UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA

PRÍRODOVEDECKÁ FAKULTA

# Stromová reprezentácia slovenskej vety

*Autor: Jana Hiľovská*
*Vedúci práce: RNDr. Ondrej Krídlo, PhD.*
*Konzultant: doc. RNDr. Stanislav Krajči, PhD.*

10. januára 2017

# Obsah

# 1 Úvod

Naša diplomová práca sa bude zaoberať problematikou spracovávania textu písaného v prirodzenom jazyku. Konkrétne sa budeme zaoberať automatizáciou vetného rozboru. Vetný rozbor je disciplínou syntaxe – náuky o tvorbe viet, ktorá je jednou z častí gramatiky. Táto disciplína sa zaoberá určovaním, akými vetnými členmi sú slová vo vete a aké sú vzťahy medzi týmito slovami. Slovenský jazyk má niekoľko vetných členov (napríklad podmet, prísudok, predmet a ďalšie) medzi ktorými sú priraďovacie, či podraďovacie vzťahy. Našou úlohou v tejto diplomovej práci bude navrhnúť a implementovať algoritmus, ktorý tieto vzťahy medzi slovami určí automaticky. Keďže vzťahy medzi slovami majú hierarchickú štruktúru, výsledkom algoritmu by mala byť stromová štruktúra slovenskej vety.

Asi je prirodzené položiť si otázku, načo je vlastne výskum v takejto oblasti potrebný. Aj keď to možno na prvý pohľad nie je vidieť, automatická analýza textu má využitie v čoraz populárnejšom odbore vedy a techniky – v umelej inteligencii. Ak totiž naučíme počítač (robota) rozoznávať nielen jednotlivé slovné druhy vo vete, ale aj vzťahy medzi týmito slovami, sme o krok bližšie k tomu, aby sme ho naučili rozumieť tomu, čo hovoríme. Ak totiž zvládneme automatizovať syntaktickú rovinu jazyka, ďalším krokom je naučiť počítač sématiku slov a prototyp umelej inteligencie je hotový. Hoci súčasné programy typu Siri, či Cortana od Microsoftu sa umelú inteligenciu snažia simulovať, v skutočnosti významom slov nerozumejú.

Ďalším dôvodom, prečo je potrebné sa venovať počítačovému spracovaniu slovenčiny je akási nesúrodosť v tejto oblasti ako takej. Pravdou je, že existuje niekoľko výskumných tímov, ktoré o vzťahu počítač – prirodzený jazyk, resp. prirodzený jazyk – počítač publikujú, no výskum ako taký nezastrešuje žiaden vedný odbor. Veríme, že počítačová lingvistika na Slovensku má dostatočný potenciál a dokáže poskytnúť priestor na spojenie vedomostí z lingvistiky a informatiky.

Dôkazom toho sú nielen konferencie na túto tému – z najznámejších spomeniem SLOVKO, ale aj úspešný výskum v Českej republike. Veľkým krokom vpred pre počítačové spracovanie českého jazyka bolo spustenie projektu *Pražský závislostní korpus (Prague dependency tree – PDT)*. Tento korpus obsahuje morfologickú, syntaktickú aj sématickú anotáciu slovných jednotiek (aktuálne sa ich počet pohybuje v rozmedzí 0.8 – 2 milióny). To, že na slovenskej akademickej pôde takto rozsiahly korpus nemáme, nám dáva možnosť vyskúšať nové prístupy.

# 2 Charakteristika problému

## 2.1 Roviny spracovania jazyka

V oblasti spracovania prirodzeného jazyka sa hovorí o tzv. rovinách popisu (a spracovania) jazyka. Tieto roviny sú usporiadané odhora dole, od roviny najjednoduchšej, ktorá sa zaoberá ortografiou, či akustickou stránkou veci, po rovinu najzložitejšiu, teda rovinu významu. Každá rovina má svoje jednotky popisu, definície vzťahov na tejto rovine, a naväzuje bezprostredne na rovinu nižšiu a vyššiu. Obvykle sa hovorí o piatich až šiestich rovinách

- akustika/ortografia
- fonetika
- fonológia
- morfológia
- syntax
- sémantika

Často sa však niektoré z týchto rovín zlučujú.
V tejto časti sa obmedzíme na spracovanie textu. Rozpoznávanie (a syntéza) hovorenej reči je síce v zmysle „porozumenia“ jazyku podobný problém, avšak tradične sa sústredí hlavne na spracovanie akustického signálu, a v istom zmysle – aspoň z dnešného pohľadu, s existujúcimi aplikáciami a systémy v ruke – sa na ňu vieme pozerať ako na spracovanie hovorenej reči na text, ktorý potom ďalej spracovávame

Syntax a sémantika takisto spolu úzko súvisia a nie náhodou sa analýza na tejto úrovni nazýva syntakticko-sémantická, pričom sa opäť zlučujú dokopy dve roviny. Naopak, niekedy je výhodné vložiť medzi morfológiu a syntax ešte jednu rovinu, a to rovinu tzv. povrchovej syntaxe. Práve táto rovina, spolu s morfologickou sa zohľadňuje v zahraničných článkoch, ktoré nespracovávajú flektívne jazyky.

Štruktúra vety prirodzeného jazyka je definovaná gramatickými pravidlami. Množina všetkých gramatických pravidiel jazyka tvorí gramatiku jazyka. Gramatika jazyka nám umožňuje generovať vety jazyka, ktoré majú základnú vlastnosť: sú gramaticky správne. Gramatickými pravidlami - gramatikou - určujeme syntax jazyka, to znamená, že určujeme prípustnú štruktúru viet jazyka a elementárne jednotky, ktoré možno na danom mieste použiť. Význam viet je určený sémantikou. Syntax a sémantika navzájom úzko súvisia. Syntax určuje štruktúru vety, ktorá je základom pre určenie sémantiky (významu) vety. Syntaktická správnosť vety nezaručuje jej sémantickú správnosť. Vetný člen je „stavebná jednotka“ vety. Je to časť vety, ktorá je k inej časti vety v istom vzťahu. Vetné členy sú navzájom spojené skladmi. Podmet a prísudok sú základné vetné členy dvojčlennej vety. Prísudok vyjadruje dynamický príznak, ktorý sa prisudzuje podmetu.

## 2.2 Vetné členy

Ako sme už spomenuli v predchádzajúcej kapitole, slovenskú vetu tvorí viacero vetných členov. Môžeme ich rozdeliť na hlavné vetné členy a rozvíjacie vetné členy. Medzi týmito entitami sú vo vete vzťahy, ktoré vieme reprezentovať ako strom. Hlavné vetné členy v slovenskej vete:

Podmet

- podľa jazykovedcov vykonávateľ činnosti, alebo nositeľ stavu,
- býva vyjadrený podstatným menom alebo zámenom v nominatíve, prípadne iným plnovýznamovým slovným druhom.

Prísudok

- vyjadruje činnosť, stav alebo vlastnosť podmetu,
- slovesné prísudky sú vyjadrený slovesom,
- menné prísudky sú vyjadrený spojením sponového slovesa a plnovýznamového slovného druhu. Rozvíjacie vetné členy

Predmet

- zvyčajne vyjadrený podstatným menom alebo zámenom,
- viaže sa s pádmi rôznymi od nominatívu a rozvíja prísudok.

