněno seskupování slovních forem s podobným významem, hierarchická organizace pak staví významy do vztahů například zmíněné hyperonymie a hyponymie, tedy nadřazenosti a podřazenosti.

Jednou z otázek tohoto směru bylo, jakým způsobem je v hierarchickém modelu paměti řešeno získávání vlastností pro význam, které jsou „podděděné“ po významech hie-

⁹ Slovník spisovné češtiny

¹⁰ vztah opačný k meronymii; tedy např. dům je holonymem pro okno, dveře, práh etc.

rarchicky výše umístěných. Aby člověk byl schopen například určit pravdivostní hodnotu výroku Kanárek může létat, musí použít svou dlouhodobou paměť. Její organizace je pak možná (minimálně) dvěma způsoby. První, redundantní, by vypadal tak, že by u každé podtřídy ptáků bylo uloženo, že její instance jsou schopny létat. Druhý, již na první pohled výrazně méně náročný na úložný prostor, by příznak schopnosti létat měl uložený pouze u třídy pták. Pro zjištění, zda kanárek létá, by pak bylo nutno zapojit inferenční proces ve stylu *kanárek je pták, tudíž může létat*. [5]

OBRAZEK?

Jak Collins a Quillian [5] dále uvádí, lze předpokládat, že v případě prvního způsobu organizace paměti by člověk mohl kteroukoliv informaci o příznacích (vlastnostech) z paměti vyvolat za konstantní čas. Naproti tomu v případě způsobu druhého by extrakce příznaku z významu v hierarchii položeného dále měla trvat delší čas než extrakce například příznaku přítomného přímo u významu, kvůli němuž celý proces extrakce probíhá. Důvodem by měla být nutnost zapojení inferenčního procesu a jeho délka rostoucí se vzdálenostmi významů cílového a zdrojového, tedy toho, pro něž je příznak zjišťován, a toho, z něhož jsou znalosti o vlastnostech zdrojového významu čteny.

Pokus, kterým podpořili Collins a Quillian [5] druhý, neredundantní, způsob ukládání příznaků v paměti, spočíval v tom, že testovací subjekty, dobrovolníci z řad zaměstnanců společnosti Bolt Beranek and Newman, měly určovat, zda jim předložený výrok připadá pravdivý, či nepravdivý. Měli tak činit co nejpřesněji a v co nejkratším čase, přičemž byla měřena rychlost jejich reakce. Ukázalo se, reakční doba při určování pravdivosti výroku Kanárek umí létat¹¹ je delší než při určování pravdivosti výroku Kanárek umí zpívat¹² a ještě delší při určování výroku Kanárek má kůži¹³. Důvodem pro tyto progresivní prodlevy podle nich byla právě zvětšující se vzdálenost od významu kanárka k významu, u něhož byl uložen příslušný příznak, tedy umí zpívat, umí létat, resp. má kůži. Příznak umí zpívat totiž je pravděpodobně uložen přímo u kanárka, jelikož jej odlišuje od ostatních ptáků, zatímco příznak umí létat je obecným znakem ptáků, tudíž je uložen u významu pták. V poslední řadě pak příznak má kůži bude patrně uložen u významu zvíře, který je z oněch tří v hierarchii nejvýše, a tudíž od významu kanárek nejdále.

check,
jestli to
nebylo
umí mlu-
vit, takhle
to ne-
dava moc
smysl..

WordNet se svou hierarchickou organizací slovní zásoby tedy pravděpodobně konceptuálně blíží organizaci lexika v lidské paměti.

Navzdory tomu, že je WordNet (a všeobecně wordnety) využíván převážně k úkolům počítačového zpracování jazyka, zdá se být pro svou podobnost v uspořádání informací vhodný i v jiných oblastech. V souladu s příklady, jež byly předloženy výše, se zdá, že zkou-

¹¹angl. A canary can fly

¹²angl. A canary can sing

¹³angl. A canary has skin

mání významů a propojenosti konceptů v jazyce, s nímž není uživatel zcela obeznámen, by mohlo ve WordNetu být snazší než v klasickém slovníku. V podobném duchu využili při svých kursech WordNet například Lemnitzer a Kunze [34].

Organizace WordNetu

Ve WordNetu lze nalézt informace autosémantikách, slovech plnovýznamových, tedy substantivech, adjektivech, slovesech a příslovcích [60]. Synsémantika (např. předložky, spojky etc.) nebyla zahrnuta, jelikož se zdá, že jsou uložena odděleně od slov plnovýznamových. Teorii, že jsou funkční slova uchovávána jako součást syntaktikonu, podpořil kupříkladu Garrett [17] při svém pozorování afatických pacientů.

Vůbec první podnět k uvědomění, že různé slovní druhy podléhají různé strukturalizaci v paměti, vyvolal asociační test, který provedli Fillenbaum a Jones [12]. Tomuto asociačnímu testu byly podrobeny anglicky mluvící subjekty, které měly za úkol uvést první slovo, které je napadne při myšlence na předložené slovo. Předkládána jim byla dobře známá a často používaná slova náležející k různým slovním druhům. Ukázalo se, že ve většině případů náleží asociované slovo ke stejnému slovnímu druhu jako slovo, které asociaci vyvolalo. Substantiva vyvolala asociaci na substantivum v 79 % případů, adjektiva v 65 % případů a slovesa v 43 % případů.

Ačkoliv není zřejmé, jak je znalost o slovním druhu určitého slova získávána, lze z uvedených dat předpokládat, že slovní druh je vskutku primární organizační vlastností lexikálního materiálu v lidském mozku a informace o něm je snadno dostupná (alespoň intuitivně). Jelikož správné tvoření vět vyžaduje minimálně podvědomé ponětí o tom, které slovo náleží do které syntaktické kategorie, není s podivem, že tato informace je dostupná lidskému uvažování velmi jednoduše. Jelikož se však slova stejného slovního druhu příliš často nevyskytují pohromadě, není evidentní, jak tyto znalosti člověk získává. [12, 38]

zkont-
rolovat
procenta,
nesedi.
spis to
melo byt,
adj->adj,
apod.

Synsety a vztahy mezi nimi

Slova (slovní formy) jsou ve WordNetu seskupována podle svého významu (a dále také slovního druhu), k němuž náležejí. Řadě slov přináležejících ke stejnému významu se v terminologii WordNetu říká synset¹⁴, neboli synonymická řada. Každý synset reprezentuje jeden význam, ale je nutno mít na paměti, že granularita synsetů nemusí být konsistentní a v podstatě záleží na tom, jak si tvůrci zadefinovali synonymum (více v kap. *Synonymie* na straně 15). Synset je ve WordNetu reprezentací významu a je definován slovy (formami),

¹⁴angl. zkrácenina pro *synonym set*

kteřé obsahuje. Jelikož význam slov je definován tím, v jakém synsetu se vyskytují (ke kterému konceptu náleží), jde v podstatě o kruhovou definici, a tudíž je zřejmé, že definice významů musí být rozšířena. Lze říci, že význam konceptu reprezentovaného synsetem je založen na jeho pozici v celé struktuře. Význam konceptu je tedy definován jeho kontextem, to znamená nadřazenými a podřazenými koncepty. [32] Vztahy mezi koncepty jsou vztahy sémantické, jelikož se týkají významů slov (cf. lexikální vztahy níže v kapitole *Lexikální vztahy ve Wordnetu* na straně 15).

Aby bylo možno WordNet použít k inferenčnímu vyvozování závěrů (získávání informací) o slovech strojově, jsou synsety ve WordNetu propojeny vztahy, z nichž je zřejmé, jakou informaci inferenční stroj získá, přejde-li po onom vztahu k dalšímu konceptu. Strojové zpracování textu, potažmo strojová inference informací, se od té lidské liší v jednom zásadním aspektu, kterým je fakt, že stroj nemá k dispozici encyklopedické znalosti o světě, jež má k dispozici člověk. Pro strojové zpracování textu je tedy nutné zajistit, aby všechna potřebná data bylo možné získat z lexikální databáze, v tomto případě z WordNetu.

Zmíněné kritérium, že slovní formy jednoho synsetu musí náležet k jedné syntaktické kategorii (slovnímu druhu), je podloženo jednoduchým závěrem o nezaměnitelnosti slov příslušejících různým slovním druhům. Seskupování konceptů podle slovního druhu vede, zřejmě navzdory snaze o ekonomii ukládání informací, kterou se lidský mozek vyznačuje, k jisté redundantnosti systému. Existuje totiž mnoho slov (zvláště např. v angličtině), která zastupují jak substantivum, tak verbum (např. angl. show, popř. české stát). Míra sémantické podobnosti takových slov může být značně odlišná. V angličtině je relativně běžné, že substantivum popisuje činnost, k jejímuž dějovému vyjádření se užívá sloveso stejné formy (např. run vyjadřuje běh i běžet). U zmíněného českého stát sice lze vypořádat poněkud vzdálenou sémantickou příbuznost (pojmenování pro stát jako organizovanou územní a politickou mocenskou jednotku [44] je zřejmě motivováno jako něco stálého, co dlouho *stojí*), ale není to příliš intuitivní a takové dva výrazy nemohou být zařazeny do stejného konceptu. Slova náležející do odlišných syntaktických kategorií se rovněž syntakticky chovají zcela rozdílně a rozhodně v žádném kontextu nemohou být zaměněna jedno za druhé, což také znemožňuje jejich zařazení do stejného synsetu. [38]

Dalším argumentem pro striktní rozdělení slovní zásoby dle slovních druhů je fakt, že různé slovní druhy mají různou hierarchizaci. Jak bude popsáno v kapitole *Sémantické vztahy WordNetu* na straně 13, například substantiva jsou hierarchizována podle vztahu hyperonymie a hyponymie, přičemž u nich existují další vztahy jako meronymie, která například u sloves existovat nemůže. Naopak vztah antonymie, který je relativně běžný u adjektiv, se

u substantiv téměř nevysktuje¹⁵. Verba jsou zase provázána vztahy vyplývání, který u substantiv není příliš evidentní a intuitivní¹⁶, ale u sloves je vcelku hojný – například z činnosti zírat vyplývá i *nadřazená* činnost hledět.

Sémantické relace mezi slovy různých kategorií ve WordNetu neexistují, avšak pro tyto případy jsou definovány relace lexikální. Oproti sémantickým relacím, které provazují celé koncepty, jsou lexikální relace definovány na úrovni jednotlivých forem. Dvě stejné formy, například *run*, jedna náležející k substantivům, druhá k verbům, budou propojeny vztahem derivačně příbuzné formy¹⁷. [67]

Sémantické vztahy WordNetu

V této kapitole budou podrobněji rozebrány sémantické vztahy konstituující WordNet. Sémantické vztahy jsou, na rozdíl od vztahů lexikálních, které jsou vztahy mezi slovními formami, vztahy mezi koncepty (významy). Rozdíl nejlépe ilustruje protipříklad. Synonymie je typickým vztahem lexikálním; kdyby byla vztahem sémantickým, znamenalo by to, že dva různé významy (koncepty) mají stejný význam, což je nesmysl, jelikož v tom případě to nejsou dva významy, ale jeden.

Struktura těchto vztahů není, jak by se na první pohled mohlo zdát, plochá, ale organizovaná podle syntaktické kategorie významů, jež jsou jimi propojeny. Substantiva mají své vlastní vztahy, stejně tak adjektiva, verba a adverbia. Pojmenování těchto vztahů vychází z lingvistických termínů k nim relevantních (např. hyperonymie), ale v některých případech se pojmenování napříč různými syntaktickými kategoriemi překrývá, ačkoliv jde o vztahy různé. Například angl. sloveso *run*¹⁸ má ve WordNetu jako hyperonymum uveden synset s významem pohybovat se velmi rychle obsahující slovesa *travel rapidly*, *speed*, *hurry*, *zip*¹⁹. Je evidentní, že tento vztah hyperonymie není identický se vztahem hyperonymie u substantiv, kde *house*²⁰ má jako přímé hyperonymum uveden synset *building*, *edifice*²¹. Z činnosti běžet vyplývá činnost rychle se pohybovat, ale budova je pro dům nadřazenou třídou. Jde tedy o vztah nikoliv nepodobný, ale ne identický. [49]

¹⁵Lze argumentovat, že např. život je antonymem pro smrt, faktem ale je, že jde o velmi volnou antonymii – život popisuje stav či průběh doby, kdy je bytost živá, smrt referuje pouze k okamžiku, kdy se z živé bytosti stává mrtvá bytost, tedy rozhodně nejde o přímý protiklad jako například u adjektiv *světlý:tmavý* nebo *špatný:dobry*. Stejně tak např. Bůh a Dábel jsou sice proti sobě pokládány bytosti, ale jejich antonymie spočívá spíše ve vlastnostech jim připisovaných, tedy subjektivních, a uživatel jazyka může prohlásit, že obě tyto bytosti jsou špatné, čímž ztratí svou protikladnost.

¹⁶Asi lze tvrdit, že z života vyplývá smrt, ale pravděpodobně takto provázaných substantiv nebude mnoho.

¹⁷angl. *derivationally related form*

¹⁸čes. běžet

¹⁹čes. cestovat rychle, uhánět, ...

²⁰čes. dům

²¹čes. budova, stavba

Vzhledem k vlastnostem sémantických vztahů, jimiž jsou synsety WordNetu provázány, tvoří tyto orientovaný graf (což je v rozporu s obecně značně rozšířenou vizualizací WordNetu jako stromové struktury). Pokud se v tomto grafu vyskytnou anomálie (chyby), může se stát i cyklickým grafem (pokud by například synset měl za hyperonymum sám sebe). To je však zřejmě ojedinělý jev. [55] Zdánlivé cykly ve vizualizaci však nastat mohou, jelikož jeden koncept může mít více hyperonym (více v kapitole *Hyperonymie a hyponymie* na straně 14), která v určitém bodě budou mít opět společné hyperonymum, jelikož má WordNet jeden kořen.

Hyperonymie a hyponymie

Vztah nadřazenosti a podřazenosti strukturalizuje především slovní zásobu substantiv vztahem tříd a podtříd. Jde o vztahy transitivní a asymetrické. [38] Díky této hierarchizaci se lze například vyhnout redundanci ukládání informací v paměti, jelikož příznaky třídy není nutné ukládat u každé podtřídy. Podtřída dědí všechny příznaky své mateřské třídy a přidává minimálně jeden další. Například tramvaj je pouličním kolovým přepravníkem, který jezdí po kolejích a je poháněn elektřinou²². Pokud některý ze zděděných příznaků pro podtřidu neplatí, je tento fakt u ní explicitně uložen (podrobněji v kapitole *K vlivu psycholinguistiky na organizaci WordNetu* na straně 9). Uspořádání, v němž jsou atributy takto děděny, se nazývá dědičný systém²³ [57].

Substantiva jsou ve Wordnetu organizována tak, že každý význam má svůj mateřský význam (hyperonymum), kromě jednoho jediného, a tím je entity, tedy uměle vytvořený pojem sloužící jako kořen celé sítě. Jeden koncept může mít hyperonymních významů více, například house má jako svá hyperonyma uvedeny synsety (n) dwelling, home, domicile, abode, habitation, dwelling house a (n) building, edifice.

Meronymie a holonymie

Meronymie (a k ní komplementární vztah holonymie) jsou, navzdory nepříliš rozšířenému názvosloví, dalším vztahem, jenž je pro uživatele jazyka intuitivní a známý. Jde o vztah *být částí*, potažmo *mít část*. Meronymie je definována tak, že B je holonymem A v případě, že jednou z částí B je A. Meronymie je vztahem stejně jako hyperonymie transitivním a asymetrickým. [7] Tento vztah také hierarchizuje lexikum do určitých úrovní, ale na rozdíl od vztahu nadřazenosti, v němž obvykle jeden nadřazený význam mívá jeden až více podřazených významů, u vztahu části a celku je situace složitější. Je totiž na snadě, že jeden

²²ve Wordnetu *streetcar, tram, tramcar, trolley, trolley car (a wheeled vehicle that runs on rails and is propelled by electricity)* [49]

²³angl. *inheritance system*

význam může být meronymem mnoha holonymům – kupříkladu dveře mohou být meronymem například u dům, auto, šatník, počítačová skříň, etc.²⁴

Vztah části a celku je vlastní výhradně substantivům. U sloves se může vyskytovat vztah vyplývání a příčiny.

Lexikální vztahy ve Wordnetu

Synonymie

Synonymie je základním definičním vztahem pro synsety ve WordNetu. Na praktických aplikacích je tento jev nejlépe pozorovatelný, jelikož při vyhledání konkrétní formy je uživateli obvykle nabídnut výběr z jednotlivých významů dané formy. Aby byly od sebe významy oné formy odlišitelné, nabídka obvykle sestává ze seznamu skupin slovních forem náležejících do nalezených synsetů²⁵ Kupříkladu při vyhledání slova kolo v českém wordnetu tak je uživatel konfrontován s několika skupinami, které obsahují zhruba následující:

- kolo (1),
- jízdní kolo (1), bicykl (1), kolo (2),
- kružnice (1), cívka (1), kolo (3),
- etc.

citaci cs
wordnetu

přičemž čísla (zde) v závorce značí index významu dané formy v daném synsetu. [45] Reprezentace v různých aplikacích a různých wordnetech se liší (standardem bývá číslo významu psát za dvojtečku), koncept však zůstává neměnný.

Navzdory zdánlivé jednoduchosti uvedeného konceptu je všeobecnou otázkou, jak synonymii pojímat. Striktní teorie (obvykle připisovaná Leibnizovi) praví, že dvě slova jsou synonymní, pokud se jejich záměnou nikdy nezmění pravdivostní hodnota výroku. Lingvistickou interpretací tohoto poněkud matematicko-logického výroku může pak být, že synonymní dvě slova jsou v případě, že se jejich záměnou nikdy neporuší význam (zhruba ona pravdivostní hodnota) a gramatičnost výroku. Je nasnadě, že takto striktně synonymní slova budou pospolu v jazyce těžko přežívat, jelikož je dokázáno, že jazyk tíhne k ekonomičnosti, která by takovým soužitím dvou slov byla hrubě porušena [35]. Pravděpodobně jedinými obecně uznávanými synonymy jsou obvykle dvojice cizího slova a domácího slova, například internacionální a mezinárodní. Jejich záměnou se velice pravděpodobně nikdy pravdivostní hodnota výroku nezmění, stejně tak jako jeho gramatičnost.

²⁴Ve WordNetu to může být ilustrováno na jednom významu dveří, u něhož jsou jako holonymum uvedeny významy dveřní otvor, dveře, přístup do pokoje, práh (*doorway, door, room access, threshold*) [49]

²⁵<https://www.englishforums.com/English/AdjectiveSatellite/nwzhv/post.htm#1126701>

Stále však zůstává ve hře stylistika, která může být podobnou náhradou narušena (např. z důvodu cílové skupiny čtenářů či stylistické příznakovosti jednoho ze slov (cf. zajímavý a zajímavý)). Co se tendence k ekonomičnosti jazyka týče, lze předpokládat, že v těchto případech převládá potřeba synonym k eliminaci opakování určitých slov v textu a tím zajištění jeho stylistické uhlazenosti.

Volnější interpretace synonymie počítá ještě s kontextem. Dvě slova jsou synonymní, jsou-li bez způsobení škod nahraditelná alespoň ve stejném kontextu. Jako příklad mohou posloužit formy *board* a *plank*. V kontextu dřevařství mohou tyto dvě formy pravděpodobně bez problému být nahrazeny jedna za druhou, ovšem v případě, že je forma *board* použita ve významu *committee*, těžko ji lze nahradit formou *plank*, neboť by se věta obsahující takové nahrazení stala zcela nesmyslnou. [38] V českém kontextu mají zřejmě podobný vztah například formy *rada* a *výbor*. Slovní forma *rada* může sloužit jak ve významu rozhodovacího orgánu, což je zhruba synonymní s formou *výbor*, tak ve významu poučení (např. upřímná rada přítele). Opět je evidentní, že záměna forem *rada* a *výbor* v prvním významu slova *rada* (tedy rozhodovací orgán) je přijatelná, ve druhém už nikoliv (upřímný výbor přítele). Nutno podotknout, že forma *výbor* je také polysémní a její další významy se nekryjí s významy slova *rada*. [25]

Bylo by nanejvýš přirozené považovat synonymii za vztah diskrétní, tedy že dvě formy buďto synonymní jsou, či nejsou. Z logického hlediska to nepochybně z již uvedeného vyplývá, ovšem lingvisticko-filosofický náhled vycházející z poznatků reálného jazyka na tuto problematiku nahlíží poněkud odlišně. Synonymie v striktním slova smyslu je velice vzácná. Její volnější interpretace je značně častější, ale také výrazně vágnější – kontext, v němž dvě formy synonymní jsou, může být velmi široký, či naopak velice úzký. Záměna některých dvojic (či spíše obecně *n-tic*, volné synonymní řady mohou být vcelku dlouhé – *textil:1, látka:1, textile:2, plena:1, tkanina:1* [45]) může měnit stylistiku a význam výpovědi více či méně, přičemž ony dvě formy stále dle daných kritérií lze považovat za synonymní. Nelze tedy než vyvodit, že synonymie, minimálně z pohledu přirozeného jazyka, je jevem graduálním, a některé *n-tice* forem jsou tak svázány silnějším vztahem synonymie než jiné (laicky řečeno jsou *synonymnější* než jiné). [38]

Zaměnitelnost forem podporuje ještě jeden koncept, na němž je WordNet postaven, a to fakt, že jednotlivé významy jsou seskupovány podle slovních druhů. Tento systém vede k jisté redundantnosti, jelikož zvláště v syntetických jazycích, jako je kupříkladu angličtina, lze nalézt mnoho případů, kdy identická slovní forma zastupuje více slovních druhů. Významy, které taková slovní forma zastupuje (například slovními druhy), mohou být velice

manko,
nekecam?

blízké, nikdy však nebudou stejné (nelze říci, že význam slovesa run²⁶ a substantiva run²⁷ je identický). Jejich záměnou by se sice nestalo vůbec nic, jelikož čtenář či posluchač textu, v němž taková záměna nastala, by automaticky formu interpretoval ve prospěch správného slovního druhu, avšak pokud by slovní druh byl nějakým způsobem „vynucen“ (necht' nyní čtenář pomine úvahy, jakým způsobem lze vynutit slovní druh formy), stala by se výpověď zcela negramatickou a nesmyslnou.