Príslovkové určenie

- rozvíja sloveso, prídavné meno alebo príslovku.
- otázky: kde?, kedy?, ako?, prečo?.
- tvorené príslovkou alebo podstatným menom

Prívlastok

- rozvíja podstatné meno,
- zhodné prívlastky : majú rovnaké gramatické kategórie ako slová, ktoré rozvíjajú. Môžu byť vyjadrené prídavným menom, príslovkou,
- nezhodné prívlastky majú iné gramatické kategórie ako ich nadradený vetný člen. Môžu byť vyjadrené napríklad podstatným menom.

Hlavným problémom pri automatickom spracovávaní viet zo skupiny flektívnych jazykov, do ktorých patrí aj slovenčina je viactvarovosť slov. Na rozdiel od napríklad angličtiny, kde plnovýznamové slová sa nemenia, v slovenčine môže mať slovo s jedným významom niekoľko tvarov, napríklad keď ho vyskloňujeme. Pre implementáciu nášho riešenia to teda bude znamenať pracovať s pomerne veľkým

množstvom dát, keďže k jednému slovu (napríklad pes) musíme brať do úvahy aj všetky jeho tvary psa, psovi atď). Špeciálnymi prípadmi, ktoré automatický proces spracovania prirodzeného jazyka sťažujú *homonymá*, teda slová s rovnakým tvarom, ale rozličným významom (t.j.: dve slová s rovnakým tvarom nemusia byť nutne rovnakým vetným členom a ani sa nemusia viazať na rovnaké slovo). Tento jav je v slovenčine pomerne častý, aj keď si ho mnohokrát pri prirodzenom prejave neuvedomujeme. Komplikácie spôsobené týmto lingvistickým javom bližšie ilustrujeme na príklade v nasledujúcej kapitole.

# 3 Analýza a návrh riešenia

## 3.1 Tvaroslovník

Na to, aby sme dokázali vetný rozbor zautomatizovať určite potrebujeme vedieť o jednotlivých slovách niekoľko dodatočných informácií. Pri našej diplomovej práci budeme používať databázu tvarov slov slovenského jazyka Tvaroslovník, ktorá bola vytvorená na Prírodovedeckej fakulte UPJŠ. V tejto databáze sa nachádza množstvo slovenských slov, ich tvarov a metadát, ktoré budeme používať pri praktickej implementácii nášho riešenia. Tvaroslovník obsahuje približne 320 000 základných tvarov slov. Spolu s vyskloňovanými tvarmi slov je v databáze približne 30 000 000 riadkov. Okrem tvaru slova každý záznam obsahuje aj informáciu o tom, akého je slovného druhu a príslušné gramatické kategórie daného slova.

Údaje v Tvaroslovníku boli získané zo Slovníka slovenského jazyka a Veľkého slovníka cudzích slov. Všetky údaje sú v databáze uložené v jednej tabuľke. Každý jej riadok obsahuje jeden z tvarov slova spolu so všetkými informáciami o ňom. Zoznam stĺpcov v tabuľke je nasledujúci:

- idSlovo – jedinečný celočíselný identifikátor slova,
- idTvar – jedinečný celočíselný identifikátor tvaru slova, kde slovo v základnom tvare má túto hodnotu nastavenú na 0,
- tvar – textový tvar slova,
- slovnýDruh – hovorí o slovnom druhu príslušného slova,
- charakteristika – textový zoznam hodnôt gramatických kategórií slova. Tieto hodnoty sú oddelené bodkočiarkami a závisia od konkrétneho slovného druhu.

### 3.1.1 Príklad záznamu v databáze Tvaroslovník:

Ak hľadané slovo je „škole“, záznam vyzerá nasledovne:

- škole [škola]
  *podstatné meno*
  *ženský r.; singulár; datív*;

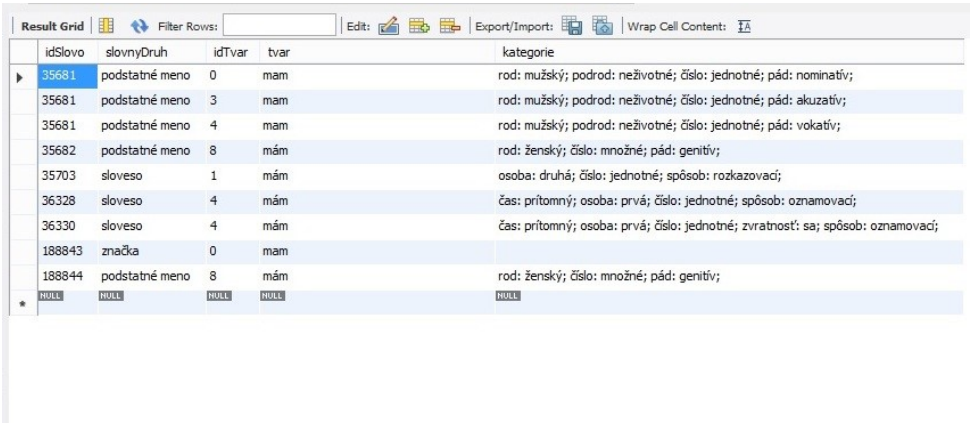
- škole [škola]
  *podstatné meno*
  *ženský r.; singulár; lokál*;

Preložené do problematiky spracovania prirodzeného jazyka to znamená, že v Tvaroslovníku majú rovnaký záznam v stĺpci *tvar*, no rôzny záznam v stĺpci *slovnyDruh*, čo spôsobuje nejasnosť v syntaktickom rozklade slovenskej vety.
Typickým príkladom je slovo **mám**, pre ktoré nájdeme v Tvaroslovníku nasledujúce záznamy:

- mám [mama]
  *podstatné meno*
  *ženský r.; plurál; genitív*;

- mám [mámiť]
  *sloveso*
  *osoba: druhá; singulár; spôsob: rozkazovací*;

- mám [mať]
  *sloveso*
  *čas: prítomný; osoba: prvá; singulár; spôsob: oznamovací*;


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| idSlovo | slovnyDruh | idTvar | tvar | kategorie |
|---|---|---|---|---|
| 35681 | podstatné meno | 0 | mam | rod: mužský; podrod: neživotné; číslo: jednotné; pád: nominatív; |
| 35681 | podstatné meno | 3 | mam | rod: mužský; podrod: neživotné; číslo: jednotné; pád: akuzatív; |
| 35681 | podstatné meno | 4 | mam | rod: mužský; podrod: neživotné; číslo: jednotné; pád: vokatív; |
| 35682 | podstatné meno | 8 | mám | rod: ženský; číslo: množné; pád: genitív; |
| 35703 | sloveso | 1 | mám | osoba: druhá; číslo: jednotné; spôsob: rozkazovací; |
| 36328 | sloveso | 4 | mám | čas: prítomný; osoba: prvá; číslo: jednotné; spôsob: oznamovací; |
| 36330 | sloveso | 4 | mám | čas: prítomný; osoba: prvá; číslo: jednotné; zvratnosť: sa; spôsob: oznamovací; |
| 188843 | značka | 0 | mam | |
| 188844 | podstatné meno | 8 | mám | rod: ženský; číslo: množné; pád: genitív; |
| NULL | NULL | NULL | NULL | NULL |

Obr. 1: Tvaroslovník

## 3.2 Charakteristika slovných druhov

Pri návrhu riešenia bude dôležitým ukazovateľom vzťahu medzi slovami nielen samotný slovný druh, ale špecifické gramatické kategórie, ktoré tomuto slovnému druhu prislúchajú. Nasledujúce údaje sú uvedené v databáze pre každé slovo v stĺpci charakteristika. Na rozdiel od predchádzajúcich prác, naše riešenie sa nespolieha na koreláciu medzi vetnými členmi a slovnými druhmi. Mnohé výskumy sa snažia postaviť automatický rozbor vety práve na lingvistických znalostiach. Slovné druhy vo vete teda rozdelia ta nie, ktoré majú, resp. nemajú vetnočlenskú platnosť. Prístup, ktorý sme navrhli my spočíva v úplnej stromovej reprezentácii slovenskej vety a to vrátane členov, ktoré nemajú v lingvistike vetnočlenskú platnosť.

Plnovýznamové slovné druhy

- podstatné mená majú rod, číslo, pád, vzor a informáciu o tom, či sú životné alebo neživotné,
- Prídavné mená majú rod, číslo, pád, ak ide o prídavné mená mužského rodu, aj podrod. Akostné a vzťahové prídavné mená majú aj uvedené, v akom sú stupni,
- Zámená, ktoré sú osobné, majú v charakteristike uvedené tieto kategórie: osoba, číslo, pád, rod
  - Osobné privlastňovacie zámená v základnom tvare majú uvedenú poznámku, že sú privlastňovacie od nejakého osobného zámena,
  - Opytovacie zámená majú uvedený pád a ak sa to dá z tvaru slova zistiť, aj rod a číslo, pri mužskom rode aj podrod.
  - Ukazovacie, zvratné zámená sa a si, neurčité a vymedzovacie zámená, ktoré sú nesklonné, nemajú v charakteristike uvedené žiadne vlastnosti,
  - Sklonné ukazovacie, tvary zvratných zámen seba a sebe, neurčité a vymedzovacie zámená majú uvedený pád, rod, číslo, v prípade mužského rodu aj podrod.,
- Číslovky majú uvedený pád, rod (v mužskom rode aj podrod), číslo. Skupinové číslovky majú uvedený pád a číslo. Násobné a nesklonné neurčité číslovky nemajú uvedené žiadne charakteristiky,
- Slovesá majú v charakteristike uvedené číslo, čas, spôsob. Slovesá, ktoré môžu byť aj zvratné, majú uvedenú aj zvratnosť spolu so zvratným zámenom, s ktorým sa viažu. Slovesá, ktoré sú prechodníky, trpné alebo činné príčastia, túto skutočnosť majú uvedené v položke charakteristiky forma. Neurčitky majú v stĺpci charakteristika uvedené iba forma: neurčitok, okrem tejto položky tam nie sú uvedené žiadne ďalšie položky,
- Príslovky môžu mať charakteristiku stupeň. Príslovky, ktoré nemožno stupňovať, nemajú v stĺpci charakteristika uvedenú žiadnu položku,

Neplnovýznamové slovné druhy

- Predložky majú v charakteristike uvedenú položku väzba, v ktorej sú uvedené pády, s ktorými sa viažu. Existujú slová, ktoré môžu byť v závislosti od kontextu predložkami alebo príslovkami. Takéto

slová majú v atribúte *slovnýDruh hodnotu* predložka, príslovka. V charakteristike majú uvedené , s akými pádmi sa viažu, ak sú predložkou,

- Spojky, častice a citoslovcia nemajú v atribúte charakteristika uvedené žiadne položky.

## 3.3 Návrh riešenia

Aktuálny návrh riešenia pozostáva z postupného prechádzania jednotlivých slov vo vete, medzi ktorými hľadáme vzťahy. K rozpoznaniu vzťahov nám pomáha databáza Tvaroslovník. Na rozdiel od diplomovej práce kolegu Júliusa Mareša, ktorá sa zaoberala týmto odvetvím informatiky sa naša práca primárne nesústredí na určovanie vetných členov. Naopak, zvolili sme opačný prístup, v ktorom najprv určíme vzťahy medzi slovami, z ktorých potom budú viditeľné nielen samotné vetné členy, ale aj celý syntaktický rozbor vety.

V aktuálnom návrhu riešenia predpokladáme, že slová, ktoré spolu tvoria vzťah stoja (resp. v istom kroku algoritmu budú stáť) vedľa seba. Každý spracovávaný vzťah má podradené a nadradené slovo. Výnimkou nie je ani lingvisticky rovnocenný prisudzovací vzťah, ktorý ma v stromovej reprezentácii vety ako nadradené slovo prísudok a podradené podmet.

Vetu na vstupe teda spracovávame po dvojiciach, pričom každej z týchto dvojíc pridelíme *prioritu* a na základe tejto *priority* ich usporiadame do radu. Z radu potom vyberieme prvú dvojicu a vytvoríme z nej časť stromu, prípadne ju pripojíme k už existujúcemu uzlu v strome. Zo spracovaného vzťahu vyberieme podradené slovo, ktoré vymažeme z vety aj zo spomínaného usporiadaného radu potencionálnych vzťahov vo vete. Takouto úpravou vety sa zníži počet jej slov upraví sa aj jej lineárne usporiadanie (t.j. vo vete vzniknú nové susediace slová predstavujúce potencionálne vzťahy). *Prioritu* sme určili empiricky a to manuálnym rozborom viet z rôznych zdrojov.

Pri návrhu riešenia sme museli špeciálnu pozornosť venovať slovným druhom, ktoré nemajú vetnočlenskú platnosť, no pri reprezentácii vety majú v strome osobitný uzol. Dobrým príkladom sú predložky. Tie síce v slovenčine nemajú vetnočlenskú platnosť, no pri spracovaní prirodzeného jazyka poskytujú niektoré dodatočné informácie (napr. väzba s určitým pádom) a teda tvoria dôležitú súčasť vety a dokonca v našom návrhu riešenia stoja ako nadradený slovný druh pre podstatné meno, s ktorým tvoria vzťah. Podobným spôsobom je nutné spracovávať aj citoslovcia, ktoré v mnohých vetách zastupujú podstatné mená, či dokonca slovesá.