Jakkoliv to není přímo svázané se synonymií, je na místě poznámka o výskytu stejné formy v různých synstetech. Slovo je kombinací slovní formy a významu, nebo slovního významu. Slovní forma je projevem „fyzickým“, tedy je to vyřčená či napsaná instance významu. Jak je zjevné z přirozeného jazyka, nelze počítat s tím, že by zobrazení významu na formu bylo bijektivní, tedy každý význam byl namapován jedna ku jedné na slovní formu. V přirozeném jazyce může jedna forma zastupovat více významů a jeden význam může být vyjádřen více formami. Příkladem budiž slovní forma koruna, která může zastupovat význam měny, vrcholku stromu, vladařského odznaku, etc. Toto zobrazení jedné formy na více významů se nazývá polysémií nebo homonymií²⁸. S polysémií souvisí ještě homonymie, což ve své podstatě dosti podobný vztah, ale totožnost formy je zcela nahodilá. Kupříkladu formu kolej lze interpretovat jako referenci k stopě po voze, případně dvojici kolejnic jako vodící dráze pro dopravní prostředky a zároveň jako zařízení vysoké školy pro ubytování studentů. [25] U významů formy koruna lze vyzkoušet nějaký společný základ (koruna stromu je nahoře, panovnickou korunu má panovník na hlavě, tedy nahoře, koruna jako mince zase pravděpodobně získala své pojmenování díky faktu, že na mincích bývá vyobrazen panovník). Naproti obě formy kolej pochází z odlišeného základu – kolej jako ubytovací zařízení pochází z latinského collegium, kdežto výraz pro dráhu je odvozeno od českého kolo. [53]

Seskupování významů podle slovních druhů a seskupování forem dle vztahu synonymie se tedy zdá v případě lexikální databáze určené pro strojové zpracování jako vhodným konceptem. Oproti tradičním slovníkům se totiž počítačově zpracovávaná lexikální databáze nemusí potýkat s problémem lidského faktoru – jednotlivé synonymické řady je stroj schopen prohledávat, na rozdíl od člověka, vcelku účinně, a nahradí tak v případě, že WordNet používá člověk, neúčinné lidské procházení restříkovaného obsahu.

²⁶běžet

²⁷běh

²⁸obojí znamená totožnost formy pro různé významy, u polysémie však ony významy mají společný základ (byť může být velmi vzdálený)

ne, není,
ale nech-
telo se mi
mazat 2k
napsaných
znaku xD

Antonymie

Antonymie, neboli protiklad, je navzdory zdánlivé triviálnosti koncept překvapivě těžce definovatelný. Všeobecně se antonymií rozumí významová opozice, faktem však je, že použití tohoto termínu je velmi široké a druhů antonymie je několik. Nejjednodušším druhem je například antonymie mezi adjektivy živý a mrtvý. Negace prvního automaticky značí druhé a naopak (je-li řeč o živých bytostech), jelikož v reálném světě neexistuje žádný další třetí stav. Tento jednoduchý vztah však nefunguje vždy – například s adjektivy bohatý a chudý je to jiné. Mnoho lidí se nepovažuje ani za chudé, ani za bohaté, a tudíž z toho, že někdo není bohatý, automaticky neplyne to, že by byl chudý. Miller et al. [38] Zajímavé je, že tento vztah není reflexivní. Pokud někdo není bohatý, tak to nemusí znamenat, že je chudý, ale pokud je o někom tvrzeno, že *je* bohatý, tak to nutně znamená, že *není* chudý. Paradis a Willners [47]

Rozdíl mezi výše uvedenými dvojicemi, tedy mrtvý:živý a chudý:bohatý spočívá ve stupňovatelnosti daných adjektiv. Pro ilustraci – lze říci, že někdo je bohatší než někdo jiný, ale nelze říci, že někdo je *mrtvější* než někdo jiný. Pokud jsou adjektiva stupňovatelná, tedy lze říci, že objekt A je více X než objekt B, neoznačují komplementární stav, ale graduální vlastnost. Označované pak může být zařazeno kamkoliv mezi tyto dva póly, přičemž nachází-li se v pomyslné střední šedé zóně, nelze jej označit výrazy odpovídajícími pólům gradientu. Tvrzení, že někdo není ani chudý, ani bohatý, dává smysl, protože tato adjektiva označují extrémní stavy, mezi nimiž je prostor pro normální stav. Paradis a Willners [47]

Vztah antonymie ve WordNetu je koncipován tak, aby zřejmě byl co nejpodobnější uvažování široké populace uživatelů jazyka, tedy užívá primitivního konceptu antonymie. Některé studie dokonce za antonymní považují výrazy pouze vágně, intuitivně protikladné, jako například muž:žena či chytrý:hloupý. [33]

Ve WordNetu se antonymie vyskytuje u substantiv (man:woman), adjektiv (rich:poor, a dokonce i white:black v rasovém významu²⁹), verb (open:close) i adverbií (well:ill).

²⁹cf. také antonymní vztah Caucasian:black ve WN

Kapitola 2

Další wordnety

Podle vzoru princetonského WordNetu začaly postupně vznikat i další sémantické sítě založené na stejném konceptu. Tyto sémantické sítě se samozřejmě svou strukturou do větší či menší míry liší, hlavním kritériem pro to, aby mohly být považovány za wordnet, je to, aby obsahovaly synsety a hyponyma. [22]

EuroWordNet

EuroWordNet je mezinárodní lexikální databáze pro osm evropských jazyků (angličtina, čeština, dánština, francouzština, italština, němčina, španělština). Jde o soubor jednotlivých národních wordnetů, které jsou propojeny takzvaným mezijazykovým indexem (ILI, *interlingual-index*). Obecně jsou wordnety EuroWordNetu založené svou strukturou na princetonském WordNetu (verze 1.5), ale z důvodu různorodosti jazyků se v některých aspektech od něj odlišují. EuroWordNet zavádí poněkud odlišené vztahy, navíc je diferencuje jemněji. [48]

Základní motivací pro vznik EuroWordNetu byla evropská jazyková různorodost a z ní pramenící problémy ve zpracování dat a napomáhání uživateli v přístupu k neanglickým datům. Vossen [61] argumentuje, že uživatel musí umět anglicky a být obeznámen s tím, jak je zdroj, v němž vyhledává napsán, aby byl schopen v něm účinně hledat. Vytvořením wordnetů pro jiné jazyky si slibuje, že se zlepší možnost přístupu uživatelů k neanglickým datům, možnosti inference znalostí z těchto dat a případně i mezijazykové vyhledávání. Poslední je založeno na faktu, že od počátku byly jednotlivé wordnety EuroWordNetu vytvářeny s tím, že budou propojeny na základě základních konceptů (BCS, *Base Concept Sets*) a mezijazykového indexu.

Jelikož se jednotlivé jazyky zapojené v projektu EuroWordNetu značně odlišují ve struktuře své slovní zásoby, jsou jednotlivé wordnety nezávislé. To znamená, že se mo-

Relace	Slovnědruhov \acute{e} kombinace	Př \acute{e} klad
antonymie	A-A, V-V	open:close
hyponymie	N-N, V-V	car:vehicle, walk:move
meronymie	N-N	head:nose
vyplývání ¹	V-V	buy:pay
následek	V-V	kill:die

Tabulka 2.1: Vztahy přejaté z princetonského WN (N: substantivum, A: adjektivum, V: verbum)

hou odlišovat například svou hierarchizací. Stejný koncept tak může ve dvou různých wordnetech mít různá hyperonyma, meronyma, etc., protože například anglické označení pro prst je odlišené, pokud jde o prst na noze (toe), či na ruce (finger). Podobně má v jiném příkladu dánština odlišené označení hlavy u zvířat, tedy kof, a hlavy lidské (hoofd). [61]

Národní wordnety jsou vzájemně propojené přes mezijazykový index s anglickým wordnetem, který je obsahově založený na princetonském WordNetu, ale není identický. Anglický wordnet byl přizpůsoben strukturně tak, aby byl použitelný v EuroWordNetu, tedy byly přidány dodatečné metainformace a druhy vztahů (podrobněji dále). V národních wordnetech existuje několik druhů konceptů, které jsou rozlišeny podle příbuznosti s koncepty v ostatních národních wordnetech. Pokud je koncept přítomen ve všech wordnetech EuroWordNetu, jde o koncept tzv. *Global Base Concept* (GBC). Koncept, jenž jen přítomen v alespoň dvou národních wordnetech je označován jako *Common Base Concept* (CBC) a v poslední řadě koncept, který se vyskytuje pouze v jednom národním wordnetu nese označení *Local Base Concept* (LBC). [21] Propojení konceptů společných pro více jazyků je zajištěno pomocí jednotných identifikátorů a mezijazykového indexu, který je nadmnožinou všech konceptů v EuroWordNetu. ILI je hierarchicky plochá struktura (proto *index*, nejde o další „všeojazykový“ wordnet). [61]

Jelikož v době, kdy EuroWordNetu vznikal, byl princetonský WordNet poněkud omezený mimo jiné co se vztahů mezi slovními druhy týče, vznikly pro EuroWordNet speciální vztahy umožňující úplnější práci s významy. Základní vztahy přejaté z princetonského WordNetu 1.5 jsou uvedeny v tabulce 2.1 na straně 20.

Navíc k těmto vztahům byly přidány štítky (*labels*), jež relaci konkretizují. Byly použity následující štítky:

- conjunction/disjunction
- non-factive
- reversed
- negation

Použití konjunktivního a disjunktivního štítku spočívá v myšlence, že například u meronymie by bylo vhodné rozlišovat, zda jde o části, které dohromady tvoří celek, nebo jde o podčásti částí (např. nůž má meronyma čepel, rukojeť, ostří, ale ostří je ve skutečnosti meronymem až rukojeti, nikoliv přímo samotného nože).

Štítek *non-factive* je používán u kauzální relace, která nemusí být nutně naplněna:

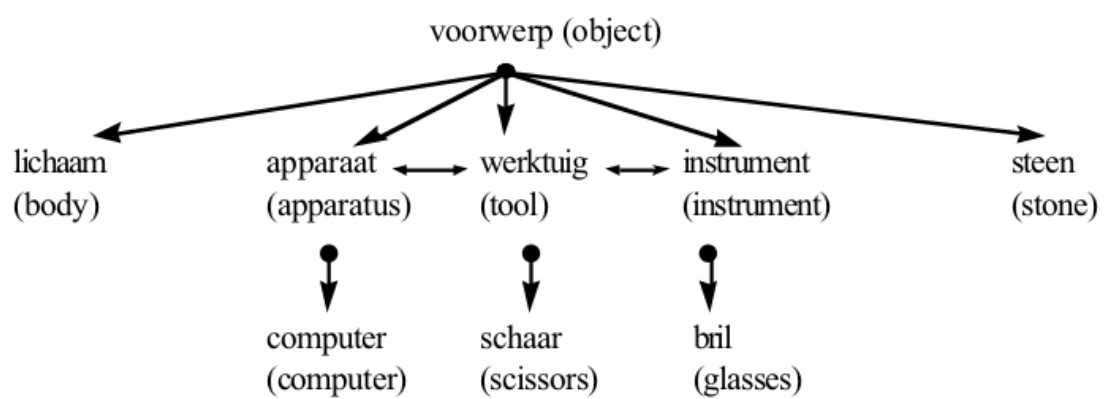
příčina	vztah	následek	non-factive?	nutně vyplývá?
zabít	vyúští v	zemřít	–	ano
hledat	vyúští v	najít	non-factive	ne

Podobně lze upřesnit pomocí štítků další relace tak, že jsou ve výsledku jednoznačnější a wordnet, v němž jsou takto označené vztahy obsaženy, může poskytovat více informací. Jejich použití v praxi je však značně omezené. [51]

Jako další rozšíření oproti tehdejší verzi princetonského WordNetu přinesl EuroWordNet také relace mezi slovními druhy a rozlišení mezi úzkým a širokým vztahem synonymie a antonymie (*near_synonym*, *near_antonym*). Argument pro zavedení mezislovnědruhových relací je relativně přímočarý, a to, že umožňují „sblížit“ koncepty, které jsou si příbuzné, jen náleží k jinému slovnímu druhu. Nutno podotknout, že v době vzniku této práce je princetonský WordNet ve verzi 3.1 a obsahuje už relaci *derivationally related form*, která zajišťuje přesně toto propojení (více o synsetech v princetonském WordNetu v kapitole *Synsety a vztahy mezi nimi* na straně 11).

Vztahy blízké synonymie a blízké antonymie byly zavedeny pro možnost vyjádření volnější sémantické příbuznosti. [48] Co se vztahu blízkého synonyma týče, důvodem pro jeho zavedení byl údajně zájem mít možnost přiblížit koncepty, které jsou si významově podobné, ale pouze na své úrovni. U takových konceptů platí, že byť jejich význam je podobný, jejich hyponyma nelze zařadit pod jeden koncept, jelikož se rozdíl mezi oněmi koncepty svázanými vztahem blízkého synonyma prohlubuje. Příkladem budiž trojice nizozemských slov *aparaat*, *werktuig* a *instrument*, jež jsou si významově nepříliš vzdálená:

Jak je z obrázku 2.1 na straně 22 zřejmé, všechna hyponyma slova *voorwerp* (objekt) jsou si rovna, avšak některá jsou si rovnější. Tři výše zmíněná slova jsou si navzájem významově výrazně bližší, než jsou si blízká s ostatními hyponymy na jejich úrovni. Právě aby bylo možno tento vztah reflektovat a tím docílit možnosti například nahrazovat za sebe slova, která sice nemohou být ve stejném synsetu, ale jsou si podobná, byl zaveden vztah blízkého synonyma. Lze totiž předpokládat, že uživatel jazyka podobná slova také může zaměnit. [61]



Obrázek 2.1: Blízká synonyma (k překreslení)

Část II

Přehled a porovnání existujících vizualizací sémantických sítí

Tato část textu bude zaměřena na zmapování existujících dostupných rozhraní a jejich zhodnocení. Výčet jednotlivých rozhraní rozhodně není vyčerpávající, jelikož některé vědecké instituce mohou vyvíjet své nástroje pro práci se sémantickými sítěmi neveřejně, některé nástroje už nejsou dostupné, etc. Nástroje pro vizualizaci² dat wordnetů byly vybrány pro přehled v této práci podle několika kritérií relevantních především pro koncového uživatele, avšak i pro případné vývojáře dalších wordnetů (tedy zda dané rozhraní mohou použít pro svá data).

²Terminologická poznámka: v této práci je vizualizací dat wordnetů míněna vizualizace jak textová reprezentace, tak grafická.

Kapitola 3

Metodologie porovnání

Výběr rozhraní

Hlavním kritériem pro zařazení do rozhraní do výběru byla jeho dohledatelnost, ať již podle seznamu příbuzných projektů na oficiálních webových stránkách princetonského WordNetu¹ (jenž je, dlužno podotknout, velice neaktuální v tom smyslu, že uvádí velké množství rozhraní, která již nejsou dostupná) či podle dotazu na vyhledávači Google²:

- wordnet visualization
- wordnet visualisation
- wordnet graph

Z přehledu byly samozřejmě vyřazeny implementace, které k době hledání³ už nebyly dostupné. Také nebyla zahrnuta rozhraní, která jsou funkčně podobná jiným. Ačkoliv většina existujících rozhraní zřejmě pracuje buďto výhradně s princetonským WordNetem, či jej jako zdroj dat nabízí jako jednu z možností, nebyl zdroj dat brán v potaz. Do přehledu byla zahrnována především rozhraní, která jsou dostupná z webového prohlížeče, jelikož je to pro koncového uživatele nejpohodlnější způsob přístupu k aplikaci (není nucen instalovat žádný přídatný software na svůj počítač). Jelikož však i rozhraní určená pro klasické počítače⁴ jsou hojně rozšířena a mohou také poskytnout cennou inspiraci i při vývoji webového rozhraní, byla některá z nich zahrnuta též. Jsou však popisována pouze povrchněji a měla by při interpretaci hodnocení být automaticky značně penalizována, jelikož je uživatel mající zájem takové rozhraní použít nucen instalovat aplikační rozhraní Java, pokud jej

¹<https://wordnet.princeton.edu/>

²<https://www.google.com>

³duben 2017

⁴terminologická poznámka: klasickým počítačem je myšleno všeobecně zařízení typu stolních počítačů, laptopů, případně tabletů etc., na nichž je provozován operační systém určený pro klasické počítače

nemá. Dalším důvodem k penalizaci je také fakt, tyto aplikace nejsou přímo spustitelné na mobilních zařízeních⁵ [24] a jejich portabilita do formy nativních aplikací může omezená a obtížná.

Podstatná z hlediska hodnocení je také univerzalita, tedy zda je zdrojový kód rozhraní otevřený a je možné jej použít i pro vizualizaci jiné sémantické sítě, než pro kterou bylo rozhraní vyvinuto. Je ovšem dlužno podotknout, že tato informace bývá často nedostupná; potom se v rámci této práce předpokládá, že kód otevřený není.

Zmiňované hodnocení záměrně není vyčísleno exaktně, jelikož každý uživatel má jiné nároky a tato práce si tedy klade za cíl pouze co nejobjektivněji zhodnotit vlastnosti jednotlivých rozhraní, nikoliv čtenáři vnuknout, že jedno rozhraní je lepší než jiné; to musí případně zhodnotit sám na základě předkládaných faktů a jeho vlastních požadavků na rozhraní.

Do přehledu byla tedy zahrnuta následující rozhraní:

- An interactive visualization of the Princeton WordNet database
<http://mateogianolio.com/wordnet-visualization/>
- Artha
<http://artha.sourceforge.net/wiki/index.php/Home>
- BabelNet
<http://babelnet.org>
- Cornetto Demo
<http://cornetto.clarin.inl.nl/wordnet.xql>
- GoldenDict
<http://goldendict.org/>
- sloWTool
<http://nl.ijs.si/slowtool/>
- Treebolic
<http://treebolic.sourceforge.net/>
- Visual Browser
<https://nlp.fi.muni.cz/projekty/visualbrowser/>
- wnbrowser⁶
<http://homepages.inf.ed.ac.uk/adubey/software/wnbrowser/index.html>
- WordNET Editor

⁵Pro Android existuje několik cest, jak aplikaci napsanou v Javě spustit (<http://www.wikihow.com/Get-Java-on-Android>), lze však předpokládat, že použitelnost vizualizací wordnetu bude na zařízení s malou obrazovkou poněkud omezená.

⁶sic erat scriptum

<http://wordventure.eti.pg.gda.pl/wne/wne.html>

- WordVis

<http://wordvis.com/>

Pro účely porovnání (určení východiska, baseline) bude do popisu zahrnuto ještě jedno rozhraní, a to oficiální vyhledávací rozhraní princetonského WordNetu (<http://wordnetweb.princeton.edu/perl/webwn>).

Strukturalizace přehledu a kritéria hodnocení

Výše vypsaná rozhraní budou v dalších kapitolách rozdělena podle toho, zda umožňují přístup z webového prohlížeče bez nutnosti instalace doplňků (například pro běh zásuvných modulů napsaných v Javě⁷), budou zhodnocena jejich positiva a negativa z hlediska použitelnosti pro získání informací o hledaném výrazu, komfort zacházení s rozhraním a případná omezení použitelnosti na různých zařízeních.

K rozdělení podle dostupnosti bylo přistoupeno jednak proto, že v době vzniku této práce je pokročilost webových technologií dostatečná k tomu, aby podobné vizualizace byly tvořeny jako webové stránky, a jednak proto, že cílem této práce je vytvořit všeobecně dostupné a použitelné rozhraní k wordnetům. Všeobecně použitelným je míněna použitelnost nejen na osobních počítačích, ale také na mobilních zařízeních, což v podstatě vyřazuje použití technologií jako jsou zásuvné moduly napsané v Javě či využívající Flash⁸ používané k realizaci různých existujících rozhraní (příkladem budiž Visual Editor využívající Javu).

Rozhraní jsou hodnocena na základě několika kritérií. Cílem je porovnat jejich přínos v kontextu ostatních existujících rozhraní a v kontextu současných trendů webových aplikací a ilustrovat, s jakými problémy se všeobecně rozhraní potýkají. Z tohoto důvodu právě nebyla do přehledu zařazována rozhraní, která se podobají svou funkcionalitou a ovládáním rozhraním již zařazeným. Kritéria hodnocení v této práci jsou následující:

- přínos oproti základnímu oficiálnímu rozhraní princetonského WordNetu
 - originální vizualizace dat vedoucí k možnosti identifikovat trendy v datech, které zůstávají uživateli skryty případně základní textové reprezentace dat
 - reprezentace dat vhodnější z hlediska zásad přístupnosti webu⁹

⁷<https://docs.oracle.com/javase/tutorial/deployment/applet/>

⁸<http://www.adobe.com/products/flashruntimes.html>

⁹web accessibility

- responsivita rozhraní – kvalitní použitelnost rozhraní i na zařízeních s menší obrazovkou, jako jsou mobilní zařízení typu chytrý telefon či tablet
- v případě webových aplikací ukládání stavu aplikace v adresním řádku prohlížeče (umožňuje sdílení nebo založení odkazu ke konkrétnímu hledání a zachování nastavení aplikace)

Podmínky testování

Pro testování rozhraní, které bylo pro účely této práce provedeno, byl použit operační systém Ubuntu 16.04 LTS v 64bitové verzi, webový prohlížeč Pale Moon ve verzi 27.1.1 (rovněž 64bitový) s nastaveným uživatelským agentem na Firefox (z důvodů kompatibility). Ačkoliv se počítalo, vzhledem k nepříliš rozšířenému vykresovacímu jádru toho prohlížeče, s možnými problémy v zobrazení některých rozhraní, nebyly tyto zjištěny, a tak nebylo nutné provádět testování rozhraní ještě v dalších prohlížečích. Lze předpokládat, že v rozšířených prohlížečích, jako jsou Firefox či Chrome, budou jednotlivá rozhraní fungovat a vypadá podobně, jako v prohlížeči Pale Moon.

Testování na mobilním chytrém telefonu bylo prováděno na zařízení Nexus 5 (displej s úhlopříčkou 5 palců a rozlišením 1080 × 1920 px) s operačním systémem Android 6 v prohlížeči Chrome (sestavení 57.0). Možnost *request desktop site* byla nastavena na *vypnuto*.