Jedným zo zaujímavých lingvistických fenoménov, ktorý sme v našej práci využili sú dodatočné informácie, ktoré poskytuje podradený vetný člen nadradenému. Napr. vo vete : ***Prišli domov veľmi unavení.*** Na prvý pohľad vidno, že veta je dvojčlenná, neúplná, teda nemá podmet. Hoci vzťah slov ***Prišli – domov*** je pomerne jednoducho zistiteľný, s určením prisudzovacieho vzťahu to je o niečo zložitejšie. V tomto prípade vieme, že zamlčaným podmetom je zámeno v tretej osobe. Podradený vetný člen – prívlastok nám v tomto prípade vie dodať informáciu, že ide o zámeno *Oni*, ktoré ma pri prídavných menách v nominatíve na konci *i/í*. Podobným spôsobom vieme na základe predložky rozhodnúť o tom, v ktorom páde je podradené slovo s ňou spojené.

### 3.3.1 Ilustračný príklad

Návrh riešenia ilustrujeme na nasledujúcom príklade rozboru ideálnej vety. Ďalej v tejto kapitole uvedieme niekoľko ilustračných príkladov pre problémy, s ktorými sme sa stretli počas implementácie riešenia.

**Príklad 1:**

Veta: ***Jano uvidel na okne veľmi peknú ženu.***

**1. iterácia**:

Zoznam slov v Tvaroslovníku pre tvar slova *Jano*:

Jano – podstatné meno – mužský rod – životné – nominatív

Jano – podstatné meno – mužský rod – životné – vokatív

Zoznam slov v Tvaroslovníku pre tvar slova *uvidel*:

uvidel – sloveso – minulý čas – mužský rod – oznamovací spôsob

Zoznam slov v Tvaroslovníku pre tvar slova *na*:

na – predložka – akuzatív

na – predložka – lokál

Zoznam slov v Tvaroslovníku pre tvar slova *okne*:

okne – podstatné meno – stredný rod – lokál

Zoznam slov v Tvaroslovníku pre tvar slova *veľmi*:

veľmi – príslovka

Zoznam slov v Tvaroslovníku pre tvar slova *peknú*:

peknú – prídavné meno – ženský rod – jednotné číslo – akuzatív

Zoznam slov v Tvaroslovníku pre tvar slova *ženu*:

ženu – podstatné meno – ženský rod – jednotné číslo – akuzatív

Utriedený zoznam potencionálnych vzťahov podľa priority: Pozn.:Prvé slovo vo vzťahu bude predstavovať podradené slovo [veľmi, peknú], [peknú ženu], [okne, na], [na, uvidel], [Jano (pád nominatív), uvidel], [Jano (pád vokatív), uvidel], [*okne, veľmi*] Pozn.: Vzťah vyznačený kurzívou má nastavenú prioritu označujúcu nemožný vzťah, teda, že tieto dve slová spolu vzťah určite netvoria Z utriedeného zoznamu teda odstránime prvú dvojicu ˇ vzťah [*veľmi, peknú*] a podradené slovo *veľmi* odstránime aj z vety. Zároveň tento vzťah zakreslíme aj do stromovej reprezentácie vety. Ak by v zozname utriedených vzťahov v niektorom figurovalo slovo *veľmi* ako podradené, odstránime aj tento vzťah. Dostaneme teda novú vetu, na ktorej urobíme druhú iteráciu. Ďalej už budem v iterácii popisovať iba zoznam utriedených vzťahov.

**2. iterácia**

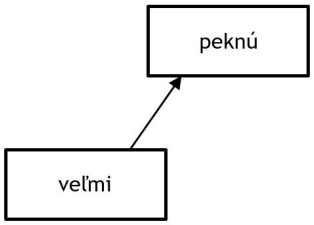
Veta: ***Jano uvidel na okne peknú ženu***

Utriedený zoznam potencionálnych vzťahov podľa priority:

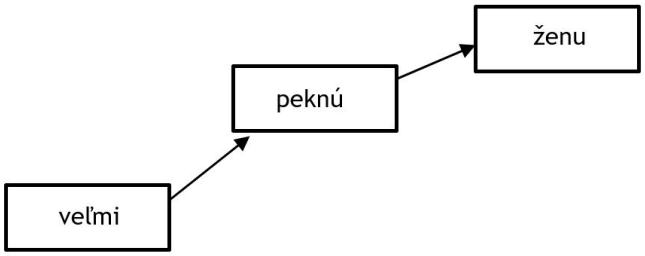
[peknú, ženu], [okne, na], [na, uvidel], [Jano (pád nominatív), uvidel],

[Jano (pád vokatív), uvidel], [*okne, peknú*].Opäť odstránime zo zoznamu prvú dvojicu –

vzťah [*peknú, ženu*] a z vety odstránime slovo *peknú*. Vzťah pridáme do stromu




Obr. 2: Strom vety po prvej iterácii




Obr. 3: Strom vety po druhej iterácii

**3. iterácia**

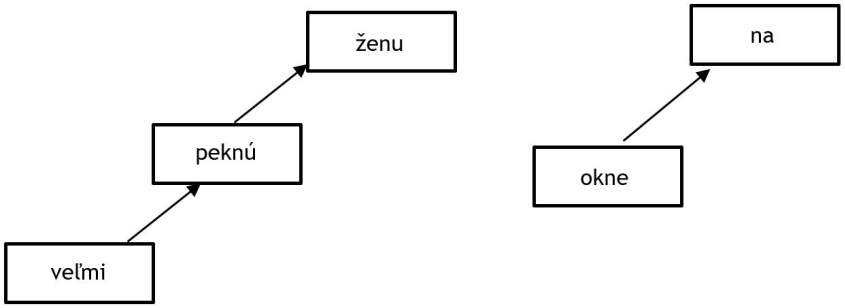
Veta: ***Jano uvidel na okne ženu***

Utriedený zoznam potencionálnych vzťahov podľa priority:

[okne, na], [na, uvidel[Jano (pád nominatív), uvidel], [Jano (pád vokatív), uvidel],

[*okne, ženu*]

Zo zoznamu odstránime prvý vzťah, z vety slovo *okne*. Nový vzťah pridáme do stromu.




Obr. 4: Strom vety po tretej iterácii

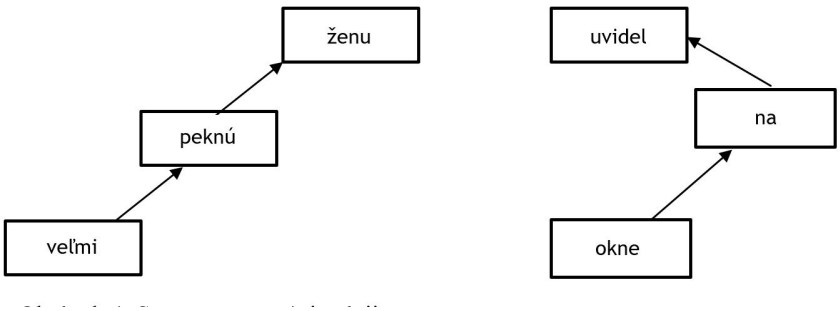
**4. iterácia**

Veta: ***Jano uvidel na ženu***

Utriedený zoznam potencionálnych vzťahov podľa priority:

[na, uvidel], [Jano (pád nominatív), uvidel], [Jano (pád vokatív), uvidel], [*na, ženu*]

Zo zoznamu odstránime prvý vzťah, z vety slovo *na*. Upravíme strom.