WordNet Search jako základ porovnání

Oficiální rozhraní k princetonskému WordNetu je koncipováno jako textová reprezentace dat, přičemž, byť volitelně, umožňuje zobrazovat veškeré k hledanému výrazu relevantní informace, jež WordNet obsahuje. Data jsou vizualizována jako soubor seznamů.

Po vyhledání zadaného výrazu jsou uživateli prezentovány v bodech jednotlivé synsety obsahující hledanou slovní formu rozdělené podle slovních druhů, k nimž přináležejí. V základním nastavení každý bod sestává ze slovních forem tvořících daný synset, jeho definici, případný příklad užití a odkaz na detaily synsetu. Kliknutím na tento odkaz se zobrazí v podseznamu sémantické a lexikální relace, v nichž je daný synset přítomen. Samotné relace jsou opět odkazy, jejichž otevřením je uživateli prezentován seznam synsetů, které jsou spolu s daným synsetem v dané relaci přítomny. Postupným otevíráním detailů synsetů v podseznamech se lze zanořovat hlouběji a hlouběji do struktury, resp. se ve struktuře vzdalovat původnímu synsetu po hranách relací.

Uživatel má možnosti si zvolit množství informací, jež jsou mu prezentovány, a to od úplného minima (pouze slovní formy nalezených synsetů a jejich syntaktické kategorie) až

po vše, co se ve WordNetu vyskytuje, včetně identifikátorů a dalších technických detailů. Rozhraní svůj stav ukládá jako parametry URL¹⁰, takže je možné jej později obnovit z adresy.

Rozhraní není responsivní, ale vzhledem k tomu, že je v základu relativně úzké (pod 700 px), je na chytrém telefonu použitelné.

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- **S: (n) wheel** (a simple machine consisting of a circular frame with spokes (or a solid disc) that can rotate on a shaft or axle (as in vehicles or other machines))
 - [direct hyponym](#) / [full hyponym](#)
 - [part meronym](#)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - **S: (n) machine, simple machine** (a device for overcoming resistance at one point by applying force at some other point)
 - [direct hyponym](#) / [full hyponym](#)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - **S: (n) mechanical device** (mechanism consisting of a device that works on mechanical principles)
 - [part holonym](#)
 - [derivationally related form](#)
- **S: (n) steering wheel, wheel** (a handwheel that is used for steering)
- **S: (n) wheel** (forces that provide energy and direction) "*the wheels of government began to turn*"
- **S: (n) wheel** (a circular helm to control the rudder of a vessel)
- **S: (n) roulette wheel, wheel** (game equipment consisting of a wheel with slots that is used for gambling; the wheel rotates horizontally and players bet on which slot the

Obrázek 3.1: Ukázka rozhraní: WordNet Search 3.1 v základním nastavení a s otevřenými detaily synsetů

¹⁰Uniform Resource Locator

Podobnou jako WordNet Search splňují i některá další rozhraní. Za zmínku stojí například WNSearch¹¹, což je minimalistické čiště textové rozhraní, které díky absenci jakýchkoliv stylů je responsivní a dobře použitelné na mobilních zařízeních s libovolně velkým displejem. Mezi další rozhraní s podobnou funkcionalitou patří URCS Wordnet Browser (pro Javu)¹² či GoldenDict (rozhraní pro klasické počítače)¹³.

¹¹<http://www.golovchenko.org/cgi-bin/wnsearch>

¹²<http://www.cs.rochester.edu/research/cisd/wordnet/>

¹³<http://goldendict.org/>

Kapitola 4

Vizualizace s webovým rozhraním

Jako bylo naznačeno v úvodu této části, vizualizací s webovým rozhraním se v této práci míní taková implementace, která ze strany uživatele nevyžaduje instalaci žádných doplňujících aplikací či aplikačních prostředí (např. Javy) kromě samotného webového prohlížeče.

An interactive visualization of the Princeton WordNet database

Jeden z projektů programátora jménem Mateo Gianolio z Lundské univerzity (Švédsko)¹. Jde o jednoduché rozhraní napojené na princetonský WordNet, které po zadání hledaného výrazu zobrazí synsety obsahující daný výraz. Jednotlivé synsety jsou barevně odlišeny podle syntaktických kategorií (slovních druhů), k nimž náležejí, a uspořádány kruhově kolem hledaného výrazu. Z každého synsetu je vždy zobrazena jen první slovní forma. Pokud uživatel najede kurzorem myši na některý ze synsetů, zobrazí se další případné slovní formy náležející do daného synsetu a jeho glosa (definice), je-li dostupná.

Další slovní formy v nalezených synsetech jsou klikatelné, což umožňuje dostat se přesně na synsety obsahující onu slovní formu, na níž bylo uživatelem kliknuto (s identickým výsledkem, jako kdyby dané slovo uživatel zadal do vyhledávacího pole).

Ačkoliv úvodní odstavec na stránce s rozhraním vybízí uživatele k „prozkoumání de-sambiguace slov“ [19], zobrazují výsledky hledání pouze synsety a slovní formy do nich náležející, přičemž synsety připojené sémantickými vztahy hyponymie, meronymie etc. nejsou zobrazitelné (jinak než případným náhodným výskytem jedné slovní formy ve více synsetech). To použití WordNetu omezuje na obyčejný thesaurus.

Samotný design grafického rozhraní je řešen poněkud nešťastně a působí dojmem, že cílem autora bylo prokázat své schopnosti používat různé standardní transformační funkce. Pro indikaci běžícího procesu vyhledávání byl použit efekt rotační animace pro

¹<https://www.linkedin.com/in/mateo-gianolio-a89558b6/?trk=profile-badge-name>

vyhledávací pole a po dokončení vyhledávání jsou výsledky zobrazeny paprskovitě okolo vyhledávacího pole. To způsobuje, že některé texty jsou zobrazeny pod úhlem až 90 stupňů, což může pro některé uživatele činit jejich přečtení obtížnějším.

Rozhraní není tzv. responsivní, tedy nepřizpůsobuje se velikosti obrazovky, na níž je zobrazeno.

Toto rozhraní tedy poskytuje velice omezený přístup k datům WordNetu a neposkytuje žádné výhody oproti základnímu oficiálnímu rozhraní k princetonskému WordNetu. Jeho grafické pojetí je čistě arbitrární, neslouží žádnému účelu a naopak zhoršuje jeho použitelnost.



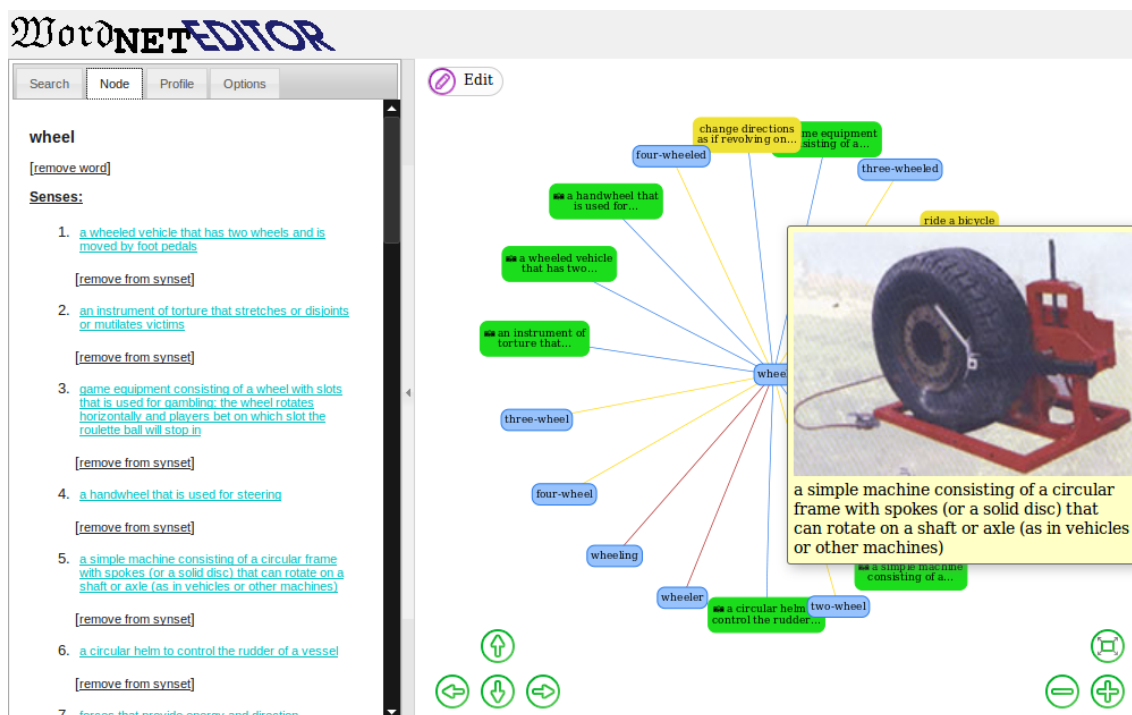
Obrázek 4.1: Ukázka rozhraní An interactive visualization of the Princeton WordNet database

WordNET Editor

Webová aplikace WordNET Editor byla vytvořena s cílem poskytnout internetové komunitě praktický a účinný nástroj pro rozvoj WordNetu. Její autoři argumentují, že jelikož vývoj WordNetu vyžaduje velké množství lidské práce, a je proto důležité jej tvořit tak, jako je tvořena internetová encyklopedie Wikipedie, tedy kolaborací nezávislých uživatelů. [56] K tomu je však potřeba nástroj, který takovou spolupráci umožní, což podle nich oficiální nástroje princetonského WordNetu nejsou (mimo jiné pravděpodobně proto, že WordNet byl vyvíjen uzavřenou skupinou lidí [11]). Vedle editoru vztahů a synsetů však obsahuje WordNET Editor i prohlížeč wordnetu, pro který bylo rozhraní do tohoto přehledu zařazeno. Data, jež aplikace používá, jsou verzí princetonského WordNetu odvětvenou v jeho verzi 2.1. [56]

Rozhraní je rozděleno do levého sloupce a pravého (většího) „plátna“, jež slouží pro grafické vyobrazení vztahů. Po vyhledání zadaného výrazu nabídne aplikace uživateli seznam synsetů, v nichž se výraz nachází. Nabídka synsetů je klikatelná a po uživateli je po zvolení synsetu kliknutím prezentováno grafické vyobrazení daného synsetu se všemi jeho vztahy. V levém sloupci si pak uživatel může vybírat, další synsety, které chce do pravé sekce přidat, může také otevírat dvojklikem již zobrazené synsety a lze i v nastavení filtrovat, které vztahy a slovní druhy se mají zobrazovat. Aplikace však zřejmě neobsahuje intuitivní způsob jak zobrazené synsety z plátna vpravo odebírat zobrazené synsetu a při otevírání dalších plátno nepřepisuje, ale synsety přidává k existujícím (neprovázaně). To může vést ke značné nepřehlednosti reprezentace, avšak aplikace umožňuje přibližování a oddalování plátna a vizuální přesouvání zobrazení po plátně, což s jistým úsilím uživatele nepřehlednost může eliminovat. Také nutno vytknout absenci textového pojmenování vztahů mezi synsety. Barevné rozlišení je v tomto případě nevhodně zvolené, jelikož vztahů je v rozhraní hodně a některé barvy jsou obtížně odlišitelné.

Grafické zobrazení vztahů mezi koncepty je řešeno hvězdčovitě, přičemž ve středu je zvolený synset, jenž byl výsledkem hledání, a od něj jsou vedeny hrany představující vztahy k příbuzným synsetům. Zajímavý je originální koncept zobrazení synsetů příbuzných k hledanému (otevřenému) synsetu. Ten je v grafu reprezentován svými členskými slovními formami, ale příbuzné koncepty jsou reprezentovány svými definicemi. Trochu na závalu však je, že se zřejmě nedá (minimálně v prohlížečím režimu) zobrazit detail příbuzného synsetu a uživatel se dozví pouze jeho definici, nikoliv slovní formy do něj náležející. Jediný detail, který je uživateli dostupný, je u některých synsetů obrázek (o jeho zdroji se však v rozhraní nepíše).



Obrázek 4.2: Ukázka rozhraní WordNET Editor: zvolený synset ve středu grafu a zobrazený obrázek k příbuznému synsetu

Test chování tohoto rozhraní na mobilním zařízení nebyl proveden, jelikož je zjevné, že je určeno především k editacím a podle toho je také navrženo. Rozhraní poskytuje grafický náhled na strukturu dat, ale pro nevhodné kódování vztahů do barev a nezobrazování dostatečného množství informací je jeho ovládání neintuitivní. Nutno podotknout, že rozhraní tak, jak je vyobrazeno v Szymanski et al. [56], vypadá značně odlišně a například obsahuje pojmenované vztahy.

Podle webové stránky projektu² je rozhraní vyvinuto s otevřeným zdrojem, pro jeho obdržení je však nutno kontaktovat vývojáře. Lze ale předpokládat jeho použitelnost i pro ostatní wordnety.

Cornetto Demo

Cornetto Demo je prohlížeč pro nizozemský wordnet a jako jedno z mála dostupných webových aplikací kombinuje grafickou a textovou vizualizaci dat. To je dáno zřejmě mimo jiné tím, že Cornetto Demo není pouze rozhraním pro wordnet, ale kombinuje data ze dvou

²<http://wordventure.eti.pg.gda.pl>

zdrojů. Jednak z Referentie Bestand Nederlands [36], a jednak z nizozemského wordnetu. Referentie Bestand Nederlands je slovník obsahující informace podobně strukturované jako FrameNet [13] spolu s rozšířením o kombinatorické chování slov v určitém významu [28].

Rozhraní je rozděleno na tři moduly, základní vyhledávání, pokročilé vyhledávání a vizualizace synsetů.

Základní vyhledávání slouží k prostému vyhledávání lexikálních jednotek. Dotaz je možné základním způsobem omezovat či rozšiřovat, a to pomocí zástupných znaků (*, ?³) a volby slovních druhů, v jakých se má vyhledávat. [6]

Pokročilé vyhledávání umožňuje v nizozemském wordnetu najít lexikální jednotky, které mají společné specifické parametry. Jako kritéria pro vyhledávání lze zvolit kterékoliv kombinace všech příznaků, jež jsou ve wordnetu u lexikálních jednotek přítomny. Takto je možné například vyhledat všechna slovesa, která jsou řazena do domény tance a jsou označena jako archaická.

Pokud hledaný výraz či zvolená kritéria odpovídají některým lexikálním vyskytujícím se v databázi, je uživateli prezentován seznam nalezených jednotek, přes něž se lze prokliknutím dostat na detail té které jednotky. Detail je textovou reprezentací dostupných dat, tedy syntaktických a sémantických informací o vyhledaném slově (nutno podotknout, že značná část informací v této části zřejmě pochází spíše Referentie Bestand Nederlands, nikoliv z wordnetu) a hierarchického zařazení konceptu, do něžž slovo patří, ve wordnetu. Hierarchické zařazení je vyvedeno ve formě podobné tradičnímu znázornění adresářového stromu a obsahuje, zřejmě z úsporných důvodů, pouze přímé nadřazené a přímé podřazené koncepty a v rozhraní chybí úplné zobrazení cesty od kořene wordnetu k zobrazenému synsetu. I tak může strom být relativně dlouhý vzhledem k tomu, že u obecných hyperonym bývá seznam hyponym rozsáhlý.

Z odkazů v hierarchii lze kliknutím na ikonu wordnetu u synsetu přejít do vizualizačního režimu, čímž se zobrazí grafické znázornění daného synsetu.

Modul vizualizace synsetů je v praxi pouze pseudomodul, jelikož jeho vyhledávání se od základního vyhledávání liší pouze tím, že vybírá jeden výsledek vyhledávání a rovnou jej zobrazí v grafické znázornění. Který výsledek se má vybrat, je možné zvolit při inicializaci hledání. Je tak eliminována nutnost zvolit synset, otevřít jeho detaily a kliknout na výše zmíněnou ikonu, aby se uživatel dostal na grafické znázornění daného synset, jedná se však pouze o zkratku v rozhraní, jež nezavádá nic nového.

Samotná vizualizace je realizovaná pomocí javascriptové knihovny d3 [4] a umožňuje uživateli prohlížet všechny synsety svázané nějakým vztahem s vyhledaným synsetem.

³přičemž * zastupuje posloupnost kterýchkoliv znaků, ? zastupuje jeden kterýkoliv znak

vizualizuje
vizuali-
zační vi-
zualizace,
mwaha-
haha

Simple Search

Advanced Search

Synset Visualization

Help

Simple Search

Matching Lexical Entries

Selected LE

Related LE or Synset

Details of Related Lexical Entry (rad-n-2)




General		Word Forms			Sense	
Lemma:	rad	Form	Art.	Number	Definition:	wiel v.e. voertuig
Part of Speech:	noun	rad	het	singular		
Morpho-syntax:	n	raderen	de	plural		
		raden	de	plural		

Semantics	Syntax	Relations
Countability:	(No syntactics)	(No sense relations)
Reference:		(No form relations)
Semantic Type:		

Examples

Type/Phrase	Canonical Form
freeCombination / np	de raderen van een treinstel/fiets

Hierarchy

-  [schijf-n-1](#)
-  [rad-n-2](#) [wiel-n-1](#)
-  [loopwiel-n-1](#)

Obrázek 4.3: Ukázka rozhraní Cornetto Demo: textová reprezentace dat

Zobrazení rozlišuje syntaktické kategorie i sémantické vztahy barevně, přičemž vztahy jsou navíc ještě popsány slovně. Autoři použili rozšířený model vizualizace synsetů tak, že synset je představován uzlem, k němuž vede sémantický vztah, a z tohoto uzlu pak vedou nepojmenované hrany představující vazbu slovní formy na daný synset (tedy v zásadě vztah synonymie). Tento způsob je alternativní k zobrazení, v němž sémantické vztahy vedou k jednotlivým slovním formám a není tak na první pohled zřejmé, které navázané formy jsou součástí kterých synsetů. Zde použité zobrazení vhodně omezuje množství hran, které jsou nutné k vykreslení grafu, byť možná na úkor srozumitelnosti pro neznalého uživatele, jelikož body označující synset a hrany označující synonymii nejsou nijak označeny v grafu (pouze v legendě a při přejetí myši). Všechny uzly mají přiřazenou i vlastní legendu ve formě informační bubliny, která obsahuje například definici, příklady, slovní druh či identifikátory. Vizualizaci je možné kolečkem myši přibližovat a oddalovat a tažením myši se po zvětšeném grafu přesouvat. Funkčnost přibližování je možná poněkud diskutabilní, jelikož se při jejím použití nemění proporce zobrazovaných informací (cf. elektronické mapy na WWW, které při přiblížení sice zvětšují detail struktury, ale zachová-

Obrázek 4.4: Ukázka rozhraní Cornetto Demo: ilustrace přeplnění grafické reprezentace hyponymy u slova plant

Rozhraní je neresponsivní, takže je na mobilním zařízení poněkud nepohodlné jej používat, byť to není nemožné. Šířka bloku s informacemi je natolik velká, že je nutné horizontálního rolování při čtení textu, což se považuje z hlediska použitelnosti webu za velmi negativní [43, 54]. Reprezentace grafem je vzhledem k ovládání, které používá, ještě hůře použitelná. Dotyk prstem do plochy s vykresleným grafem je totiž zároveň interpretován jako rolování celé stránky a zároveň jako tažení plátna s grafem do stran, které je běžně prováděno tažením myši po plátně. Přibližování a oddalování grafu, nikoliv celé stránky, v testovacím prostředí nebylo možné vůbec, ale vzhledem k vlastnostem této funkce popsa-

⁴niz. rostlina

ným výše to použitelnost nijak neomezuje. Co v mobilním testovacím prostředí v podstatě nefungovalo, byly informační bubliny u slov, jež se normálně zobrazují nad slovy; zde se zobrazovaly zcela mimo inkriminované slovo a částečně mimo zobrazovanou plochu.

Oproti základnímu rozhraní WordNet Search 3.1 přináší toto rozhraní přehlednější textovou reprezentaci dat spojenou s vizualizací vztahů vyhledaného synsetu. Textové rozhraní však zobrazuje poněkud omezené množství informací z wordnetu a soustřeďuje se především na data ze slovníku Referentie Bestand Nederlands. Vztahy mezi slovy jsou zobrazovány pouze v grafické reprezentaci dat, což vzhledem k její nepříliš vysoké použitelnosti na mobilních zařízeních s menší obrazovkou může být omezující. Rozhraní uchovává svůj stav podrobně v URL a umožňuje jej z ní plně obnovit.

WordVis

WordVis je rozhraní zaměřené na vizualizaci dat princetonského WordNetu grafem. Se-stává z vyhledávacího pole, levého sloupce s výběrem synsetů a plátnem, na němž je vykreslen graf vztahů zvoleného synsetu nebo slovní formy.

Ačkoliv to na první pohled uživateli nemusí být zřejmé, vizualizace funguje ve dvou režimech. Prvním je zobrazení slovní formy a synsetů, v nichž se tato slovní forma vyskytuje, druhé je zobrazení synsetu jako centrální jednotky a vztahů a slovních forem, které k danému synsetu patří. První zobrazení je užíváno například při prvotním zobrazení výsledků hledání. Vizualizace zobrazí ve středu grafu vyhledanou slovní formu a připojí k ní nepojmenovanými hranami synsety, jež danou slovní formu obsahují. Pokud pak uživatel zvolí kliknutím myši některý konkrétní synset, zobrazí se ve středu vizualizace jeho značka, k níž jsou připojeny pojmenovanými hranami představujícími sémantické vztahy další synsety. Stejně jako v prvním zobrazení, i v tomto se slovní formy náležející k určitému synsetu připojují nepojmenovanými hranami jako textové uzly. Pokud jedna slovní forma náleží k více synsetům (např. substantivum *bicycle* a sloveso *bicycle*), je v grafu její textový uzel pouze jednou a vedou od něj dvě nepojmenované hrany k náležitým synsetům.