Obr. 5: Strom vety po štvrtej iterácii

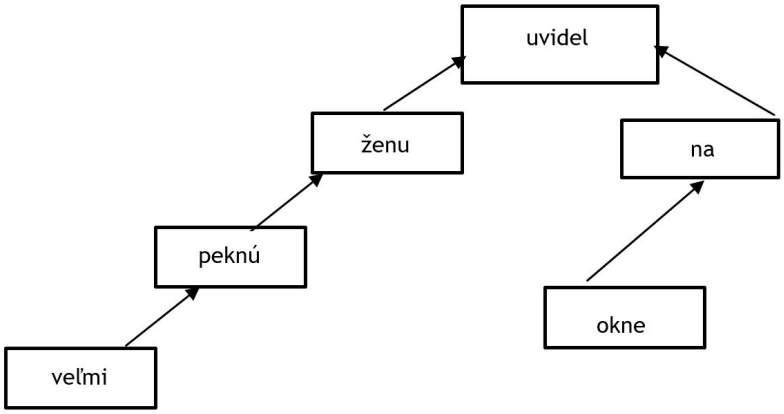
**5. iterácia**

Veta: ***Jano uvidel ženu***

Utriedený zoznam potencionálnych vzťahov podľa priority:

[ženu, uvidel], [Jano (pád nominatív), uvidel], [Jano (pád vokatív), uvidel]

Zo zoznamu odstránime vzťah [*ženu, uvidel*] a z vety slovo *ženu*. Upravíme strom.




Obr. 6: Strom vety po piatej iterácii

**6. iterácia**

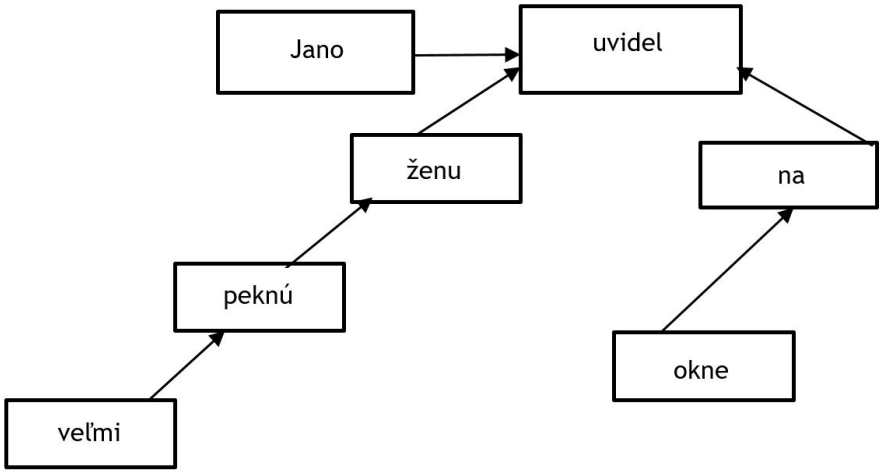
Veta: ***Jano uvidel***

Utriedený zoznam potencionálnych vzťahov podľa priority:

[Jano (pád nominatív), uvidel], [Jano (pád vokatív), uvidel]

Zo zoznamu odstránime vzťah [*Jano (pád nominatív), uvidel*], o ktorom vieme, že je prisudzovací. Teraz odstránime z vety slovo *Jano* a zo zoznamu vzťahov odstránime všetky, ktoré obsahujú slovo *Jano* (v

akomkoľvek
páde) ako podradené. Tam patrí aj vzťah [*Jano (pád vokatív), uvidel*]. V takom prípade ale už v zozname nezostanú žiadne ďalšie vzťahy na spracovanie. Keďže sme už našli aj prisudzovací vzťah, algoritmus končí a
strom vety je kompletný.




Obr. 7: Výsledný strom vety

### 3.3.2 Špecifické prípady

Slovenský jazyk obsahuje okrem homoným ešte niekoľko lingvistických javov, ktoré nami navrhnuté automatické spracovanie prirodzeného jazyka sťažujú. Problémom bývajú nielen *neúplné dvojčlenné vety* (t.j.: vety bez vyjadreného podmetu), ktorých výskyt nie je vôbec ojedinelý, ale aj spracovanie *nezhodného prívlastku* (napr. koláče od mojej babky). Samostatnou kapitolou je spracovanie viacnásobných vetných členov, či súvetí. Aktuálne sa venujeme problémom pri spracovaní dvojčlenných úplných viet, ktoré vznikajú pri výskyte slov, ktoré majú rovnaký tvar v akuzatíve aj v nominatíve. Opäť tento problém ilustrujeme na príklade

**Príklad 2:**

Aby bola neštandardná situácia lepšie viditeľná, zvolili sme podobnú vetu tej z Príkladu 1, z ktorého preberáme zápis vzťahov aj ďalšiu terminológiu.

Veta: ***Dievča uvidelo na okne veľmi pekné mača***

**1.iterácia**

Zoznam slov v Tvaroslovníku pre tvar slova uvidel:

uvidelo – sloveso – minulý čas – stredný rod ˘ oznamovací spôsob

Zoznam slov v Tvaroslovníku pre tvar slova na:

na – predložka – akuzatív

na – predložka – lokál

Zoznam slov v Tvaroslovníku pre tvar slova okne:
okne – podstatné meno – stredný rod – lokál

Zoznam slov v Tvaroslovníku pre tvar slova veľmi:
veľmi – príslovka
Zoznam slov v Tvaroslovníku pre tvar slova pekné:
pekné – prídavné meno – stredný rod – jednotné číslo – nominatív
pekné – prídavné meno – stredný rod – jednotné číslo – akuzatív
pekné – prídavné meno – stredný rod – jednotné číslo – vokatív
pekné – prídavné meno – ženský rod – jednotné číslo – nominatív
pekné – prídavné meno – ženský rod – jednotné číslo – akuzatív
pekné – prídavné meno – ženský rod – jednotné číslo – vokatív
pekné – prídavné meno – mužský rod – jednotné číslo – neživotné – nominatív
pekné – prídavné meno – mužský rod – jednotné číslo – neživotné – akuzatív
pekné – prídavné meno – mužský rod – jednotné číslo – neživotné – vokatív

Zoznam slov v Tvaroslovníku pre tvar slova dievča:
mača – podstatné meno – stredný rod – jednotné číslo – nominatív
mača – podstatné meno – stredný rod – jednotné číslo – akuzatív
mača – podstatné meno – stredný rod – jednotné číslo – vokatív