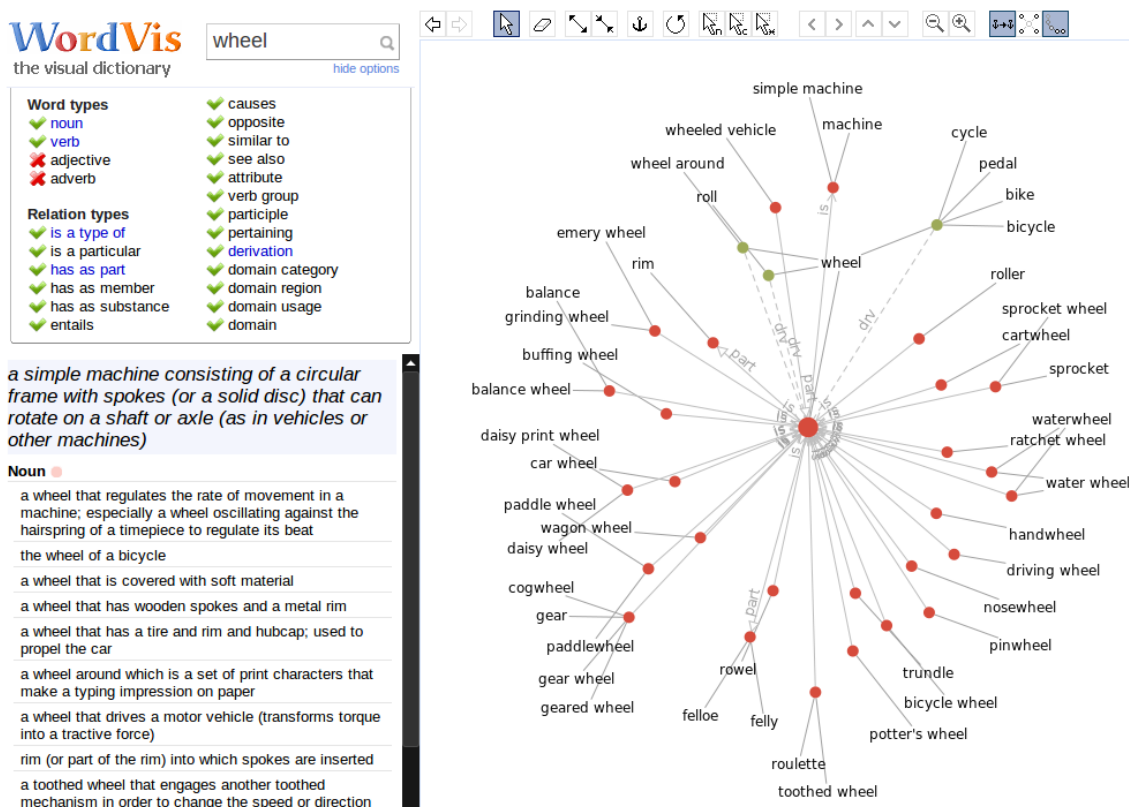
Hrany mezi synsety jsou orientované (znázorněné jako šipky) a jejich k jejich pojmenování zvolil autor sémantické významy daných vztahů, tedy např. hyperonymie je značena slovem *is*⁵; např. *apple tree is fruit tree*⁶.

Vyhledávací formulář umožňuje filtrování výsledků hledání podle slovních druhů a typů vztahu. Graf je navíc aktualizován v reálném čase podle toho, jak jsou podmínky

⁵angl. *je*

⁶angl. jabloň je ovocný strom

filtrování uživatelem měněny. Nevýhodou však je absence funkcí *vybrat vše* a *nevybrat nic*, která je citelná zvláště v případě, že uživatel chce vybrat pouze jeden druh relace (kterých je mnoho, tudíž odebrání všech relací z výběru kromě jedné může být časově náročné). Hledání také nabízí funkci napovídání poté, co uživatel zadá několik znaků z počátku slova. To umožňuje vybrat hledané slovo z nabídky a ušetřit několik úhozů, což je zvláště užitečné u delších slov či pro uživatele, který nevyhledává ve svém rodném jazyce a není si jist pravopisem hledaného slova.



Obrázek 4.5: Ukázka rozhraní WordVis (po zvolení konkrétního vyhledaného synsetu)

Technologicky je vizualizace řešena javascriptovou knihovnou, jež vykresluje graf na HTML5 plátně [63]. Uspořádání prvků na plátně je řešeno modelováním fyzikálních vlastností našeho světa. Jednotlivé uzly na plátně mají zdefinováno, že se navzájem odpuzují (podobně jako elektrony), naopak vazby mezi nimi fungují jako pružiny, jež mají nějakou ideální délku, v níž se snaží setrvávat. Díky těmto dvěma soupeřícím silám jsou body na plátně rozmístěny tak, aby využívaly prostor co nejúčinněji. Hrany navíc mohou mít zadanou preferovanou orientaci, což umožňuje orientovat například hyperonymní synsety směrem nahoru, hyponymní směrem dolů etc. [59] Vzhledem k tomu, že však zřejmě v pro-

středí neexistuje žádná penalizace za křížení hran, je relativně běžné, že se hrany překříží. To značně snižuje přehlednost výsledného grafu, vzhledem k tomu, že pak není jasné, zda jsou hrany překříženy z nějakého hlubšího důvodu (například jedna slovní forma náleží k více synsetům), či nikoliv. Knihovna vykreslující graf umožňuje uživateli také modifikovat tažením jednotlivých uzlů myši, prodlužovat či zkracovat hrany a podobně. V aplikaci pro zobrazení dat z WordNetu toto nehraje příliš zásadní roli a implementace této funkcionality je zřejmě dána tím, že knihovna je určena pro univerzálnější použití. [59]

Rozhraní je neresponsivní a na mobilním zařízení je nutné k zajištění čitelnosti textů na stránce obsah přiblížit, což vede k nutnosti rolování jak vertikálnímu, tak horizontálnímu. Také není možné využít funkcionality přesouvání prvků na plátně, což ale při použití na vizualizaci dat z wordnetu není velkým omezením funkčnosti. Co se udržování stavu aplikace v URL týče, je implementováno pouze čtení parametru *q* (od *query*), zpětně do něj ale už zapisováno není. Odkazy na synsety na levé straně tento parametr obsahují, stejně tak jej lze zkopírovat přes kontextové menu pro každý uzel v rozhraní, takže teoreticky je možné obnovit kterýkoliv stav aplikace, byť dostupnost této funkcionality není příliš intuitivní.

Přínos tohoto rozhraní tkví především v relativně přehledné vizualizaci dat princetonského WordNetu, nutno však podotknout, že pro některé své vlastnosti nemusí pro nezkušeného uživatele být zpočátku příliš intuitivní (např. zmíněné křížení hran).

Na knihovně, na níž je toto rozhraní postaveno, stojí ještě některá další rozhraní k princetonскому WordNetu, například VisuWords⁷ či Visual Thesaurus⁸. Ta nebyla do hodnocení v této práci zahrnuta, protože jsou funkčně podobná tomuto rozhraní.

sloWTool

Rozhraní sloWTool bylo vyvinuto pro potřeby slovinského wordnetu, jenž je založen na princetonském WordNetu a vznikl skombinováním několika zdrojů, například Wikipedie, dvojjazyčných slovníků či paralelních korpusů. Je víceúčelovým nástrojem, který umožňuje textovou reprezentaci, vizualizaci a editaci dat z wordnetu. Tvůrcům sloužilo k revizím slovinského wordnetu po jeho rozšiřování automatickými nástroji. [15] Cílem při vývoji tohoto rozhraní podle Fišer a Novak [14] bylo překonat nevýhody tehdejších dostupných rozhraní a vyvinout nástroj, který by mezi jiným umožňoval prohlížení i editaci, spolupráci mnoha autorů včetně anonymních editací (záměr tvůrců byl využít náhodných návštěvníků k opravování chyb, jež ve wordnetu naleznou) či možnost jednoduché registrace uživatelů. Mezi dalšími podmínkami byla možnost přidávání dalších wordnetů do systému

⁷<http://www.visuwords.com/>, <https://www.linux.com/news/visuwords-wordnet-goes-graphical>

⁸<https://www.visualthesaurus.com>

a s tím spojená schopnost rozhraní zobrazovat vícejazyčná data. V neposlední řadě se autoři v kontextu tehdejších nástrojů také snažili zaměřit na platformní nezávislost a přenositelnost rozhraní.

Rozložení stránky je vzdáleně podobné tomu u základního oficiálního rozhraní k princetonскому WordNetu, a to v tom smyslu, že neobsahuje relativně rozšířený levý sloupec na výběr synsetů, ale jednotlivé významy, které jsou nalezeny po zadání hledané slovní formy do vyhledávacího pole, seskupuje do sekcí v hlavním bloku s textovou reprezentací dat. Vyhledávací pole podporuje napovídání existujících významů, což může usnadnit práci s rozhraním. V textové reprezentaci rozhraní zobrazuje zřejmě všechny dostupné informace o synsetech, tedy se chová podobně jako základní rozhraní princetonského WordNetu, je-li nastaveno tak, aby nefiltrovalo žádné informace. U jednotlivých vztahů, v nichž je daný synset přítomen, pak je možné kliknutím otevřít synset nacházející se pomyslně na druhé straně daného vztahu (např. *bicykle* má *meronymum* *sedlo*, takže lze otevřít detail synsetu *sedlo*). To trpí stejným neduhem, jako základní rozhraní princetonského WordNetu, to jest, že lze ve smyčce otevírat nadřazený a podřazený synset a stále se zanořovat do smyčky hlouběji. To je nejen nesmyslné, ale zároveň to v implementaci tohoto rozhraní postupně může začít zpomalovat rychlost reakcí prohlížeče pro nadměrné množství uzlů v DOM⁹.

Po levé straně se nachází lišta s ikonami odkazujícími na dalšími moduly, které se otevírají v emulacích nových oken (v témže panelu). Mezi tyto moduly patří mimo jiné i pokročilé hledání, vizualizace a nápověda.

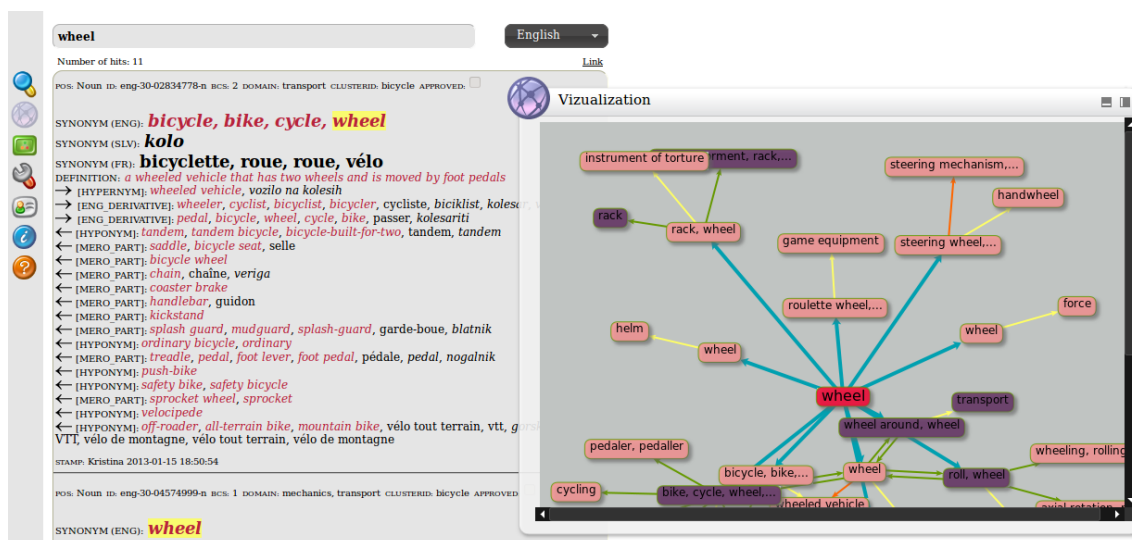
Pokročilé hledání funguje podobně jako u ostatních rozhraní, tedy umožňuje používat zástupné znaky (* a ?), filtrovat slovní druhy, nebo vyhledávat v jiných polích než jsou slovní formy patřící do synsetů (například v definicích či podle identifikátoru synsetu).

Vizualizace, modul z hlediska této práce nejpodstatnější, je určen k zobrazení grafické interpretace vztahů vyhledaného synsetu. Na rozdíl od ostatních rozhraní tato vizualizace zobrazuje všechny vyhledané synsety (s kořenem grafu označeným jako *search*¹⁰) a neumožňuje jejich filtrování. Také nepodporuje zřejmě žádnou formu interakce kromě přesouvání prvků na plátně a zvýraznění příslušného synsetu v hlavním bloku textové reprezentace poté, co je na něj (nebo jeho členskou slovní formu) ve vizualizaci kliknuto. Pokud uživatel potřebuje zobrazit vizualizaci jednoho konkrétního synsetu, je nucen použít pokročilé vyhledávání a vyhledat daný synset nejlépe podle jeho identifikátoru, který je vždy unikátní. Vizualizace je zatížena také dalšími problémy, jako jsou nedostatečné možnosti přizpůsobování velikosti zobrazovací plochy (zvětšování okna je podporováno, ale plátno

⁹document object model

¹⁰hledání

s grafem zůstává konstantně velké), občasnou desynchronizací obsahu vizualizace s výsledky hledání, náročností na uživatelské technické vybavení počítače (především výpočetní jednotku) a značnou nepřehledností způsobenou nedostatečně či nevhodně řešenou penalizací překrývání prvků na zobrazovací ploše.



Obrázek 4.6: Ukázka rozhraní slowTool (s otevřenou grafickou vizualizací)

Rozhraní je svými možnostmi zprostředkování informací z wordnetu poměrně inovativní, reálná použitelnost ovšem trpí zmíněnými nedostatky a odchyluje jej tím od původního záměru autorů, který minimálně zčásti zůstává nenaplněn. Je sice pravdou, že *s aplikací je možné pracovat v každém moderním přehlížeči, ať už na počítači, tabletu, či dokonce mobilním telefonu* [14], nutno ale podotknout, že rozhraní je zcela neresponsivní a jeho ovládací prvky jsou zcela nevhodné pro ovládání na menší obrazovce a dotykem.

Stav aplikace lze do jisté míry uložit pomocí odkazu *Link*, který vede na adresu, z níž lze obnovit hledané slovo (konfiguraci otevřených nástrojů však už nikoliv).

Navzdory všem nedostatkům této aplikace je dlužno uznat, že se svou univerzalitou a funkcionalitou značně přibližuje rozhraní, které je praktickým cílem i této práce.

SlowTool bylo vytvořeno pod licencí Creative Commons¹¹ a jeho zdrojový kód je dostupný na platformě pro aplikace s otevřeným zdrojem Launchpad¹².

¹¹<https://creativecommons.org/licenses/by-nc-sa/3.0/>

¹²<https://launchpad.net/slowtool>

BabelNetXplorer

BabelNet je rozsáhlý projekt vícejazykové sémantické sítě, který čerpá data z více zdrojů. Záměrem autorů bylo při tvorbě této sémantické sítě eliminovat základní faktory definující nevýhody tehdejších (a potažmo i aktuálních) mezinárodních projektů zabývajících se sémantickými sítěmi. Těmi jsou manuální tvorba dat, které síť konstituují, a s tím související nerovnoměrnost množství dat přes jednotlivé jazyky. Jazyky s vysokou hustotou zdrojů, jakým je například angličtina, tak ve výsledku mají více dat i v sémantické síti. Autoři BabelNetu se pokusili tento problém překonat kombinací několika metod, jejichž společným jmenovatelem je automatizace. Informace o významech jsou v BabelNetu doplňovány z Wikipedie, která je podle autorů díky mnohačetným zásahům expertů z různých oborů ve výsledku přesným a informarčně bohatým zdrojem. Druhá důležitá metoda, zaměřující se především na nerovnoměrnost dat v různých jazycích, je automatický překlad zdrojů. [41]

BabelNetXplorer je webové grafické rozhraní vytvořené pro vizualizaci dat z BabelNetu. [42] uvádějí, že rozhraní slouží k vizualizaci vztahů pro slova nalezená v BabelNetu a ilustrují vzhled rozhraní dvěma snímky obrazovky. V době vzniku této práce však rozhraní BabelNetXploreru vypadá výrazně odlišně, což je vzhledem k tomu, že od vzniku práce [42] uběhlo pět let, pochopitelné. V této práci bude z evidentních důvodů rozebrána použitelnost a funkcionalita současného rozhraní.

Rozhraní obsahuje klasicky vyhledávací pole, vedle něhož se nacházejí rozbalovací menu, v nichž si uživatel může vybrat, ve kterém jazyce chce vyhledávat a, případně, do kterého jazyce chce slovo přeložit. Po úspěšném dokončení vyhledávání jsou uživateli zobrazeny odkazy na jednotlivé synsety v seznamu, který není nepodobný ostatním rozhraním. Podstatným rozdílem však je, že se v něm zobrazují i ilustrační obrázky, které, byť nejsou vždy velmi informativní, mohou napomoci uživateli v orientaci, který synset jej zajímá.

V detailu synsetu, který se zobrazí po kliknutí na příslušný odkaz, je uživateli zobrazen seznam slovních forem, které k danému synsetu náleží, jeho definice (s možností zobrazit i jeho definice nejen z daného wordnetu, v němž uživatel vyhledává, ale i z Wikipedie a dalších zdrojů), jeho případný překlad a sémantické vztahu, do nichž tento synset náleží. K dispozici má uživatel i možnost zobrazit si daný synset paralelně v dalších jazycích pomocí nabídky pod vyhledávacím polem (viditelné na snímku obrazovky 4.7 na straně 44). Níže na stránce jsou uživateli prezentovány informace z dalších zdrojů napojených na BabelNet, jako jsou obrázky, překlady, odkazy na Wikipedii, etc. a odkaz na vizuální reprezentaci sémantických vztahů otevřeného synsetu.

The screenshot displays the BabelXplorer web interface. At the top, there is a search bar with the word 'wheel' entered. Below the search bar, the language is set to 'English'. The interface shows the synset for 'wheel' in English, including its categories (Cycle types, Sustainable transport, etc.) and its relationships (IS A, PART OF, HAS PART, etc.). The Czech translation 'Jízdní kolo' is also shown, along with its own semantic network. The interface is designed to be user-friendly, with clear labels and a structured layout.

Obrázek 4.7: Ukázka rozhraní BabelXplorer (detail synsetu v textové reprezentaci)

Vizuální reprezentace je řešena tradičním hvězdicovitým grafem, který má dva režimy. Jeden zobrazuje vedle textových názvů jednotlivých synsetů přítomných v grafu i jejich zastupné obrázky, druhý je čistě textový a, nutno podotknout, značně přehlednější. Uzly reprezentující synsety jsou provázány barevně odlišenými hranami, kde barvy značí druh vztahů, který dva synsety spojuje. Uzly jsou klikatelné, přičemž po kliknutí na některý z příbuzných synsetů se zobrazí tento synset a opět hvězdicovitě další synsety, s nimiž je zkoumaný synset provázán některým ze vztahů. Při najetí kurzorem myši na určitý synset se zobrazí jeho detail; tyto informační bubliny jsou však vázány na pozici kurzoru myši a nelze tak kurzorem najet do detailu tak, aby bylo možné kliknout na odkazy, které jsou v informační bublině obsaženy, a dostat se tak na textovou reprezentaci daného synsetu. Zdá se tedy, že není možné přejít z grafické reprezentace do textové.

Poněkud nešťastně je řešen design vizualizace, jelikož jednak není možné graf přibližovat a oddalovat, a jednak pokud uživatel nenajede na konkrétní synset kursorom myši, graf se samovolně neustále otáčí, což ztěžuje orientaci v něm.

Rozhraní všeobecně responsivní, textová reprezentace je dobře použitelná i na mobilním rozhraní s malou obrazovkou. Grafická reprezentace je však omezena na větší obrazovky, jelikož v ní není možné graf posouvat po obrazovce, a tudíž jeho značná část (byť se částečně přizpůsobuje zobrazovací ploše), zřejmě v závislosti na velikosti obrazovky zařízení, zůstává uživateli skryta.

V době vzniku této práce mělo rozhraní dvě verze, z nichž jedna byla označena jako *živá beta*, druhá jako *současná* (3.7), avšak při testování nebyly zjištěny zásadní rozdíly mezi těmito verzemi. Nutno ovšem podotknout, že na testovacím desktopovém prohlížeči vykazovala verze beta minoritní chyby v rozložení stránky.

Kapitola 5

Vizualizace v aplikačním prostředí Java

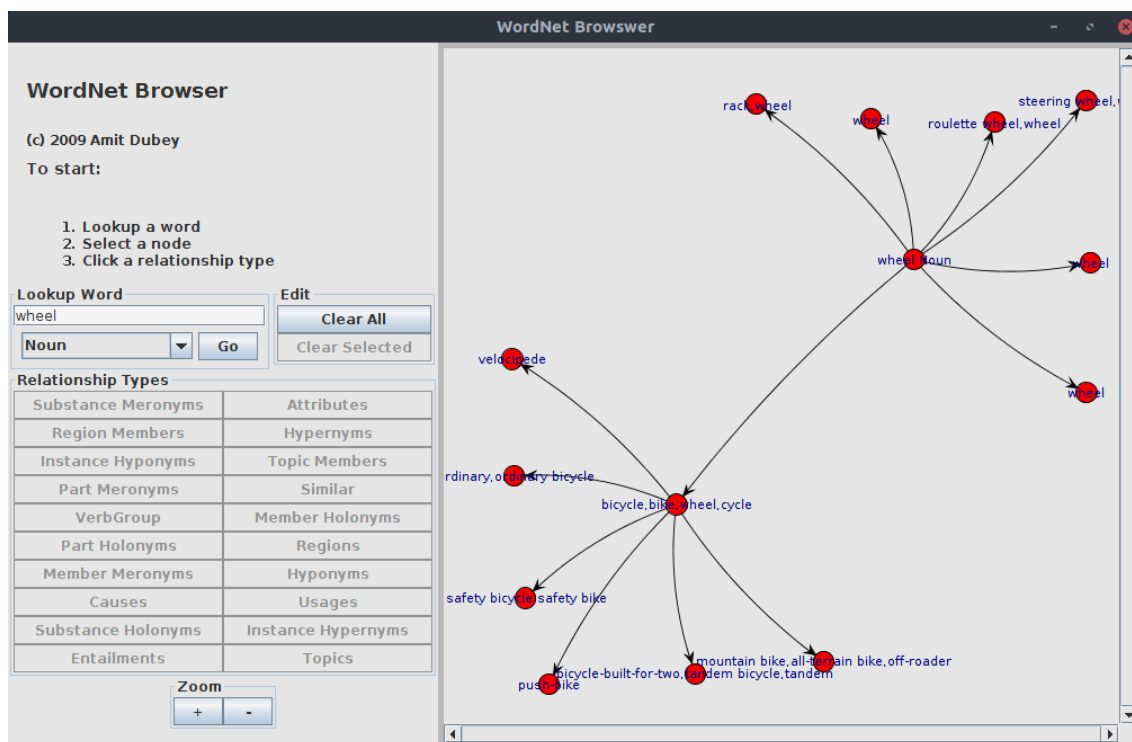
Java je objektově orientovaný programovací jazyk, který je zaměřený na to, aby měl co nejméně závislostí na konkrétním operačním systému a technickém vybavení počítače, jak jen to je možné. To je výhodné zejména kvůli tzv. konceptu WORA (*write once, run anywhere*, tedy napiš jednou, spust' kdekoliv), protože to znamená, že skompilované programy napsané v Javě lze spustit na jakémkoliv stroji, který podporuje Javu. Takto skompilované programy se spouští v aplikačním prostředí zvaném Java Virtual Machine¹ (JVM), jež vytváří uniformní prostředí pro běh aplikací. Výhodou Javy tedy je velká přenositelnost programů v tomto jazyce napsaných, jelikož jednou skompilovaný kód lze spustit na různých operačních systémech a zařízeních. Nevýhodou je, že si uživatel, který chce program v Javě spustit, musí nainstalovat aplikační prostředí Javy (JVM).