Záznam z Tvaroslovníka ukazuje, že pre slovo *mača* i pre slovo *pekné* existuje viacero rôznych záznamov. Hoci prvá iterácia prebehne presne ako v *Príklade 1*, tieto rôzne záznamy pre rovnaké slovo komplikujú rozbor pri piatej iterácii. V tomto príklade sme sa rozhodli bližšie rozpísať iba vybrané iterácie. Zobrazením druhej iterácie chceme ilustrovať ako sa spracovávajú slová, ktoré vo vete susedia, no majú viacero záznamov z Tvaroslovníka. Zameriame sa na dvojicu slov *pekné* a *mača*, pričom predpokladáme, že dvojica slov [*prídavné meno, podstatné meno*] sa dostane do zoznamu potencionálnych vzťahov, len ako zhodný prívlastok (teda sa zhodujú v čísle a páde). Záznam pre slovo *pekné* sa teda dostane do zoznamu vzťahov iba vtedy, ak je v strednom rode.
Utriedený zoznam potencionálnych vzťahov podľa priority: [pekné (pád nominatív), mača], [pekné (pád akuzatív), mača], [pekné (pád vokatív), mača], [okne, na], [na, uvidelo], [dievča (pád akuzatív), uvidelo], [dievča (pád nominatív), uvidelo], [dievča (pád vokatív), uvidelo], [*okne, pekné*]. Z vety odstránime slovo *pekné*. Takisto odstránime zo zoznamu vzťahov všetky tie, ktoré majú ako podradené slovo *pekné*. Strom vety vyskladáme analogicky ako v Príklade 1.
Tretia a štvrtá iterácia prebieha analogicky s Príkladom 1. Ďalšou iteráciou, ktorú rozoberieme bližšie je piata iterácia.

**5.iterácia**

Veta: ***Dievča uvidelo mača***

Utriedený zoznam potencionálnych vzťahov podľa priority:

[dievča (pád akuzatív), uvidelo], [mača (pád akuzatív), uvidelo],
[dievča (pád nominatív), uvidelo], [mača (pád nominatív), uvidelo],
[dievča (pád vokatív), uvidelo], [mača (pád vokatív), uvidelo]
V tomto prípade vzťahy [*dievča (pád akuzatív), uvidelo*] a [*mača (pád akuzatív), uvidelo*] majú rovnakú prioritu(rovnako ako vzťahy, kedy sú tieto podstatné mená obe v nominatíve). Algoritmus nesprávne vyberie slovo *dievča* ako predmet a slovo *mača* označí ako podmet. Je to spôsobené rovnakým tvarom slov v nominatíve aj v akuzatíve. V tomto prípade je nutné poznať aj sématický význam vety. Výsledkom algoritmu by mali byť dva stromy – jeden v ktorom je podmetom slovo *dievča* a druhý, v ktorom je podmetom slovo *mača*.

# 4 Implementácia

Pred implementáciou samotného algoritmu sme najprv automatizovali import samotnej databázy. Keďže Tvaroslovník je uložený v textových súboroch, ktorých je vyše 200 000, nebolo možné tento import urobiť ručne. Vytvorili sme si teda jednoduchú triedu *AutomateImport*, ktorá postupne tieto textové súbory prechádzala a ukladala ich do MySQL databázy. Táto trieda, ako aj samotná implementácia algoritmu je v programovacom jazyku Java. Pri implementácii navrhnutého riešenia vetu zo vstupu spracovávame po slovách. Záznamy o jednotlivých slovách hľadáme v Tvaroslovníku a ukladáme ako inštancie triedy *Slovo*. Dva objekty triedy *Slovo* spolu tvoria objekt triedy *Vzťah*. Každému z týchto objektov sa pri vytváraní priradí *priorita*, ktorá bola spomenutá v kapitole *Analýza a návrh riešenia*. Trieda *Vzťah* implementuje rozhranie *Comparable*, aby sme vedeli pohodlne zoradiť vzťahy podľa *priority* a vhodne ich spracovať. V hlavnej triede *DPGUI* potom spracovávame text zo vstupu po vetách. Výstupom algoritmu je samostatný strom pre každú z viet. Vetu spracovávame po jednom slove a vytvárame inštancie triedy *Slovo*, ktoré ukladáme do zoznamu ***tvarySlov***, s ktorým potom ďalej pracujeme. Opäť zdôrazňujeme, že jedno slovo môže mať v Tvaroslovníku viacero záznamov. Príklad zoznamu ***tvarySlov*** pre vetu z Príkladu 2: ***Dievča uvidelo na okne veľmi pekné mača tvarySlov: dievča, dievča, uvidelo, na, na, okne, okne, veľmi, pekné, pekné , pekné , pekné , pekné , pekné , pekné , pekné , mača, mača, mača***. Na tomto zozname je pomerne dobre viditeľné, že nájsť dvojice slov, ktoré susedia vo vete, nie je to isté, ako nájsť susedov v zozname.

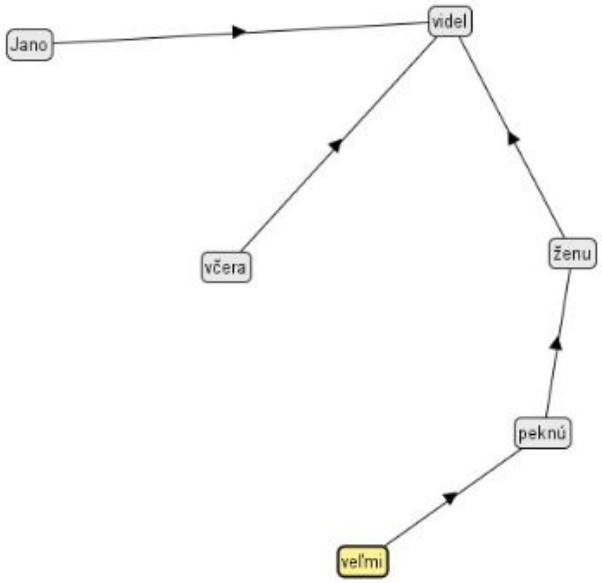
Po vytvorení zoznamu ***tvarySlov*** sa spracujú dvojice (v pôvodnej vete) susediacich slov, z ktorých sa vytvoria inštancie triedy *Vzťah*. Tieto inštancie sa vložia do ďalšieho zoznamu ***vzťahyVoVete***, ktorý sa utriedi podľa priority jednotlivých vzťahov. Ako je už spomenuté v predchádzajúcej kapitole, z tohto zoznamu vyberáme vždy prvý člen. Ten pridávame do zoznamu ***výslednéVzťahyVoVete***, ktorý je výstupom programu. Podradené slovo vybraného vzťahu potom ešte musíme vymazať nielen zo samotnej vety (t.j.: v našom prípade zo zoznamu ***tvarySlov***), ale upraviť musíme aj zoznam ***vzťahyVoVete***. Z tohto zoznamu vymažeme všetky vzťahy, ktorých podradené slovo sa tvarom zhoduje s našim vybraným slovom. V tomto prípade sme nemohli zvoliť typický prístup práca so zoznamom – prechádzaj zoznam – nájdi slovo – spracuj ho. Keďže by sme v takomto prípade upravovali zoznam, ktorý práve čítame, Java ohlási chybovú hlášku *ConcurrentModificationException*. Na spracovanie zoznamu sme teda použili iterátor, implementujúci rozhranie *Iterator*. Rozhodli sme sa neimplementovať toto rozhranie v osobitnej triede, keďže každý objekt triedy *Collections* má svoju vlastnú, korektnú implemnetáciu tohto rozhrania. Obe premenné z ktorých chceme mazať, ***tvarySlov*** aj ***vzťahyVoVete*** sú inštancie triedy ArrayList, čo nám dáva možnosť využiť na získanie iterátora metódu *iterator()*. Po vymazaní vzťahu s najvyššou *prioritou* z oboch zoznamov algoritmus pokračuje ďalším prechádzaním vety (t.j.: zo zoznamu ***tvarySlov***). Toto prehľadávanie skončí, ak:

- jediné slovo vo vete je sloveso
- našli sme vzťah s prioritou označujúcou nemožný vzťah

V prvom spomenutom prípade skontrolujeme, či zoznam ***výslednéVzťahyVoVete*** obsahuje prisudzovací sklad. Ak sa takýto vzťah nenájde, algoritmus spustí metódu na zistenie zamlčaného podmetu

a to skúmaním najprv jednotlivých slovesných kategórii a v prípade nedostatočných informácií aj gramatických kategórií ostatných uzlov stromu. V druhom prípade sme v stave, kedy sú v zozname iba nemožné vzťahy, no stále nemáme prisudzovací vzťah. Typicky tento prípad nastane ak sa vyskytne niektorý so špecifických prípadov spomenutých v predošlej kapitole. Implementácia týchto prípadov ešte nie je dokončené, preto ju v tejto kapitole nebudeme uvádzať

Výstupom algoritmu je teda jeden prípadne viac stromov pre každú vetu, pričom uzly stromu reprezentujú slová vo vete a prepojenia medzi nimi reprezentujú vzťahy. Jednotlivé vzťahy v zozname ***výslednéVzťahyVoVete*** určujú štruktúru stromovej reprezentácie vety. Aby bol výstup v užívateľský príjemnejšej, grafickej, podobe, použili sme na vizualizáciu knižnicu PAZGraphs. Táto knižnica bola vyvinutá na Univerzite Pavla Jozefa Šafárika. Objekty triedy *Vzťah* boli už pri návrhu algoritmu navrhované tak, že ich vizualizáciu bude zabezpečovať práve knižnica PAZGraphs. Nemuseli sme teda meniť jej štruktúru, ani implementovať žiadne dodatočné súčasti.

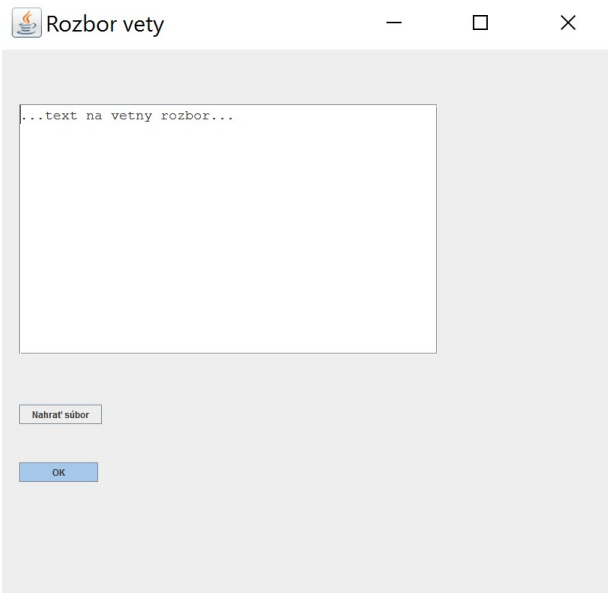



Obr. 8: Výstup algoritmu

# 5 Grafické používateľské rozhranie (GUI)

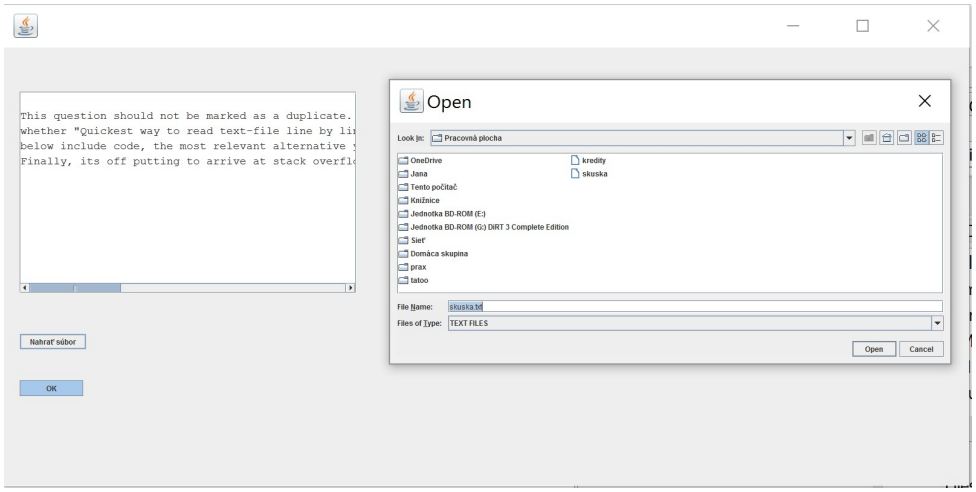
Ako bolo spomenuté v predchádzajúcich kapitolách, výsledný spustiteľný program očakáva od používateľa na vstupe jednu alebo viacero viet. Vetný rozbor prebieha automaticky, neočakáva sa dodatočný vstup od používateľa a teda používateľské rozhranie nemusí mať veľa komponentov. V aktuálnom návrhu obsahuje grafické používateľské rozhranie v hlavnom okne tri komponenty. Hlavným z nich je textové pole, do ktorého používateľ napíše text, na ktorom si želá vykonať vetný rozbor. Ak by bol text rozsiahlejší, program ponúka možnosť načítať ho priamo zo súboru. Aktuálne podporované formáty sú *.txt, .doc a .docx*. Ak je vstup načítavaný zo súboru, jeho obsah sa vloží do textového poľa, kde ho môže používateľ podľa potreby ešte upraviť. Začiatok rozboru potvrdí stlačením tlačidla.

Na implementáciu grafického rozhrania sme použili pomerne robustné API pre vývoj grafických komponentov v jazyku Java – Swing. Pre toto API sme sa rozhodli hlavne preto, že pomerne dobre nasleduje tzv. *MVC (Model-View-Controller)* architektúru. V tejto architektúre je oddelené dáta komponentu (Model), jeho vzhľad (View) a akcia, ktorá sa vykoná pri zmene dát (Controller). Hlavné okno programu je rozšírením triedy *JFrame*, textové pole vstupu je inštanciou triedy *JTextArea*.




Obr. 9: Návrh GUI

V prípade, že si používateľ želá nahrať text zo súboru, po kliknutí na tlačidlo sa zobrazí vyhľadávacie okno, ktoré je inštanciou triedy *JFileChooser*. Pre úplnosť dodávame, že predstavené používateľské rozhranie ešte bude predmetom vizuálnych zmien, zatiaľ čo typ a počet komponentov by sa už meniť nemal.