Java je velmi oblíbená pro vývoj aplikací, které fungují podle modelu klient–server, což je uspořádání, v němž data má uložená centrální server, k němuž se připojují uživatelské klienty, odesílají mu své požadavky a uživatelům zobrazují vrácená data. Na podobném principu může fungovat i rozhraní pro wordnet, pokud jsou data wordnetu uložena na serveru.

Vizualizace vytvořené pro prostředí Java nemá příliš smysl hodnotit z hlediska responsiveness, jelikož a priori nejsou vytvářena pro mobilní zařízení (až na výjimky, které nebylo možné zhodnotit z důvodu jejich nedostupnosti v době vzniku této práce). Co však bráno v potaz být může je jejich schopnost přizpůsobit se velikosti obrazovky. Jakkoliv tento parametr může se zdát být samozřejmým, ukázalo se, že ne všechna rozhraní jsou na různé velikosti obrazovek osobních počítačů přizpůsobena.

¹virtuální stroj Javy

wnbrowser



Obrázek 5.1: Ukázka rozhraní wnbrowser

Rozhraní wnbrowser² je určeno k prezentaci dat z princetonskému WordNetu, které pracuje pouze s grafovou grafickou reprezentací. Bylo vytvořeno pro reprezentaci dat z WordNetu ve verzi 3.0 a vyšší a vyžaduje lokální instalaci dat. Rozhraní je zřejmě svázáno se vznikem rumunského wordnetu [14]. Sestává z vyhledávacího pole, volby, mezi kterými syntaktickými kategoriemi se má vyhledávat, a selektoru na typy vazeb, jež mají být zobrazeny. Po úspěšném vyhledání konkrétního slova se zobrazí graf konceptuálně podobný grafu z rozhraní sloWTool, tedy ve středu je uzel reprezentující hledaný výraz, z něhož vedou hrany k jednotlivým synsetům, jež danou slovní formu obsahují.

Grafická reprezentace je interaktivní, tedy na ní lze zvolit, jaký druh sémantického vztahu se má zobrazit, a lze tak u každého uzlu postupně zobrazit všechny synsety, které jsou s ním provázány; toho lze docílit buď přes kontextové menu, nebo přes volby vztahů v levém sloupci rozhraní, v němž se mimo jiné nachází i vyhledávací pole. Rozhraní zobra-

²sic erat scriptum

zuje i detaily zvoleného synsetu (po kliknutí na daný synset), takže až na technické informace typu číslo významu a podobně je schopno zobrazovat všechna dostupná data.

Jakkoliv velikost okna s aplikací měnit lze, velikost plochy pro vykreslování grafu zůstává neměnné velikosti, lze tedy v terminologii webových aplikací říci, že je neresponsivní, jelikož se nedokáže přizpůsobit velikosti obrazovky zařízení uživatele.

Aplikaci je možné používat bez připojení k Internetu, jelikož spoléhá na data WordNetu nainstalovaná lokálně.

Poněkud komicky působí fakt, že rozhraní má tři různé názvy: *wnbrowser* (hlavní nadpis stránky projektu), *WordNet Browser* (titulek okna aplikace) a *wnbrowser* (ostatní výskyty).

Treebolic

Treebolic je aplikace určená k zobrazování hierarchických dat v hyperbolicke reprezentaci, která zajišťuje, že důležitá data v centru obrazovky jsou zobrazena uživateli ve větším měřítku, než data na okrajích grafu. Reprezentaci si lze představit jako plochu zobrazenou přes čočku.

Zobrazení používá model prezentující vše, co je nalezeno k hledanému slovu, tedy při úspěšném vyhledání výsledků se ve prostřed vizualizace zobrazí reprezentace vyhledávání, s níž jsou hranami provázány všechny nalezené synsety. Oproti ostatním zde popisovaným prostředím navíc seskupuje tyto synsety podle slovního druhu, tedy zavádí další metakategorii v zobrazení, což přispívá k přehlednosti zobrazení.

Co jí naopak ubírá, jsou poněkud kryptické ikony sémantických vztahů, jejichž význam je sice popsán na hraně, kde jsou dané ikony umístěny, popřípadě je možno jej odhalit pomocí nabídky *Info* v kontextovém menu po kliknutí pravým tlačítkem danou ikonu, ale při prvním setkání se toto řešení nezdá být příliš ergonomickým (mimo jiné proto, že duplikuje informace). Ne zcela evidentní jsou také číselné a procentuální údaje v místě uzlů se synsety. Uživatel se sice díky řešení, které používá konexe hranami ze synsetů k členským slovním formám, dozví, o který synset jde (díky oněm členským slovním formám), ale význam zmíněných údajů zůstává skryt. Výhodou je barevné kódování významů jednotlivých hran. Barvy jsou navíc editovatelné v nastavení, takže teoreticky rozhraní vyhoví i osobám se zhoršeným vnímáním barev.

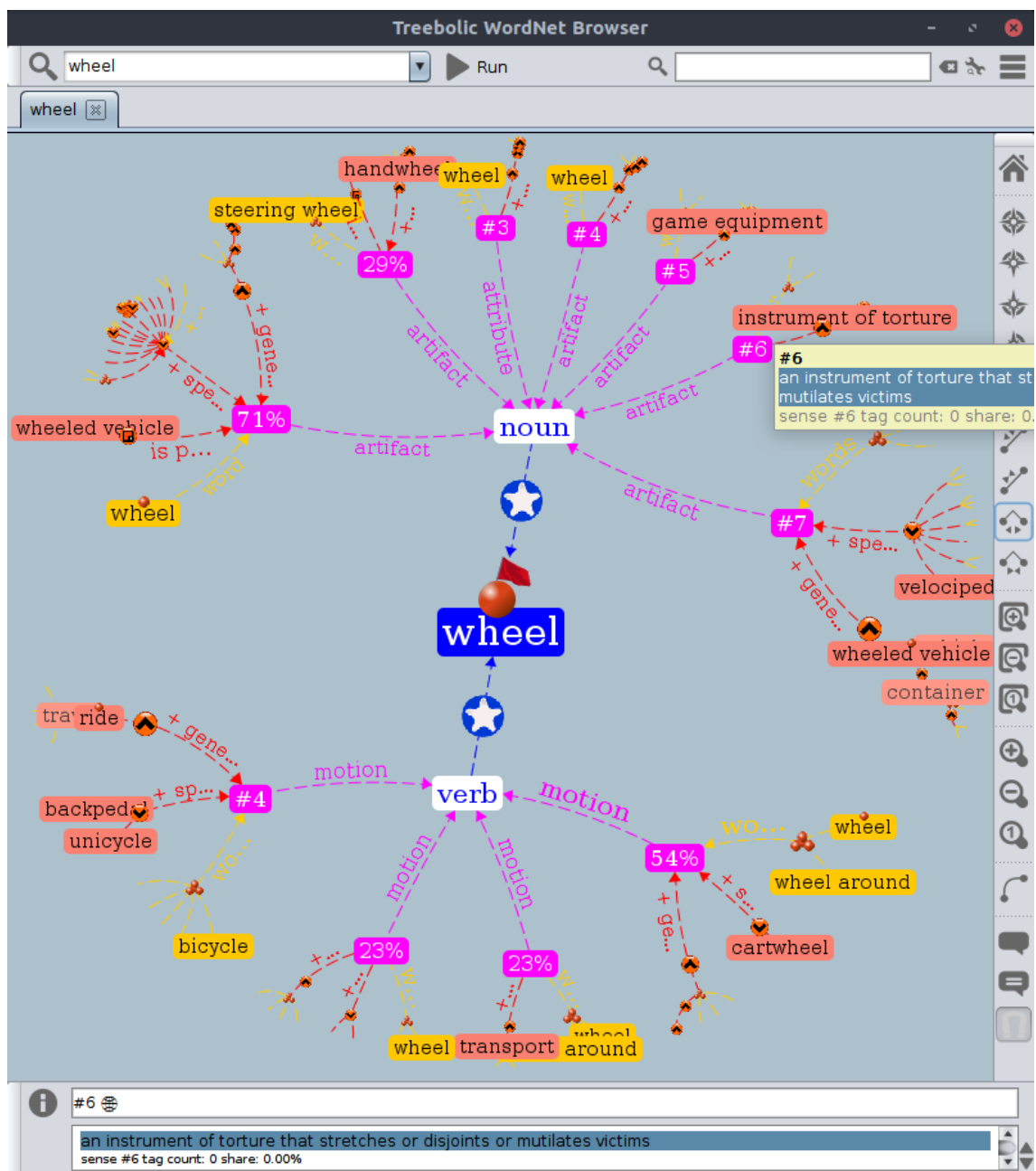
Aplikace byla vyvinuta i pro Android (jakkoliv v rámci této práce netestována), tudíž lze říci, že jistým zprostředkovaným způsobem splňuje požadavky na responsivitu.

Aplikaci je možné používat bez připojení k Internetu, jelikož spoléhá na data WordNetu nainstalovaná lokálně.

Teoretický přínos této implementace spočívá v pohledu na stratifikaci důležitosti vizualizovaných dat. Idea hyperbolického zobrazení, rozvinutá dále především v oblasti ovladatelnosti, by mohl být správným krokem prezentace dat ve větším množství, než je schopna obrazovka (a potažmo zorné pole) uživatele pojmout.

VisualBrowser

Verze aplikace VisualBrowser testovaná v rámci této práce neprokázala dostatečnou intuitivitu a funkčnost, aby její vlastnosti mohly být dále rozebrány.



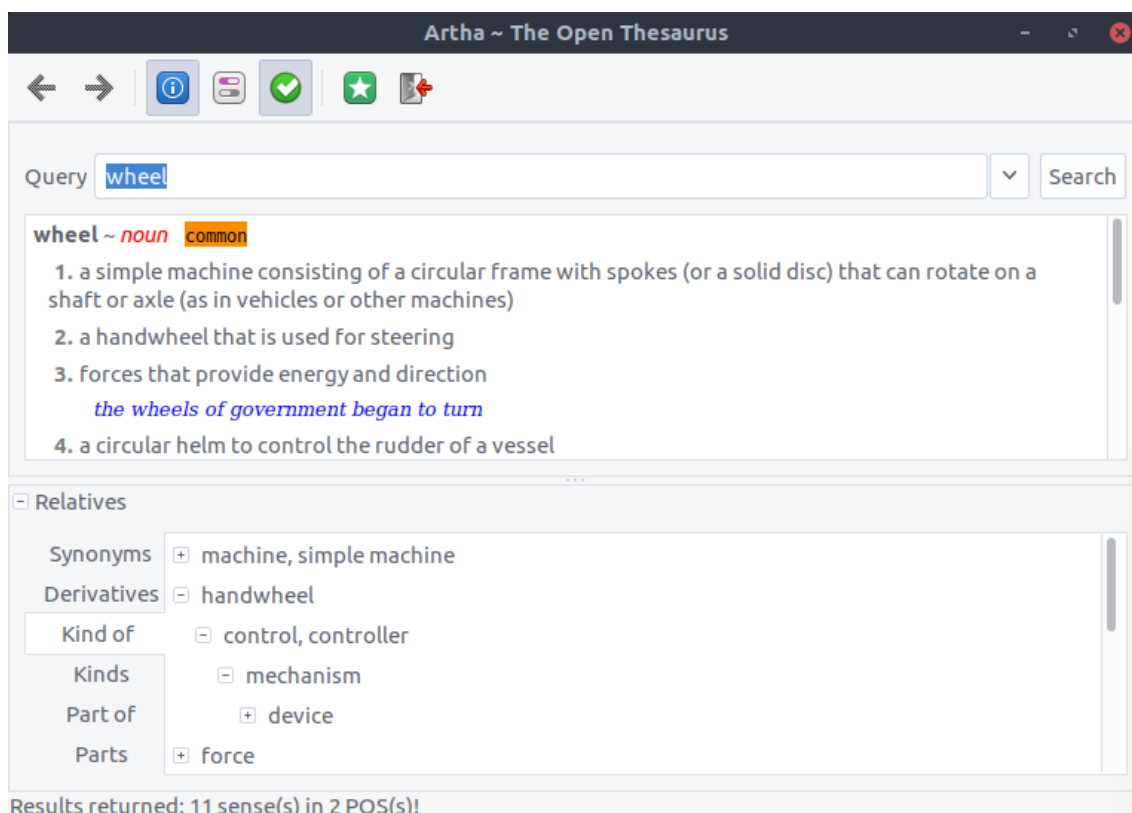
Obrázek 5.2: Ukázka rozhraní Treebolic (se zobrazenou informační bublinou na synsetu)

Kapitola 6

Desktopové aplikace

Desktopové aplikace jsou takové aplikace, které jsou nativně určeny pro běh na operačním systému uživatelského počítače. Nativně spouštěná aplikace je, ať už přímo, či nepřímo, závislá na programových vlastnostech operačního systému daného stroje, stejně jako zprostředkovaně na jeho technickém vybavení. Přístupy k distribuci desktopových aplikací jsou v zásadě dva; první, z evidentních důvodů běžný především pro operační systém Windows, spočívá v distribuci již skompilovaných binárních souborů, které zajistí instalaci daného programu do systému a uživateli zbývá jen příslušný program spustit. Druhý způsob, běžný spíše pro různé linuxové distribuce, je založený na distribuci souborů obsahujících zdrojový kód programu, přičemž uživatel je nucen, má-li zájem program používat, si binární spustitelný soubor skompilovat. Druhý způsob pro programátora skýtá zásadní výhodu v tom, že se nemusí zajímat o prostředí, na němž bude koncový uživatel jeho program provozovat, protože kompilace programu pro všechny dostupné konfigurace (kombinace programového a technického vybavení koncového uživatele) je náročná na zdroje. [9] Pro onoho koncového uživatele však slyne druhá varianta tou nevýhodou, že kompilace programového vybavení vyžaduje z jeho strany netriviální usilí a znalosti o vlastním systému.

V rámci rozhraní určených pro zprostředkování dat z wordnetů jsou desktopové aplikace nejméně zastoupeny. Je tomu pravděpodobně proto, že jsou nejnáročnější na vývoj, zvláště mají-li být dostupné ze všech majoritních desktopových platforem. Dalším důvodem může být fakt, že často bývá zvolen model klient–server, pro který jsou vhodnější spíše aplikace v Javě [23, s. 13] a webová rozhraní.



Obrázek 6.1: Ukázka rozhraní aplikace Artha

Artha

Artha je multiplatformní thesaurus založený na WordNetu 3.0 [50]. Grafické prostředí aplikace je vytvořeno v GTK¹, takže umožňuje (minimálně na testovací konfiguraci, tedy operační systém Ubuntu 16.04) komfortní práci. Je možno ji nainstalovat z repozitářů (tedy nevyžaduje náročnou kompilaci), což je pro koncového uživatele velmi pozitivní. Dle dokumentace je k dispozici instalátor i na operační systém Windows². O Apple macOS³ se dokumentace nezmiňuje. Funguje bez připojení k Internetu a zobrazení je zaměřeno především na zobrazení významů synsetů (význam jsa reprezentován definicemi). Problémem je ukládání stavů aplikace, který se projevuje na nepříliš propracované funkcionalitě možnosti kroku zpět (na předchozí vyhledávání). Pokud například uživatel dvojklikne na slovo v definici, Artha se jej pokusí⁴ přesměrovat na definici tohoto slova ve WordNetu, ale při

¹multiplatformní soubor nástrojů pro tvorbu grafických rozhraní (<https://www.gtk.org/>)

²<https://www.microsoft.com/en-us/windows/>

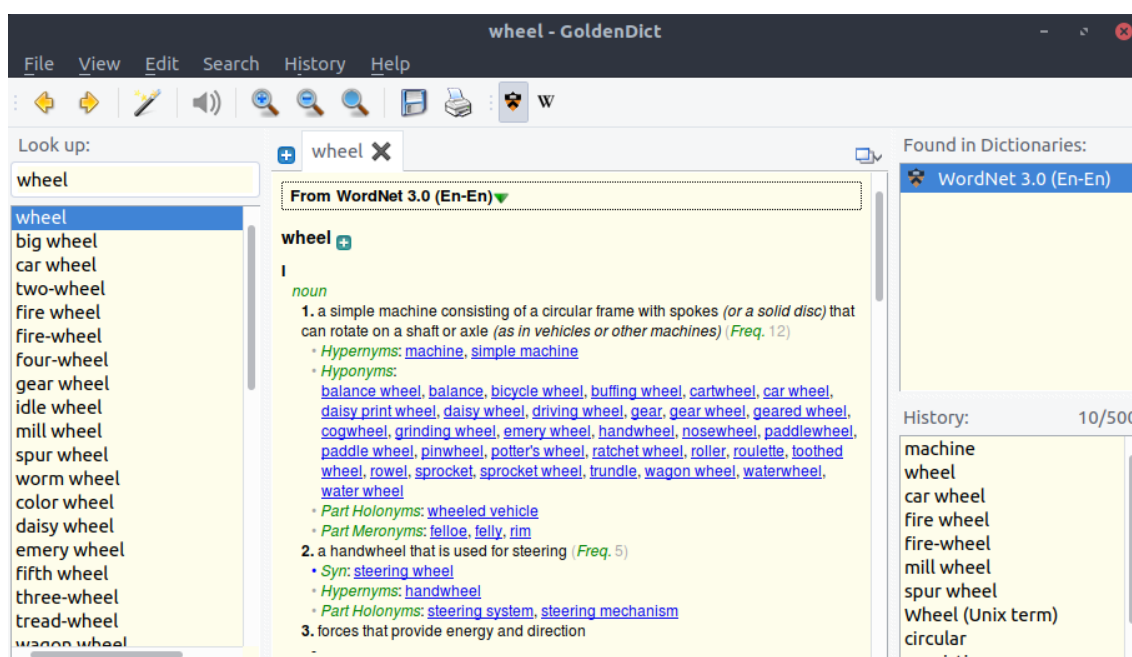
³<https://www.apple.com/lae/macos>

⁴selže, pokud uživatel vybere před dvojklikem více slov

kroku zpět už mu zobrazí slovo které hledal *před* slovem, z něhož se dostal na *současnou* definici. Oproti základnímu rozhraní WordNetu nabízí Artha větší pohodlí v podobě desktopové offline aplikace, což pro uživatele například znamená, že není při používání tohoto rozhraní závislý na připojení k Internetu.

Artha zobrazuje data čistě textově (ve smyslu, že nenabízí žádnou formu vizualizace), přičemž rozhraní dělí na dvě části, a to nalezené synsety a sémantické vztahy, které se k danému synsetu vážou (ilustrováno na snímku obrazovky 6.1 na straně 52).

GoldenDict



Obrázek 6.2: Ukázka rozhraní aplikace GoldenDict

Podobně jako Artha funguje GoldenDict, který, byvše univerzálnějším rozhraním pro více lexikografických zdrojů, byl testován jen ve své verzi pro WordNet. GoldenDict zobrazuje čistě textovou reprezentaci umožňující kliknutím zobrazit detaily jednotlivých synsetů. Slovníková data si obstarává aplikace sama, takže uživatel nemusí podstupovat instalaci dat princetonského WordNet (která může být netriviální, protože například pro Windows neposkytují oficiální webové stránky instalátor pro data WordNetu ve verzi 3). Zároveň však rozhraní umožňuje přidávat další slovníky a v základu umožňuje zobrazovat výsledku hledání z Wikipedie.

Rozhraní obsahuje levý sloupec pro zobrazování výsledků hledání a společně s hlavní plochou pro zobrazování detailů synsetu se chová podobně jako ostatní rozhraní (ilustrováno na snímku obrazovky 6.2 na straně 53). Aplikace udržuje historii hledání, umožňuje hledat slova z definic synsetů pomocí dvojkliku kursoru myši a dává uživateli k dispozici relativně široké možnosti konfigurovatelnosti. Její rozhraní se dobře přizpůsobuje velikosti obrazovky (potažmo okna).

Grafické rozhraní aplikace GoldenDict je postaveno na Qt⁵ a pro zobrazování dat z webu používá vykreslovací jádro WebKit⁶. [30]

⁵<https://www.qt.io/ui/>

⁶<https://webkit.org/>

Kapitola 7

Shrnutí přehledu rozhraní a vyplývající závěry o vhodném rozhraní k sémantickým sítím

Rozhraní v přehledu prezentovaném v této práci byla hodnocena dle různých aspektů odpovídajícím účelu a provedení dané třídy rozhraní. Nejpodrobněji byla hodnocena webová rozhraní. U těch byla jako největší společný nedostatek identifikována nedostatečná responsivita, což vede k tomu, že uživatel dané rozhraní nemůže (nebo může velmi omezeně) používat na mobilním zařízení. To v době, kdy počet uživatelů přistupujících k webovým službám přes mobilní zařízení je vyšší než počet těch, kteří k nim přistupují pomocí klasických počítačů [26], aplikaci značně diskriminuje. Z podobného důvodu jsou v nevýhodě také rozhraní vytvořená pro aplikační prostředí Java a v ještě větší míře aplikace vytvořené přímo pro operační systémy klasických počítačů.

Aplikace vytvořené v Javě jsou sice teoreticky spustitelné i na některých mobilních operačních systémech (zejména Android [27], u ostatních je podpora omezená či žádná (Amado [2], Gizmo [20] inter alia), prakticky je jejich použitelnost vzhledem k jejich designu grafického rozhraní omezena ale také výhradně na klasické počítače.

Pro některé aplikace vyvinuté v Javě byly vytvořeny tzv. applety Java¹, ale vzhledem k tomu, že některé silně rozšířené webové prohlížeče tento druh zásuvných modulů už vůbec nepodporují [40], je jejich použití značně omezené a do budoucnosti lze předpokládat plošnou nepoužitelnost.

Rozhraní závislá na platformě (operačním systému, případně aplikačním rozhraní) mají navíc tu nevýhodu, že uživatel nalezená data nemůže jednoduše sdílet například se

¹<https://nlp.fi.muni.cz/projekty/visualbrowser/applet/index.html>

svými kolegy (od toho u webových rozhraní hodnocené reflektování stavu aplikace v adresním řádku prohlížeče).

Z výše uvedeného je zřejmé, že je potřeba rozhraní, které bude použitelné univerzálně na všech platformách, a zároveň bude implementovat funkcionalitu všeobecně očekávanou u rozhraní pro wordnet. Tou je jednak textová prezentace dat a k ní dodatečná vizualizace ve formě grafu. Ideální rozhraní by také mělo splňovat designové požadavky moderních webových aplikací na přístupnost, tedy například nepředpokládat určitou velikost obrazovky uživatelského zařízení či ctít zásady vizuální přístupnosti (kontrastní barevné kódování, nepřekrývání textových prvků na obrazovce etc.). Je samozřejmě evidentní, že grafická prezentace dat nemůže být přizpůsobena velikosti obrazovky mobilního zařízení, mají-li být textové prvky čitelné a zároveň zachované rozložení prvků, může však být alespoň zajištěno, že funkcionalita posouvání zobrazovací plochy dotykem (u dotykových zařízení) není znemožněna funkcionalitou svázanou s prvky na obrazovce (jak tomu často bývá u implementací grafů synsetů, jejichž uzly je možné tažením myši přesouvat po zobrazovací ploše).

Kýžené rozhraní by také mělo splňovat požadavky na vizuální atraktivitu, aby se uživatelé cítili komfortně při práci s takovým rozhraním. Aspekty vizuálně komfortního rozhraní jsou do značné míry subjektivní, zdá se však, že uživatelé očekávají jistou prototypičnost designového konceptu od webové stránky s určitým účelem [66, 58]. Studie [58] také potvrzuje, že uživatelé pozitivně vnímají webové stránky s nízkou komplexitou, což může u implementace rozhraní wordnetu být problémem, jelikož takové rozhraní inherentně musí poskytovat velké množství informací. Cílem tedy je zobrazované informace vhodně strukturovat tak, aby uživatel měl pocit, že danou stránku zná a ví, kde na ní co najde, a necítil se zahlcen množstvím informací. Detaily designu takové aplikace jsou rozebrány v kapitole *Návrh rozhraní* a následujících.

Část III

Praktická část: rozhraní k sémantické síti

Kapitola 8

Návrh rozhraní

Východiska

Cílem praktické části této práce bylo vytvořit rozhraní k sémantickým sítím typu wordnet, které by nebylo zatíženo nedostatky, jež byly pozorovány u rozhraní testovaných v předchozí části práce. Zřejmě nejrozšířenějším problémem u existujících rozhraní pro webové prohlížeče je jejich do různé míry značná problematičnost použitelnosti na mobilních zařízeních s malým displejem (u některých naprostá nepoužitelnost). Rozhraní implementovaná ve formě aplikace pro klasické počítače jsou na mobilní zařízení nepřenositelná již ze své podstaty a jejich použitelnost na těchto zařízeních nemůže být hodnocena vůbec, faktem však zůstává, že je jejich nepřenositelnost v porovnání s webovými rozhraními diskriminuje.

Rozhraní, které je výstupem této práce si tedy klade za cíl umožnit uživatelům přístup k datům wordnetů odkudkoliv, bez ohledu na to, zda pracují na mobilním telefonu, tabletu či klasickém počítači. Jeho návrh se snaží být technologicky nadčasový, ačkoliv v tomto aspektu budou ve světě webových technologií, jenž se neustále vyvíjí, pravděpodobně vždy existovat jisté limity. Pro rozhraní vytvořené v rámci této práce to jsou však v podstatě pouze požadavky na webový prohlížeč koncového uživatele. Uživatelův prohlížeč musí podporovat technologie ECMAScript verze 6¹, HTML verze 5 a CSS verze 3. Podrobněji se vysvětlení požadavků na verze věnuje kapitola *Použité technologie* na straně 64.

Rozhraní bylo v rámci této práce navrženo tak, aby umožňovalo zobrazení dat jak textové, tak grafické. Textové zobrazení je vhodné pro zobrazení synsetů, které obsahují velké množství potomků (hyponyma, meronyma, etc.), a mohou v grafickém rozhraní být nepřehledné. Také se textové rozhraní spíše hodí k použití na mobilních zařízeních, jelikož

možno
jen 5, ale
zatím to
na něm
nefunguje

¹také známý jako ECMAScript 2015; všeobecně označován jako JavaScript, čehož se bude nadále držet i tato práce

se lépe přizpůsobuje velikosti obrazovky. Grafické zobrazení je naopak vhodné pro rychlý přehled míry komplexity synsetu a jeho vztahů. Může sloužit především jako ilustrace daného slovního významu. Textové a grafické zobrazení se vzájemně nedoplňují, ale jsou různým zobrazením týchž dat a je na uživateli, které ze zobrazení je pro něj komfortnější či užitečnější.

Cíle nového rozhraní

Přesto, že existujících rozhraní je dostupných mnoho, žádné z nich nesplňuje požadavky moderní platformně nezávislé aplikace, jak bylo ukázáno v části *Přehled a porovnání existujících vizualizací sémantických sítí*. Proto vznikl záměr vytvořit takové rozhraní, které by tyto požadavky (do co největší míry) splňovalo. Vzhledem k původním záměrům samotného WordNetu (a na něj navazujících projektů), mezi nimiž byly například edukativní účely, je nutné, aby rozhraní k těmto projektům bylo intuitivní, vizuálně přitažlivé a snadno přístupné. Snadná přístupnost například minimalizuje nutnost správné přípravy školních počítačů k použití takového rozhraní, což opět zvyšuje šance, že aplikace bude skutečně použita.

S touto vizí tedy byla vytvořena webová aplikace *NSFW viewer*, která je v dalších částech této práce prezentována.

Kapitola 9

Prezentace vytvořeného rozhraní

Základní představení

Rozhraní *NSFW viewer* uživateli při prvním otevření prezentuje prázdnou stránku s vyhledávacím formulářem v levém sloupci a nápovědou, jak začít s rozhraním pracovat, v pravém prostoru určeném k následné prezentaci dat.¹

Vyhledávací formulář sestává jednak ze samotného pole pro vložení hledaného výrazu a tlačítka na spuštění vyhledávání, a jednak z rozbalovací nabídky, v níž může uživatel vybrat zdroj dat vyhledávání (typicky různojazyčné wordnety). Nad polem pro vkládání jsou také tlačítka určená k volbě reprezentace synsetu.

Pokud je zadané slovo po spuštění vyhledávání úspěšně nalezeno, zobrazí se uživateli v levém sloupci pod vyhledávacím formulářem nabídka synsetů, v nichž se hledaný výraz vyskytoval, a v pravém prostoru podle zvolené reprezentace buďto textová data o zvoleném (případně prvním) synsetu, či jeho reprezentace grafem.

Při přepínání vizualizace dat mezi textovou a grafickou se mění pouze obsah pravé části rozhraní, obsah levého sloupce s vyhledávacím polem a výpisem zobrazených synsetů zůstává neměnný.

Zadá-li uživatel výraz, který se v daném wordnetu nevyskytuje, je mu po neúspěšném dokončení vyhledávání prezentována zpráva o negativním výsledku hledání.

¹Při zobrazení rozhraní na displeji užším než 768 pixelů se z levého sloupce stane horní část stránky, k pravému prostoru se pak uživatel dostane rolováním po stránce dolů. Při popisu rozhraní bude k těmto částem nadále referováno jako k levé a pravé z důvodu přehlednosti.

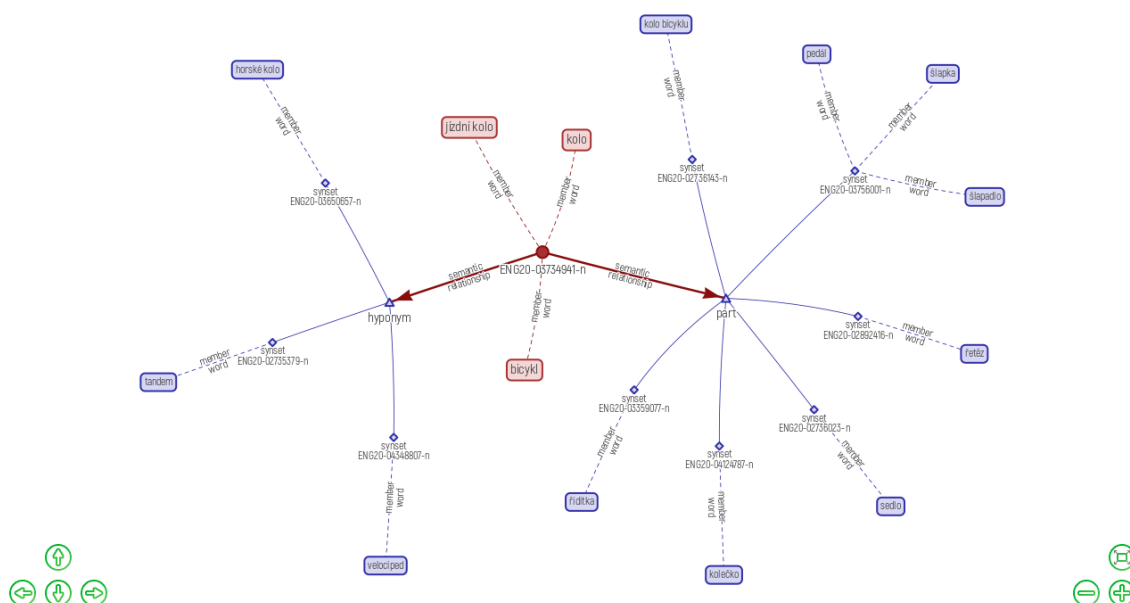
Textová reprezentace

Textový režim reprezentace dat je až na několik odchylek relativně konservativní a snaží se uživateli přinést přehledný a zároveň komplexní výpis nalezených informací.

Hlavní nadpis tvoří výpis slov patřících do zobrazeného synsetu, pod ním je zobrazena definice významu, cesta a další informace, jako například slovní druh či číslo základního konceptu (BCS). Cestou k synsetu je míněn seřazený seznam synsetů, přes něž je možnost se ze zobrazeného synsetu po relacích hyperonymie dostat ke kořenovému synsetu. Pod tímto blokem s informacemi uživatel nalezne výpis synsetů, s nimiž je ten zobrazený provázán sémantickými relacemi. Jednotlivé vztahy jsou rozděleny do sloupců, což přispívá k lepšímu škálování reprezentace sémantických relací. Uživateli v této sekci není zobrazena jedna důležitá sémantická relace, a to hyperonymie; synsety k zobrazenému synsetu nadřazené jsou totiž zobrazeny ve zmíněné cestě.

Aby byla zvýšena interaktivita rozhraní, jsou slovní formy v hlavním nadpisu a synsety v sekci sémantických relací uživateli vypisovány jako hypertextové odkazy. Přes tyto odkazy je možné vyhledat dané slovo či přejít na příslušný synset, čímž je umožněno procházení wordnetu přes vztahy jako jsou hyponymie, meronymie, hyperonymie etc. V textovém zobrazení se vyskytují dva druhy hypertextových odkazů. První jsou v levém sloupci v seznamu synsetů nalezených při vyhledávání a slouží k volbě zobrazeného synsetu. Vzhledem k tomu, že server po vyhledávání vrací soubor všech nalezených synsetů, nevyvolává přechod po těchto odkazech nové hledání. Druhým druhem odkazů jsou zde takové, které neodkazují na data načtená po posledním vyhledávání. Vyskytují se právě části textového rozhraní, a to v hlavním nadpisu, v cestě k zobrazenému synsetu a v sémantických relacích, do nichž tento náleží. Pokud na ně uživatel klikne, spustí se nové vyhledávání podle slovní formy nebo identifikačního řetězce synsetu, na nějž uživatel klikl. Výsledkem hledání dle identifikačního řetězce jeden konkrétní synset.

Jednou z odchylek od tradičních postupů v textové reprezentaci dat z wordnetu je udávání příslušnosti slovní formy k významu. Tato příslušnost je tradičně reprezentována číslem za slovní formou odděleným dvojtečkou. To však může pro uživatele neznalého tradice v reprezentaci wordnetů být matoucí, a tak bylo v rámci zlepšení přístupnosti nového rozhraní *NSFW viewer* přistoupeno k zobrazení čísla významu jako horního indexu za slovní formou. To by mělo údaj učinit méně obtrusivní a intuitivně snáze pochopitelný, jelikož na horní indexy bude uživatel alespoň vizuálně spíše zvyklý než na čísla za dvojtečkou.



Obrázek 9.1: Grafická reprezentace dat v NSFW viewer

Grafická reprezentace

Grafická reprezentace nalezených dat je alternativou k textové a jejím cílem je uživateli především vizualizovat vztahy, jichž je prohlížený synset členem. Podobně jako textová reprezentace se snaží příliš se neodchylovat způsobem zobrazení od již existujících zobrazení, ale zároveň eliminovat nedostatky v použitelnosti, jimiž se většina existujících implementací vyznačuje. Jmenovitě grafické zobrazení v NSFW viewer vůbec nepoužívá interaktivní prvky typu informačních bublin, jež se zobrazují při přejetí myší². Tyto bubliny jsou velmi oblíbené zřejmě proto, že umožňují bez ztráty přehlednosti celého grafu uživateli přístup k podrobnějším informacím. Jejich zásadní nevýhoda ovšem tkví v tom, že na dotykových obrazovkách, jimiž bývají vybavena současná mobilní zařízení, koncept přejetí myši přes prvek neexistuje. To je z toho důvodu, že grafické rozhraní na těchto zařízeních nebývá ovládáno externím technickým vybavením počítače typu myš, které pohybuje po obrazovce kurzorem, ale uživatel používá dotyku svých prstů či stylu a dotek na určitém místě obrazovky znamená obvykle kliknutí. Nelze tedy tak jako u klasických počítačů rozlišit přejetí kurzorem myši a kliknutí a funkcionalita zmíněných informačních bublin je tedy buďto zcela nepřístupná či přístupná jen velmi omezeně.

²tyto bývají anglicky označovány jako *popup tooltip*

Další interaktivita značně rozšířená u existujících vizualizací je možnost přetažením uzlu myší přeuspořádat graf. U některých z vizualizací může taková možnost být užitečná proto, že při větším počtu uzlů se některé uzly překrývají a přetažením lze takový překryv eliminovat (většinou však na úkor rozložení jiné části grafu). Obecně by však při správném nastavení zobrazení taková funkcionality neměla být potřebná, a proto nebyla zavedena ani do vizualizace v *NSFW viewer*.

Koncept zobrazení grafem spočívá v použití tří druhů uzlů s odlišným významem, které jsou spojeny hranami, jejichž význam je závislý na tom, které uzly spojují. Prvním druhem uzlů jsou ty, jež reprezentují synsety. Takový uzel se v grafu vždy vyskytuje minimálně jeden kořenový, což je v tomto případě ten synset, jehož detaily jsou otevřeny. Další jsou pak členové případných sémantických relací, jichž je kořenový členem. Z kořenového uzlu vedou hrany ke dvěma zbývajícím druhům uzlů, jimiž jsou slovní formy a sémantické relace. Uzly reprezentující slovní formy jsou se synsetem, k němuž náležejí, propojeny hranou pojmenovanou *member word*, tedy členské slovo, a žádné další hrany z nich nevedou.

Co se uzlů reprezentujících sémantické relace týče, tam může význam jejich přítomnosti v grafu být na první pohled poněkud obskurnější, důvody k jeho použití jsou však jednoduché. Intuitivně by sémantická relace měla být hrana, jež vede mezi synsety spojenými touto relací. Pojmenování hrany by se pak odvíjelo od druhu relace, již reprezentuje. Problémem s tímto způsobem zobrazení tkví v tom, že značně centralizuje zobrazení do hvězdčovitěho obrazce okolo kořenového uzlu, což zobrazované uzly vizuálně uspořádá do hierarchicky ploché struktury a sníží přehlednost grafu. Z tohoto důvodu byly do vizualizace grafem zavedeny uzly reprezentující sémantické relace, které synsety svázané s kořenovým uzlem stejným vztahem sdružují do jednoho podstromu. Díky tomu je dosaženo rozčlenění grafu na jednotlivé sekce podle těchto sémantických relací a uživatel není nucen zkoumat význam každé hrany mezi synsety (cf. vizualizace v rozhraní Cornetto, ilustrace 4.4 na straně 37).

Responsivita vizualizace je inherentně omezena tím, že se graf byť i nepříliš komplexního synsetu nemůže na malou obrazovku mobilního zařízení vejít, zvláště mají-li být textové popisy uzlů čitelné. Tím, že vizualizace umožňuje přibližování a oddalování, je však tento problém do jisté míry eliminován, protože uživatel může graf přiblížit a přejíždět po něm podobně, jako to umožňují například aplikace na zobrazování elektronických map³. Toto řešení samozřejmě zavádí na dotykových obrazovkách problém s tím, aby nebylo zneemožněno rolování po zbytku stránky. To bylo vyřešeno v *NSFW viewer* zachováním okrajů po stranách kolem plochy s grafem, kde se běžně vyskytují rolovací ovládací lišty, a uživatel tak tohoto okraje může využít k rolování stránky namísto posouvání grafu.

³<https://www.mapy.cz> inter alia

Kapitola 10

Použité technologie

Rozhraní *NSFW viewer* je vytvořeno jako aplikace pro webové prohlížeče, přesněji řečeno jde o dokument v HTML¹ (k čemuž se všeobecně referuje jako k webové stránce), jenž je dynamicky generovaný na straně klienta pomocí skriptovacího jazyka JavaScript a jím zpracovávaných dat ze serveru posílaných ve formátu JSON². Vzhled dokumentu je definován pomocí jazyka CSS³. Jednotlivé role uvedených technologií budou rozebrány v následujících sekcích.

HTML

Značkovací jazyk HTML je základním stavebním prvkem většiny aplikací pro webové prohlížeče. Jako značkovací jazyk umožňuje HTML strukturalizaci textového dokumentu do částí, definici významů jednotlivých částí (například označení nadpisu) a do jisté míry jej lze užívat, byť se to nedoporučuje, i pro definici vzhledu textu (tučný řez písma, etc.). Poskytuje také prostředky pro tzv. hypertextové odkazy, jimiž lze provázat různé dokumenty či části jednoho dokumentu. Nezřídka se tyto odkazy v dnešní době používají jako ovládací prvky k řízení změn obsahu dynamicky generovaného dokumentu, kterým je například zde popisované rozhraní *NSFW viewer*.

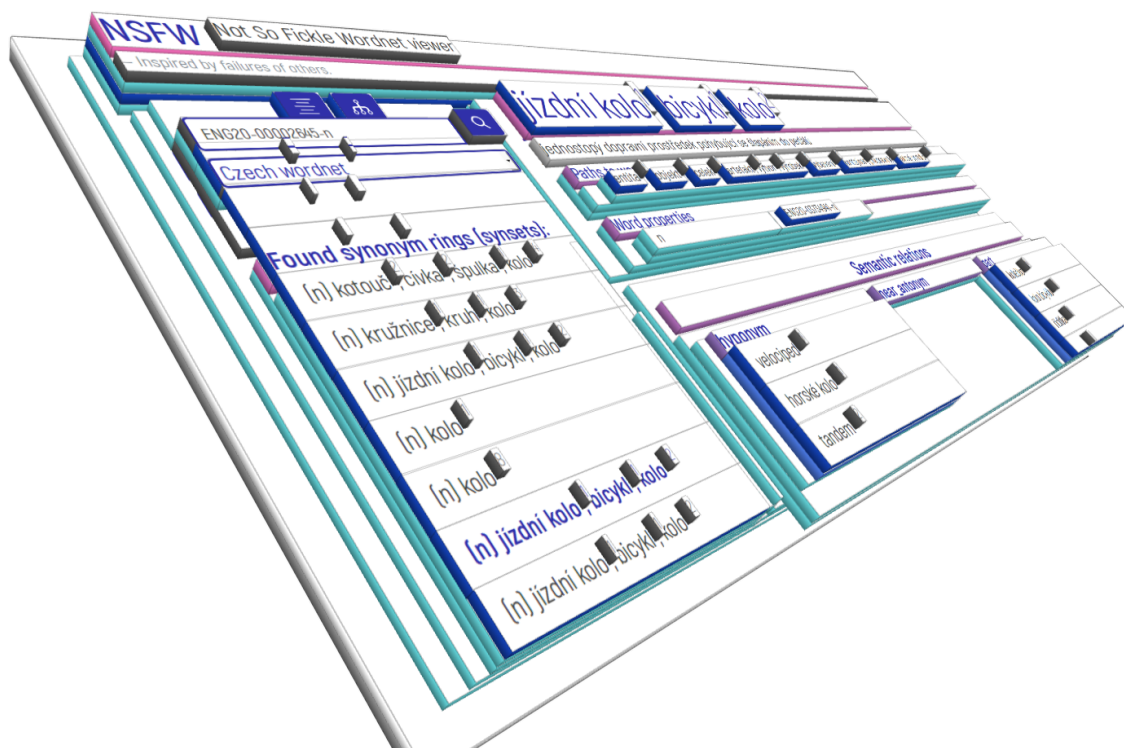
V rozhraní *NSFW viewer* HTML slouží k definici struktury zobrazené webové stránky, jejímu rozdělení na jednotlivé sekce (levý sloupec, pravá plocha, záhlaví, etc.) a ke strukturalizaci posléze zobrazovaných informací. Příkladem může být výpis sémantických relací, v nichž je zobrazovaný synset přítomen. Jednotlivé relace jsou rozděleny do bloků (jež jsou

¹HyperText Markup Language

²JavaScript Object Notation

³Cascading Style Sheets

později pomocí CSS nastaveny tak, aby se zobrazovaly jako sloupce) a jednotlivé synsety svázané se zobrazovaným příslušným vztahem jsou vypsané v seznamu.



Obrázek 10.1: Třídímenzionální vizualizace struktury rozhraní

Pátá verze HTML je v aplikaci využita jakožto nejnovější s ohledem na to, že je podporovaná ve všech moderních prohlížečích. [64] Důležitou motivací pro použití této verze je také podpora prvku `<canvas>`, který umožňuje vykreslovat do dokumentu grafiku [63] a je pomocí něho implementována grafická reprezentace dat.

JavaScript

JavaScript je vysokoúrovňový⁴ dynamický⁵ netypovaný⁶ a interpretovaný⁷ programovací jazyk používaný vedle HTML a CSS jako jedna ze tří základních technologií pro tvorbu

⁴silně abstrahovaný od nízkourovňových instrukcí procesoru a nezávislý (či méně závislý) na technickém vybavení počítače

⁵úkony, jež jsou u statických jazyků prováděny při kompilaci, jsou prováděny za běhu programu

⁶kterékoli operace jsou povoleny nad jakýmkoli daty, nekontroluje se typ proměnných

⁷program spouštěný bez předchozí statické kompilace, interpretem následně rozložený na kompilované podrutiny

webových aplikací. Jeho funkce spočívá v zásadě v ovládání obsahu stránky na základě událostí, kterými mohou být například akce uživatele jako kliknutí na odkaz či akce serveru jako je dokončení požadavku a navrácení jeho výsledků.

Popisované rozhraní momentálně vyžaduje JavaScript (resp. ECMAScript) ve verzi 6 a využívá jej na sledování akcí uživatele (inicializace hledání a přepínání zobrazeného synsetu či vizualizace), odesílání požadavků uživatele serveru (hledání), přijímání odpovědi na tyto požadavky a následné zpracování odpovědi a aktualizace HTML dokumentu, tedy zobrazení výsledků hledání uživateli. Zjednodušený vývojový diagram této logiky je ukázán na ilustraci 11.1 na straně 70.

JSON

JSON je formát určený pro výměnu dat a je zároveň strojově i lidsky čitelný (jedná se o čistý text). Strukturálně se podobá objektům v JavaScriptu, tedy sestává z párů klíč : hodnota. Klíčem bývá řetězec znaků, hodnotou může být cokoliv, co je serializovatelné. [8]

Tento formát je vhodný na přenos informací, jakými jsou data z wordnetu, jelikož se jedná o strukturovaná, čistě textová data. Další možností pro přenos takových dat je například formát XML⁸. Ten oproti formátu JSON umožňuje přesnější specifikace typů hodnot a validaci dat, avšak pro stejné množství informací je datově větší (má větší režii na značkování). XML je navíc oproti formátu JSON méně pohodlně zpracovatelné v JavaScriptu, jelikož syntax formátu JSON reflektuje strukturu objektů v JavaScriptu. Vzhledem k tomu, že data z wordnetů bývají krátké řetězce, je evidentní, že při použití XML by bylo přenášeno zbytečně velké množství strukturních informací v podobě značkování. [65] To v kombinaci s jednoduchostí zpracování formátu JSON v JavaScriptu přispělo k rozhodnutí použít v aplikaci *NSFW viewer* k přenosu dat JSON. Ukázka formátu JSON je v kapitole *Struktura odpovědi serveru* na straně 71

Ajax

Ajax⁹ je souhrnné označení skupiny technologií používaných pro asynchronní přenos dat mezi webovým prohlížečem a serverem, tedy pro komunikaci webových aplikací se serverem. Rozdíl oproti tradičnímu přenosu informací mezi serverem a webovou stránkou načtenou u klienta tkví v tom, že bez použití Ajaxu je nutné pro zpracování odpovědi na požadavek načíst stránku znova. Hlavním využitím Ajaxu bývají právě aplikace, jejichž funkčnost je závislá na komunikaci se vzdáleným serverem bez nutnosti načítat celou webovou stránku znovu. [16]

⁸eXtensible Markup Language

⁹Asynchronous JavaScript And XML

U komplexních webových aplikací může být načítání celé stránky znova nevýhodné z mnoha důvodů. Jedním ze scénářů může být například potřeba obnovit data na části stránky, zatímco uživatel vkládá data (například komentář) na jiné části¹⁰. Úplným znovunačtením by byla uživatelská vložená data ztracena, takže je nutné obnovovat jen její část, k čemuž je výhodné využít Ajax.

Dalším důvodem k využívání Ajaxu je uživatelský komfort – je možné během načítání zobrazovat vlastní indikátor načítání, při výpadku spojení zobrazit uživatelsky přívětivou zprávu, etc.

Asynchronní výměna dat mezi webovým prohlížečem a serverem probíhá pomocí aplikačního rozhraní XMLHttpRequest (XHR) dostupného prohlížečům pomocí JavaScriptu. [39]

Historie v prohlížeči

Historie je schopnost webových prohlížečů přecházet mezi jednotlivými stavy, které nastaly během uživatelského užívání daného okna (či karty) prohlížeče. Klasicky jsou těmito stavy jednotlivé stránky, které uživatel navštívil, ať už v rámci jednoho nebo různých portálů. Pokud například navštíví domovskou stránku vyhledávače DuckDuckGo¹¹, vyhledá výraz brno a následně otevře například výsledek hledání vedoucí na portál Wikipedia¹², zaznamenaná prohlížeč automaticky tři stavy (domovská stránka vyhledávače, stránka s výsledky hledání, stránka na Wikipedii), k nimž je možné se vrátit. Pro porovnání s nastíněním problému, jež následuje níže, je dlužno podotknout, že v obou zmíněných přechodech mezi stavy karty prohlížeče se stránka načítá zcela znovu¹³, takže se změna stavu zaznamenává automaticky (netřeba ji ošetřovat na straně aplikace).

Při užívání dynamického generování obsahu stránky na straně klienta a přenosu požadavků a dat pomocí Ajaxu nastává problém, že z hlediska prohlížeče se stav nezmění, tudíž není nic zapsáno do historie. Tento problém byl v minulosti řešen různými způsoby, pátá verze HTML však implementuje aplikační rozhraní pro možnost ovládat uměle historii prohlížeče. Je tedy možné v libovolný okamžik běhu JavaScriptu přidat do historie další stav a tím umožnit jeho obnovení například při uživatelské akci „přejít zpět“.

Druhým problémem při dynamickém generování obsahu stránky je udržování adresního řádku prohlížeče, který obsahuje URL¹⁴, aby bylo později možné z adresy obnovit stav; to je výhodné zejména z toho důvodu, že by mělo být možné adresu uložit či sdílet. Ja-

¹⁰ Existuje jistě možnost toto obejít pomocí rámců, ale ty mají mnoho zásadních nevýhod.

¹¹ <https://duckduckgo.com>

¹² <https://en.wikipedia.org/wiki/Brno>

¹³ stav k 1. 6. 2017

¹⁴ Uniform Resource Locator

vaScript umožňuje nativně manipulaci s adresou v adresním řádku, přičemž *NSFW viewer* její udržování implementuje pomocí knihovny *URI.js*. [52]

Použité knihovny a zásuvné moduly pro JavaScript

jQuery je knihovna navržená s cílem usnadnit programování v JavaScriptu. Je zaměřená především na zjednodušení práce s elementy v HTML dokumentu, práci s událostmi (např. stisk klávesy) a komunikaci mezi klientem a serverem pomocí Ajaxu. [31]

URI.js zjednodušuje práci s adresami (URL). Knihovna *URI.js* umožňuje jednak obsah adresního řádku zpracovávat pro další práci, a jednak jej umožňuje nastavovat tak, aby odpovídal stavu, v němž se aplikace nachází. [52]

Perfect Scrollbar je zásuvný modul určený k zobrazování rolovacích lišt, jež jsou flexibilnější než základní systémové lišty a nezávislé na operačním systému. Dají se na rozdíl od těch systémových mimo jiné vzhledově přizpůsobit zbytku aplikace, a tudíž nepůsobí tak obtrusivně. [29]

Vis.js je vizualizační knihovna, navržená pro zobrazení velkého množství dynamicky generovaných dat. Umožňuje navíc pokročilou manipulaci s daty, jako je přidávání a odebírání uzlů grafu, změny štítků uzlů, etc. V *NSFW viewer* je využit její modul *Network* k zobrazení detailů synsetu grafem. [1]

CSS

CSS, neboli kaskádové styly, je jazyk pro definování vzhledu prvků (obvykle náležejících do HTML dokumentu). Pojem kaskádový má tento jazyk v názvu proto, že vlastnosti prvků jsou děděny od rodičů a podle dědičnosti je prvků možné i vybírat. Prvotním smyslem užívání CSS bylo oddělení obsahu webové stránky od jejího vzhledu. To je podstatné pro udržitelnou správu stránky, větší flexibilitu a sémantiku zdrojového kódu. Použití CSS umožňuje zobrazit jednu stránku různými způsoby, jedním z nichž může být například vzhled přizpůsobený pro tisk či pro barvoslepe uživatele.

Kaskádové styly jsou v *NSFW viewer* tedy využity k tomu, aby byla stránka s aplikací uživatelsky přívětivá, atraktivní a vhodně zobrazena na zařízeních různých velikostí (částečně je využito stylování podmíněného šířkou obrazovky). Třetí verze CSS je využita proto, že je nejnovější, je vyžadována pro kvalitní implementaci responsivity rozhraní a také ji jako takovou využívá Bootstrap.

Bootstrap

Aplikace *NSFW viewer* využívá Bootstrap¹⁵, což je sada nástrojů v HTML a CSS pro tvorbu webových stránek. Obsahuje šablony různých prvků, které se na webových stránkách vyskytují, definuje základní vlastnosti některých prvků, aby rozložení stránky bylo responzivní a mimo jiné také do jisté míry zajišťuje konformní vzhled napříč různými prohlížeči.

¹⁵<http://getbootstrap.com/>

Implementace rozhraní

```

graph TD
    A[načte rozhraní] --> B{obsahuje URL parametry?}
    B -- ano --> C{jsou v URL známé parametry?}
    B -- ne --> D[uvede rozhraní do výchozího stavu]
    C -- ano --> E{vyhl. pole prázdné?}
    C -- ne --> F[zobrazí data textově]
    E -- ano --> G[upozorní uživatele]
    E -- ne --> H[zahájí vyhledávání]
    H --> I{hledání úspěšné?}
    I -- ano --> J{je v URL ID synsetu?}
    I -- ne --> G
    J -- ano --> K[přepne vizualizaci]
    J -- ne --> L{přepne synset}
    L --> M{je v URL režim?}
    M -- ano --> N[použije ID prvního synsetu]
    M -- ne --> O{který?}
    O -- text --> F
    O -- graph --> P[zobrazí data grafem]
    N --> M
    K --> M
    L --> M
  
```

Obrázek 11.1: Zjednodušený vývojový diagram logiky rozhraní

Získávání dat

Data, jež jsou ve výsledku zobrazována uživateli, jsou načítána ze serveru v závislosti na vstupu, který uživatel tím či oním způsobem zadal. Vstup uživatele může být dvojího druhu (byť z hlediska výsledného zpracování se nijak neliší). Prvním je, že uživatel zadává kýžené slovo do vyhledávacího pole, druhým, že klikne na odkaz vedoucí na jiný synset. Druhý případ se potom od prvního liší pouze v tom, že text ve vyhledávacím poli je buď identifikátor, nebo členské slovo synsetu¹, na něž bylo kliknuto. Jde tedy o jistou zkratku, během níž se automaticky zadá identifikátor daného synsetu do vyhledávacího pole a spustí se vyhledávání.

Parametry vyhledávání

Aby bylo možno vyhledávání spustit se správnými parametry, je nutno získat od uživatele dva údaje. Jedním je vyhledávané slovo, které je extrahováno z vyhledávacího pole, druhým je zdroj dat, v němž se má vyhledávat. Pro výběr druhého slouží rozbalovací nabídka pod vyhledávacím polem, která obsahuje seznam zdrojů dat, v nichž je možno vyhledávat.

Zde je na místě podotknout, že schopnost aplikačního rozhraní, jež zajišťuje samotné vyhledávání, vyhledávat jak podle slovní formy, tak dle identifikátoru, otevírá možnost vyhledávat konkrétní synset napříč dostupnými wordnety.

Po spuštění vyhledávání je na server s aplikačním rozhraním odeslán požadavek složený ze vstupních dat. Jeho forma je následující:

```
https://nlp.fi.muni.cz/~xrambous/fw/abulafia/ZDROJ?action=jsonvis&query=
SLOVO ,
```

kde ZDROJ zastupuje prohledávaný wordnet a SLOVO zastupuje vyhledávaný výraz.

Struktura odpovědi serveru

V případě úspěšného hledání je ze serveru přijata odpověď ve formátu JSON (bližší popis formátu v kapitole JSON na straně 66). Jeho strukturu lze popsat jako pole² objektů³, kde každý z těchto objektů reprezentuje jeden nalezený synset. Objekty synsetů pak obsahují jednak řetězec jako identifikátor, slovní druh či definici, a jednak další pole. Pole pod klíčem

¹podrobněji o tomto rozdílu v [KDE O TOM KURVA PISU?! nemuzu to najít]

²Seřazená množina prvků, v terminologii jiných jazyků například seznam (*list*)

³Objekt je množina dvojic klíč:hodnota, v terminologii jiných jazyků je tento datový typ nazýván slovníkem (*dictionary*), asociativním polem (*associative array*), etc.

paths slouží k ukládání cesty k danému synsetu, pole pod klíčem synset obsahuje jeho členská slova a pole s klíčem children⁴ pak obsahuje sémantické relace daného synsetu.

Struktura odpovědi může být ilustrována následujícím zjednodušenou ukázkou:

```

1  [
2    { // hlavní synset:
3      "id": "ENG20-02759263-n",
4      "pos": "n",
5      "def": "definice synsetu",
6      "synset": [
7        {
8          "name": "kotouč",
9          "meaning": "2"
10       },
11       ... // další členská slova synsetu
12     ], // cesty k synsetu:
13     "paths": [
14       {
15         "name": "path-1",
16         "breadcrumbs": [
17           {
18             "id": "ENG20-00020486-n",
19             "def": "",
20             "synset": [
21               {
22                 "name": "abstrakce",
23                 "meaning": "1"
24               },
25               ... // další členská slova synsetu
26             ]
27           },
28           ... // další části cesty
29         ]
30       },
31       ... // alternativní cesty
32     ],
33     "children": [
34       { // sémantické relace synsetu
35         "name": "hyponym",
36         "children": [
37           {
38             "id": "ENG20-04050919-n",
39             "def": "",
40             "synset": [
41               {
42                 "name": "člunek",
43                 "meaning": "1"
44               },
45               ... // další členská slova synsetu
46             ]
47           },
48           ... // další synsety v dané relaci
49         ]
50       },
51       ... // další sémantické relace
52     ],
53   },
54   ... // další nalezené synsety

```

⁴Klíč children se užívá všeobecně k označení pole s potomky.

55]

Pokud je vyhledávání neúspěšné, vrací server pouze prázdné pole. Tento stav je kontrolován aplikací a v případě, že nastane, je uživatel upozorněn, že dané hledání nevrátilo pro zvolený zdroj dat žádné výsledky.

Zpracování odpovědi

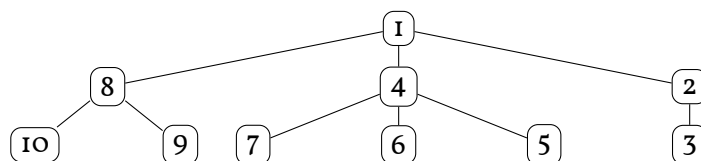
Získaná data jsou v několika krocích zpracována tak, aby jejich použití v aplikaci bylo jednodušší a jejich struktura odpovídala potřebám rozhraní. Jmenovitě je pole objektů se synsety serializováno do jednoho objektu, v němž lze odkazovat na potřebný synset univerzálnějším způsobem (podle jeho identifikátoru).

Textová reprezentace nevyžaduje další reorganizaci dat, ale zpracování odpovědi pro potřeby vizualizace grafem je poněkud komplexnější. Je evidentní, že pro vykreslení grafu je nutné mít k dispozici uzly a hrany, jimiž jsou uzly spojeny. V JSONu získaném ze serveru jsou tyto vztahy obsaženy pouze implicitně tak, že určitý objekt je rodičem dalších objektů, či dítětem výše postaveného objektu. Explicitní údaje o těchto vztazích ve vlastnostech objektů nejsou, a je tudíž nutné je pro knihovnu, která graf vykresluje, vytvořit. Pro tento účel obsahuje aplikace *NSFW viewer* funkci `DFSThroughSynsets()`, která prochází synsety v JSONu metodou procházení do hloubky⁵, každý nalezený synset zapisuje do seznamu uzlů a zároveň k němu do seznamu hran zapisuje uspořádanou dvojici, která sestává z identifikátoru jeho rodičovského uzlu a jeho vlastního identifikátoru. Pořadí, v jakém jsou jednotlivé uzly procházeny, je znázorněno na grafu 11.2 na straně 74. Procházení do hloubky je implementováno pomocí zásobníku a rekurse. Během procházení dat je pomocí různých atributů nalezených objektů rozlišováno pět druhů uzlů, a to kořenový synset (`root`), tedy ten, jež si uživatel vybral, jeho členská slova (`rootLeaf`), ostatní synsety (`synset`), jejich členská slova (`leaf`) a uzly představující sémantické relace (`semGroup`).

Zobrazení dat

Ve chvíli, kdy jsou data zpracována, může být jejich příslušná část, tedy jeden synset, zobrazena uživateli. Který synset se bude zobrazovat, je v případě, že v adrese (URL) existuje

⁵Procházení do hloubky (DFS, depth first search) je algoritmus pro prohledávání grafu, jenž prochází postupně celé větve grafu. Druhou možností je procházení do šířky (BFS, breadth first search), které prochází nejprve všechny uzly co nejbližší kořeni (tedy na stejné úrovni) a až poté přechází na nižší (hlubší) úroveň. [10] Z hlediska aplikace *NSFW viewer* se tyto dva přístupy liší jen tím, že vykreslují mírně odlišně uspořádané grafy (struktura je zachována)



Obrázek 11.2: Ilustrace pořadí, v jakém jsou uzly zapisovány při procházení grafu do šířky

parametr synset specifikující, rozhodnuto na základě onoho parametru, jinak je vybrán pro další operace objekt se synsetem, jenž se v datech vyskytuje jako první.

Zároveň je uživateli v levém sloupci zobrazen seznam nalezených synsetů, tedy výpis jejich členských slov. Každý řádek je zároveň odkazem, kliknutím na něj si uživatel může vybrat, kterou část dat si přeje prohlížet. Taková akce vyvolá výběr objektu, který obsahuje synset s identifikátorem shodným s tím, který byl uveden v odkazu.

Objekt obsahující příslušnou část dat je následně předán funkci `renderView()`, která na základě stavu aplikace uloženého v adrese zavolá funkci zobrazující data textovou reprezentací, či grafickou.

Textové zobrazení dat je řešeno jednoduchým naplňováním částečně předpřipraveného HTML dokumentu očekávanými daty. Rozhraní počítá s neúplností dat, takže není v tomto ohledu striktní a pokud se požadované informace v datech nevyskytují (typicky definice je hodnota, která u nezanedbatelného počtu konceptů v českém wordnetu chybí), je daná struktura naplněna prázdným řetězcem a nezobrazuje se. Kromě informací, jejichž existence je v datech nejistá, se v nich vyskytují i informace s předem neznámou délkou. Mezi takové patří například počet synsetů odpovídajících hledanému výrazu, pro každý z nich pak počet relací, jichž jsou členem, počet dalších synsetů v dané relaci či počet členských slov jednotlivých zobrazených synsetů. Zobrazení takových informací je v rozhraní řešeno cykly a z hlediska struktury HTML to bývají seznamy, jakožto prvky nejvhodnější pro výpis položek. Výjimkou jsou členská slova synsetů, ta jsou vypisována jako řetězce znaků rozdělené čárkami (pomocí funkce `synString()`).

Grafické zobrazení realizuje knihovna `Vis.js`, které jsou vstupem data vytvořená ze serverové odpovědi (struktura a tvorba těchto dat je popsána v kapitole *Zpracování odpovědi* na straně 73). Nastavení vykreslovací funkce je upraveno tak, aby výsledný graf byl statický a s jeho uzly nebylo možné tažením myši hýbat. Důvody k tomuto nastavení jsou rozebrány v kapitole *Návrh rozhraní* na straně 55. Mezi dalšími nastaveními jsou například parametry zajišťující penalizaci překryvu uzlů (nutno podotknout, že jakkoliv vysoká tato penalizace může být, nepodařilo se najít taková nastavení, aby k ní za žádných okolností – zejména

na velkých synsetech – s jistotou nedošlo), vzhled jednotlivých druhů uzlů⁶ či parametry stabilizace grafu. Stabilizace grafu je proces, který probíhá poté, co jsou knihovnou vytvořeny na vykreslovacím plátně všechny uzly a hrany mezi nimi. Vykreslovací funkce využívá emulace fyzikálních zákonů mezi jednotlivými prvky, takže uzly se odpuzují podobně, jako v reálném světě se odpuzují elektrony, a hrany se chovají jako pružiny, které mají svou ideální délku. Kombinace odpudivých a přitažlivých sil zajišťuje tendenci k rovnoměrnému rozmístění prvků na ploše.

Uchovávání stavu v URL

Aplikace svůj stav reflektuje v URL a umožňuje tak případné pozdější obnovení do identického stavu, v jakém se v libovolném okamžiku právě nachází. Aby bylo toto dosažitelné, je nutné uchovávat zvolený zdroj dat, obsah hledacího pole, zvolenou část výsledků pro zobrazení (levý sloupec) a aktivní režim zobrazení dat.

Vzhledem k tomu, že se ve všech případech jedná o relativně krátké textové informace, lze je pohodlně ukládat jako parametry URL⁷. Ty jsou nepovinnou částí adresy a mohou obsahovat data ve formátu klíč:hodnota, například `http://www.example.org/index.html?klic=hodnota`. [3] Ačkoliv Berners-Lee [3] nespecifikuje použití více dvojic klíč:hodnota v parametrech URL, běžně se toto implementuje oddělováním jednotlivých dvojic znakem ampersand (&). Tak je řešena problematika ukládání více údajů i rozhraní *NSFW viewer* :

```
...wordnet-ui.html?input=kolo&source=wncz&vis=graph#ENG20-03403613-n
```

Další částí adresy obsahující informace o stavu aplikace je fragment, tedy řetězec znaků za znakem křížku (#). Fragment je podobně jako parametry nepovinná část adresy, která se využívá na bližší identifikaci informace v dokumentu, tedy může být kupříkladu odkazem na podsekcí v dokumentu. [3] Všeobecně se fragment používá k odkazování uvnitř dokumentu lokálně na straně uživatele, naproti tomu parametry bývají odesílány na server pro účely parametrizace požadavků. V rozhraní *NSFW viewer* je fragment využíván k ukládání identifikátoru zvoleného synsetu, jeho změna tedy nevyvolává proces hledání a komunikace se serverem.

Při načtení v prohlížeči aplikace nejprve zjišťuje, zda adresa obsahuje parametry, ze kterých by se měl obnovit stav. Pokud ne (tedy neobsahuje žádné parametry, či se existující parametry neshodují s těmi používanými aplikací), je rozhraní uvedeno do výchozího stavu. V opačném případě proběhne pokus o uvedení rozhraní do stavu odpovídající-

⁶vzhled prvků grafu není možné definovat externími pravidly v CSS

⁷běžně označované jako query strings

cího hodnotám parametrů, tedy spustí se vyhledávání podle parametru `input` ve wordnetu zvoleném podle parametru `source`. Pokud hledání proběhne úspěšně, zobrazí se v režimu vizualizace podle parametru `vis synset`, jehož identifikátor odpovídá fragmentu.

Kapitola 12

Možnosti dalšího vývoje

Ačkoliv je v rámci této práce vytvořená implementace funkčním rozhraním k datům wordnetů, skýtá zároveň široké možnosti ohledně vylepšení, kterými by mohla být obohacena. Tato kapitola stručně probere ty, které byly identifikovány jako nejzajímavější pro zvýšení reálné použitelnosti rozhraní *NSFW viewer*.

Jelikož už nyní aplikace umožňuje pracovat s více wordnety, je evidentní, že by bylo vhodné dát uživateli možnost přepnout jazyk rozhraní do jeho vlastního jazyka. Množství textů, které by bylo potřeba lokalizovat, sice není nijak enormní, navíc dosavadní jazyk rozhraní je angličtina jakožto *lingua franca*, přesto by však lokalizace i těch několika řetězců, jež se v rozhraní vyskytují, mohla mít potenciál zvýšit komfort uživatele při používání rozhraní.

S tímto souvisí i další krok, který by jistě vedl ke zvýšení komfortu laického koncového uživatele. V rozhraní se momentálně vyskytují názvy sémantických relací přímo ve formě ze serverové odpovědi, která vychází z lingvistických názvů příslušných vztahů. Bylo by tedy vhodné zajistit jejich překlad do přirozeného jazyka tak, aby i uživatel bez lingvistického vzdělání byl schopen identifikovat, o jakou relaci jde. Lze například předpokládat, že nežli pojem *hyponymum* bude takovému uživateli srozumitelnější označení *Významy podřadné*, etc.

Samotné vyhledávání by bylo možné usnadnit uživateli tak, že by mu byly na základě částečného vstupu nabízeny výrazy, jež se ve zvoleném wordnetu vyskytují. Takovou nápovědu používá například rozhraní *WordVis*. [59]

Pokud by rozhraní mělo být využito jako výukového materiálu, nebylo by od věci uvažovat o jeho propojení například s Internetovou jazykovou příručkou¹, z níž by bylo možné čerpat definice či slovní formy.

¹<http://prirucka.ujc.cas.cz>

Dalším směrem, kterým by se budoucí vývoj mohl ubírat, je rozšíření podpory více jazyků. V současné implementaci sice již existuje možnost přepínat jazyky hledání (pomocí přepínání zdrojů dat) a je i možné díky identifikátorům synsetů jednotným přes více jazyků vyhledávat jeden koncept v různých jazycích, bylo by však vhodné umožnit jednodušší a rychlejší cestu, jak procházet významy slov v mezijazykovém kontextu.

V neposlední řadě jsou možnosti rozvoje i v detailech týkajících se uživatelské zkušenosti s rozhraním. V tomto ohledu se naskýtají možnosti rozšíření schopností aplikace obsloužit uživatele různými způsoby znevýhodněnými, ať už jde o jejich fyzické znevýhodnění (například poškození zraku) či znevýhodnění v oblasti programového vybavení jejich zařízení (tedy omezená podpora technologií vyžadovaných pro správný běh aplikace, zejména JavaScriptu). Nutno podotknout, že co se prvního týče, byla během návrhu grafické stránky rozhraní snaha o co nejvyšší přístupnost, takže byl zvolen vzhled s vysokým kontrastem kontur a nezaváděly se ani žádné prvky závislé na pohybu kursoru myši po obrazovce.

Závěr

Cílem praktické části této práce bylo vytvořit funkční uživatelské rozhraní k sémantickému síťovému typu wordnet. Vlastnosti, které by mělo takové rozhraní mít, byly určeny na základě podrobného empirického rozboru některých existujících rozhraní a rozebrány v kapitole *Návrh rozhraní*. Byly určeny klíčové nedostatky testovaných rozhraní vzhledem k rozvoji technologií a trendům v oblasti způsobu využívání rozhraní zpřístupňujících data. Jako nejrozšířenější nedostatek existujících rozhraní byla identifikována jejich nedostatečná univerzalita, tedy schopnost obsloužit uživatele bez ohledu na to, zda právě pro přístup k danému rozhraní chtějí použít mobilní zařízení s relativně malou obrazovkou, tablet či klasický počítač a bez ohledu na to, který operační systém na svém zařízení používají. Přitom nezávislost na zařízení a platformě je z dnešního hlediska klíčovou vlastností, ježto poměr uživatelů, kteří pracují na mobilním zařízení a těch, kteří pracují na klasickém počítači, se stále posouvá ve prospěch těch prvních.

Na základě zhodnocení potřebných parametrů rozhraní, které by nebylo zatíženo vystavšími nedostatky testovaných rozhraní, bylo definováno, že nové rozhraní by mělo být tzv. webové, tedy zobrazitelné ve standardních prohlížečích určených k procházení sítě WWW. To jednak eliminuje potřebu uživatele přesvědčit k instalaci přídavného programového vybavení na jeho počítač, a jednak je tím zajištěna nezávislost na platformě (operačním systému), kterou uživatel používá.

Další podstatnou vlastností související s univerzalitou, které by nové rozhraní mělo mít, aby přineslo oproti existujícím rozhraním přidanou hodnotu, je schopnost pracovat s více zdroji informací (wordnety) a mít otevřený kód. Schopnost práce s více zdroji informací implikuje univerzální formát vstupních dat, což je v kombinaci s otevřeným kódem důležité proto, aby bylo možné rozhraní nasadit nezávisle na již existujících instalacích, modifikovat podle konkrétních potřeb a ideálně též vylepšovat. Všechny komponenty rozhraní, kterých je pro jeho funkci využíváno (jQuery, vis.js, etc.) používají licenci MIT, což umožňuje rozhraní vytvořené v rámci této práce bez omezení redistribuovat a modifikovat.

Po analýze všech těchto požadavků bylo vytvořeno rozhraní určené pro webové prohlížeče, které na zařízení uživatele neklade žádné nestandardní požadavky ani po stránce technického, ani programového vybavení. Aby byla splněna podmínka vyžadující možnost užívat uživatelské rozhraní aplikace na zařízení (téměř) jakékoliv velikosti, bylo toto vyvinuto za použití moderních technologií a koncipováno jako tzv. responsivní, tedy schopné se dynamicky přizpůsobit velikosti zobrazovací plochy beze ztráty čitelnosti obsahu a jednoduchosti navigace v něm. Vzhled rozhraní je minimalistický, aby se uživatel mohl zaměřit na obsah. Zároveň se však snaží být elegantní a nadčasový.

Jako každý projekt, který se zabývá programovým vývojem, ani tento není bez možností dalšího pokračování. Možné směry v tomto ohledu naznačené v kapitole *Možnosti dalšího vývoje* bezpochyby nejsou jediné, kterými by se rozvoj aplikace mohl ubírat. Vzhledem k tomu, že již nyní aplikace umožňuje práci s různojazyčnými sémantickými sítěmi, se zdá nejbližším dalším vhodným cílem být implementace lokalizace rozhraní do dalších jazyků a překlad lingvistických termínů do pojmů laickými uživateli srozumitelných.

Jak bylo naznačeno v teoretické části této práce, wordnety navzdory svému původnímu určení mohou teoreticky sloužit lidským uživatelům. K tomu je ovšem potřebné rozhraní, které bude použitelné nejen pro lingvisticky vzdělané uživatele a bude odpovídat současným trendům technologickým i vizuálním. Praktická část této práce dokázala navrhnout a realizovat takové rozhraní, jež je vhodným základem pro další kroky směrem ke zpřístupnění wordnetů lidským uživatelům, zejména z řad studentů.

Bibliografie

- [1] Almende B.V. *vis.js*. URL: <http://visjs.org/> (cit. 25. 05. 2017).
- [2] Christian Amado. *Can Windows Phone Support Java?* Stack Overflow. 2013. URL: <https://stackoverflow.com/a/15501535> (cit. 30. 05. 2017).
- [3] T. Berners-Lee et al. *Request for Comments: 3986. Uniform Resource Identifier (URI): Generic Syntax*. 2005. (Cit. 28. 05. 2017).
- [4] Michael Bostock et al. „D3 Data-Driven Documents”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (pros. 2011), s. 2301–2309. ISSN: 1077-2626. DOI: 10.1109/TVCG.2011.185. URL: <http://dx.doi.org/10.1109/TVCG.2011.185> (cit. 21. 04. 2017).
- [5] Allan M Collins a M Ross Quillian. „Retrieval time from semantic memory”. In: *Journal of verbal learning and verbal behavior* 8.2 (1969), s. 240–247.
- [6] Cornetto. *Getting started with the Cornetto demo*. URL: <http://cornetto.clarin.inl.nl/help/GettingStarted.pdf>.
- [7] D Alan Cruse. *Lexical semantics*. Cambridge University Press, 1986.
- [8] Ecma International. *Standard EMCA-404. The JSON Data Interchange Format*. 2013. URL: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> (cit. 31. 05. 2017).
- [9] Laura Elizabeth. *Should You Develop a Desktop or Web App?* 2015. URL: <https://www.sitepoint.com/web-desktop-apps/> (cit. 29. 04. 2017).
- [10] David Epstein. „ICS 161: Design and Analysis of Algorithms. Lecture notes for Feb. 15, 1996”. In: (1996), s. 1–5. URL: <https://www.ics.uci.edu/~eppstein/161/960215.html> (cit. 28. 05. 2017).
- [11] Christiane Fellbaum. „WordNet and wordnets”. In: *Encyclopedia of Language and Linguistics* (2005). Ed. Keith et al. Brown, s. 665–670.

- [12] Samuel Fillenbaum a Lyle V Jones. „Grammatical contingencies in word association”. In: *Journal of Verbal Learning and Verbal Behavior* 4.3 (1965), s. 248–255.
- [13] Charles J Fillmore et al. „FrameNet as a”Net”.” In: *LREC*. 2004.
- [14] Darja Fišer a Jernej Novak. „Visualizing sloWNet”. In: *Proceedings of the Electronic Lexicography in the 21st Century (eLex 2011)* (2011). Ed. Iztok Kosem a Karmen Kosem, s. 76–82. (Cit. 25. 04. 2017).
- [15] Darja Fišer et al. „sloWNet 3.0: development, extension and cleaning”. In: *GWC 2012 6th International Global Wordnet Conference*. 2012, s. 113. (Cit. 23. 04. 2017).
- [16] Jesse J. Garrett et al. *Ajax: A new approach to web applications*. 2005. URL: https://courses.cs.washington.edu/courses/cse490h/07sp/readings/ajax_adaptive_path.pdf (cit. 26. 05. 2017).
- [17] Merrill F Garrett. „Production of speech: Observations from normal and pathological language use”. In: *Normality and pathology in cognitive functions* (1982), s. 19–76.
- [18] Lars Marius Garshol. „Metadata? Thesauri? Taxonomies? Topic Maps!” In: *Ontopia* (). URL: <http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html#N773>.
- [19] Mateo Gianolio. *An interactive visualization of the Princeton WordNet database*. URL: <http://mateogianolio.com/wordnet-visualization/> (cit. 01. 05. 2017).
- [20] Gizmo. *Can we run Java applications on iPhone?* Ed. Shai Almog. Stack Overflow. 2012. URL: <https://stackoverflow.com/a/1193541> (cit. 30. 05. 2017).
- [21] Global WordNet Association. *GWA Base Concepts*. URL: <http://globalwordnet.org/gwa-base-concepts/> (cit. 06. 04. 2017).
- [22] Global WordNet Association. *Wordnets in the World*. URL: <http://globalwordnet.org/wordnets-in-the-world/> (cit. 25. 03. 2017).
- [23] James Gosling a Henry McGilton. „The Java language environment”. In: *Sun Microsystems Computer Company* 2550 (1995). (Cit. 29. 04. 2017).
- [24] Tor-Morten Gronli et al. „Mobile application platform heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS”. In: *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*. IEEE. 2014, s. 635–641.
- [25] Bohuslav Havránek et al. *Slovník spisovného jazyka českého*. cit. dle Internetové jazykové příručky, 2. nezměn. vyd. Praha: Academia, 1989.

- [26] Yoni Heisler. *Mobile internet usage surpasses desktop usage for the first time in history*. 2016. URL: <http://bgr.com/2016/11/02/internet-usage-desktop-vs-mobile/> (cit. 29. 04. 2017).
- [27] Himanshu. *Can you run 'normal' Java programs on Android?* Android Enthusiasts Stack Exchange. 2014. URL: <https://android.stackexchange.com/a/92854> (cit. 30. 05. 2017).
- [28] Aleš Horák et al. „The development of a complex-structured lexicon based on WordNet”. In: (cit. 17. 04. 2017).
- [29] Jun Hyunje. *Perfect Scrollbar*. 2017. URL: <https://github.com/noraesae/perfect-scrollbar> (cit. 25. 05. 2017).
- [30] Konstantin Isakov. *GoldenDict - About*. 2013. URL: <http://goldendict.org/> (cit. 29. 04. 2017).
- [31] jQuery. *What is jQuery?* URL: <https://jquery.com/> (cit. 25. 05. 2017).
- [32] Jaap Kamps a Maarten Marx. „Visualizing wordnet structure”. In: *Proc. of the 1st International Conference on Global WordNet*. 2002, s. 182–186.
- [33] Adrienne Lehrer a Keith Lehrer. „Antonymy”. In: *Linguistics and philosophy* 5.4 (1982), s. 483–501.
- [34] Lothar Lemnitzer a Claudia Kunze. „Using WordNets in Teaching Virtual Courses of Computational Linguistics”. In: *Second International WordNet Conference, GWC 2004*. Ed. P. Sojka et al. 2004, s. 374. (Cit. 23. 04. 2017).
- [35] Edvard Lotko. *Slovník lingvistických termínů pro filology*. 3. nezm. vyd. Olomouc: Univerzita Palackého v Olomouci, 2003. ISBN: 80-244-0720-5.
- [36] Willy Martin a I Maks. „Referentie Bestand Nederlands”. In: *Met medewerking van S. Bopp en M. Groot* (2005).
- [37] Karen McGrane. „The rise of the mobile-only user”. In: *Harvard Business Review* (2013). URL: <https://hbr.org/2013/05/the-rise-of-the-mobile-only-us> (cit. 23. 07. 2017).
- [38] George A Miller et al. „Introduction to WordNet: An on-line lexical database”. In: *International journal of lexicography* 3.4 (1990), s. 235–244.
- [39] MOZILLA DEVELOPER NETWORK. *XMLHttpRequest*. 2017. URL: <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest> (cit. 01. 06. 2017).

- [40] Mozilla Foundation. *Why do Java, Silverlight, Adobe Acrobat and other plugins no longer work?* 2017. URL: <https://support.mozilla.org/en-US/kb/npapi-plugins> (cit. 29. 04. 2017).
- [41] Roberto Navigli a Simone Paolo Ponzetto. „BabelNet: Building a Very Large Multilingual Semantic Network”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden: Association for Computational Linguistics, 2010, s. 216–225. URL: <http://dl.acm.org/citation.cfm?id=1858681.1858704> (cit. 27. 04. 2017).
- [42] Roberto Navigli a Simone Paolo Ponzetto. „BabelNetXplorer: A Platform for Multilingual Lexical Knowledge Base Access and Exploration”. In: *Proceedings of the 21st International Conference on World Wide Web*. WWW '12 Companion. Lyon, France: ACM, 2012, s. 393–396. ISBN: 978-1-4503-1230-1. DOI: 10.1145/2187980.2188057. URL: <http://doi.acm.org/10.1145/2187980.2188057> (cit. 27. 04. 2017).
- [43] Jakob Nielsen. *Scrolling and Scrollbars*. 2005. URL: <https://www.nngroup.com/articles/scrolling-and-scrollbars/> (cit. 22. 04. 2017).
- [44] Dorling Kindersley Oxford University Press. *Illustrated Oxford dictionary*. Rev. ed. London, Oxford: Dorling Kindersley, 2003. ISBN: 07-513-6436-3.
- [45] Karel Pala a Pavel Smrž. „Building czech wordnet”. In: *Romanian Journal of Information Science and Technology* 7.2-3 (2004), s. 79–88.
- [46] Karel Pala a Pavel Ševeček. „Česká lexikální databáze typu WordNet”. In: *Feedback* 48.A47 (2013).
- [47] Carita Paradis a Caroline Willners. „Antonymy and negation—The boundedness hypothesis”. In: *Journal of pragmatics* 38.7 (2006), s. 1051–1080.
- [48] Maria Teresa Pazienza et al. „A Bottom-up Comparative Study of EuroWordNet and WordNet 3.0 Lexical and Semantic Relations.” In: *LREC*. 2008.
- [49] Princeton University. *About WordNet*. 2010. URL: <http://wordnet.princeton.edu> (cit. 30. 07. 2017).
- [50] Sundaram Ramaswamy. *Artha The Open Thesaurus*. 2012. URL: <http://artha.sourceforge.net/wiki/index.php/Home> (cit. 29. 04. 2017).
- [51] Adam Rambousek. *osobní sdělení*. Brno, 22. 4. 2017.
- [52] Rodney Rehm. *URI.js: Intro*. URL: <https://medialize.github.io/URI.js/> (cit. 25. 05. 2017).

- [53] Jiří Rejzek. *Český etymologický slovník*. 2., nezměň. vyd. Voznice: Leda, 2012. ISBN: 978-80-7335-296-7.
- [54] John T. Richards a Vicki L. Hanson. „Web accessibility: a broader view”. In: *Proceedings of the 13th conference on World Wide Web - WWW '04*. ACM Press, 2004. DOI: 10.1145/988672.988683. URL: <https://doi.org/10.1145/988672.988683> (cit. 21. 04. 2017).
- [55] Tom Richens. „Anomalies in the WordNet verb hierarchy”. In: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics. 2008, s. 729–736. (Cit. 01. 05. 2017).
- [56] J Szymanski et al. „Cooperative editing approach for building wordnet database”. In: *Proceedings of the XVI international conference on system science*. 2007, s. 448–457.
- [57] David S Touretzky. *The mathematics of inheritance systems*. Sv. 8. Morgan Kaufmann, 1986.
- [58] Alexandre N Tuch et al. „The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments”. In: *International Journal of Human-Computer Studies* 70.11 (2012), s. 794–811. (Cit. 30. 04. 2017).
- [59] Steven Vercruysse. *About WordVis*. 2010. URL: <http://wordvis.com/about.html> (cit. 23. 04. 2017).
- [60] Piek Vossen. „Introduction to eurowordnet”. In: *Computers and the Humanities* 32.2-3 (1998), s. 73–89.
- [61] Piek et al. Vossen. „EuroWordNet: a multilingual database for information retrieval”. In: *Proceedings of the DELOS workshop on Cross-language Information Retrieval*. 1997, s. 5–7.
- [62] John W. Hutchins. „The evolution of machine translation systems”. In: *Lawson* (1982), s. 21–37. URL: <http://mt-archive.info/Aslib-1981-Hutchins-1.pdf> (cit. 26. 07. 2017).
- [63] W3Schools. *HTML Canvas Reference*. URL: https://www.w3schools.com/tags/ref_canvas.asp (cit. 23. 04. 2017).
- [64] W3Schools. *HTML5 Browser Support*. URL: https://www.w3schools.com/html/html5_browsers.asp (cit. 30. 05. 2017).
- [65] W3Schools. *JSON vs XML*. URL: https://www.w3schools.com/js/js_json_xml.asp (cit. 01. 06. 2017).

- [66] Tommy Walker. *Why “Simple” Websites Are Scientifically Better*. 2013. URL: <https://conversionxl.com/why-simple-websites-are-scientifically-better/> (cit. 30.04.2017).
- [67] WordNet Reference Manual. *wngloss – glossary of terms used in WordNet system*. URL: <https://wordnet.princeton.edu/wordnet/man/wngloss.7WN.html> (cit. 01.05.2017).