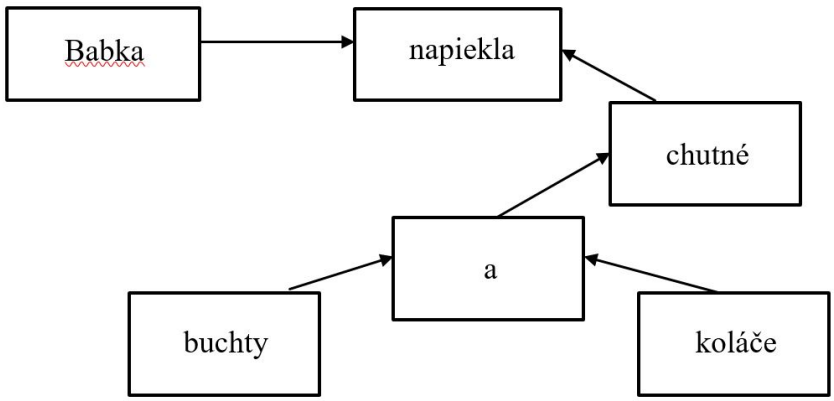

Obr. 10: Vyhľadávacie okno pre načítanie vstupu zo súboru

# 6 Ďalšie smerovanie práce

Aby sme sa mohli v našej práci posunúť k zložitejším vetným konštrukciám, potrebujeme sa vysporiadať s typmi špecifických prípadov, ktoré sme uviedli v druhej kapitole. V **Príklade 2** sme spomenuli jav, pri ktorom má slovo rovnaký tvar v nominatíve aj v akuzatíve a teda môže zastávať funkciu predmetu aj podmetu. Čiastočne vieme tento problém vyriešiť úpravou databázy. Ak by sa pred podmetom nachádzala predložka, vieme jednoznačne určiť jeho vetný člen. Do databázy sme k niektorým predložkám dodali pád, s ktorým sa viažu, čo nám dá ešte viac informácií o danom vzťahu.

Jedným z cieľov, ktorý chceme v práci dosiahnuť je rozbor úplných rozvitých dvojčlenných viet. K naplneniu tohto cieľa nám zostáva implementácia navrhnutého riešenia pre viacnásobné vetné členy (napr.: Babka napiekla chutné koláče a buchty). Charakteristickou črtou tohto lingvistického javu je to, že sú spojené buď niektorou s priraďovacích spojok, prípadne čiarkou. V prípade viacnásobných vetných členov sa spojky nebudú správať inak ako spojky, či predložky vo vete. Vo vzťahu budú nadradené slovám, ktoré spájajú. Navyše by tento nadradený uzol mal uchovávať informácie o gramatických kategóriách, ktoré majú jeho potomkovia spoločné.

Ďalším krokom bude rozbor neúplných viet, teda viet, ktorým chýba podmet. Ako sme už spomenuli v predchádzajúcich kapitolách, algoritmus dokáže zistiť neprítomnosť prisudzovacieho vzťahu. Samotný prísudok je potom dobrým zdrojom bližších informácií o zamlčanom podmete. Aj keď je pravdepodobné, že hlavne pri zamlčanom podmete v 3. osobe jednotného aj množného čísla bude jednoznačné určenie zámena problematické.




Obr. 11: Obrázok 9: Očakávaný výstup pri spracovaní viacnásobných vetných členov

Rozšírením našej práce sú určite *súvetia* (t.j.: vety, ktoré majú dva, prípadne viac prísudkov). Pri týchto vetách nám môže pomôcť jazykovedná teória, ktorá ich rozdeľuje na *priraďovacie* a *podraďovacie*. Priraďovacie súvetia sú spojené priraďovacími spojkami (napr. a, i, alebo ani, či) a spájajú dve rovnocenné vety. V tomto prípade by sa rozbor veľmi nelíšil od aktuálneho prístupu. Ak by sa nám podarilo nájsť spojku, ktorá vety spája, vedeli by sme ich touto spojkou rozdeliť na dve dvojčlenné vety, ktoré už spracovať vieme. Zložitejším prípadom sú podraďovacie súvetia, pri ktorých nie vždy vieme vety takto jednoducho rozdeliť. Typickým príkladom, v ktorom je jedna veta vložená do druhej, je veta ***Jano, ktorý mal vždy dobré známky, študuje na vysokej škole.*** Pri takomto type súvetí sa zohľadňuje nielen podraďovacia spojka, ale aj niektoré interpunkčné znamienka.

# 7 Podobné práce

Ako sme už naznačili v úvode tejto práce, počítačová lingvistika na Slovensku je pomerne málo obľúbenou oblasťou výskumu. Tento fakt síce dáva možnosť odhalenia úplne nového prístupu k riešeniu problémov, no na druhej strane prakticky odpadá možnosť porovnávať prácu, či dosiahnuté výsledky. V tejto kapitole sa preto budeme venovať skôr poukázaniu na ďalšie prístupy, ktoré sa dajú pri počítačovom spracovaní prirodzeného jazyka použiť.

Zaujímavým prínosom do tejto oblasti je určite článok ***Spracovanie prirodzeného jazyka pre interaktívne rečové rozhrania v slovenčine*** od autorov *Staš J., Hládek D., Ondáš S., Zlacký D. a Juhár J.*, ktorí pôsobia na Technickej Univerzite v Košiciach. V ich článku používajú ako korpus dáta vydolované prostredníctvoj špeciálneho nástroja z rôznych článkov, či fór v sieti Internet. Tento veľký korpus dát sa potom rozdelí do podkorpusov podľa toho, aký veľký je prienik fráz, či kľúčových slov medzi dokumentami. Podstatný rozdiel oproti našej práci je samotná existencia korpusu, ktorú nevyžadujeme. Ďalší podstatný rozdiel oproti našej práci je samotné spracovanie textu. Autori na vydolovanom texte najprv vykonajú morfologickú analýzu, na ktorú použili nástroj *Dagger*, ktorý je vlastne implementáciou štatistického *Hidden – Markovho modelu*. V našej práci tento krok nie je nutný, keďže máme k dispozícii Tvaroslovník, ktorý obsahuje slová aj ich morfémy.

Na tomto mieste by som chcela pripomenúť aj konferenciu *SLOVKO*, ktorá sa koná každédva roky. Konferencia má za úlohu priblížiť výskum v oblasti počítačového spracovania jazykov a to nie len slovenčiny, ale aj češtiny, či slovinčiny. Hlavným cieľom je nie len informovať o novinkách v tejto oblasti, ale aj zjednotiť vedcov a výskumníkov, ktorí pracujú s flektívnymi jazykmi. Z poslednej konferencie je zaujímavé spomenúť napríklad článok ***Corpus of Dialects of the Slovak National Corpus*** od autorov *Gajdosová K., Garabík R. a Simková M.*, ktorý by mohol v budúcnosti rozšíriť našu prácu zo spracovania spisovného jazyka k spracovaniu reči v slovenských dialektoch.

Veľkým prínosom na medzinárodných konferenciách sú práce z Ústavu aplikovanej lingvistiky z matematicko – fyzikálnej fakulty z Karlovej Uvinerzity v Prahe. Práve tu vznikol aj projekt Pražského závislostného korpusu, spomínaný v úvode tejto práce. Väčšina prác a článkov z českých vysokých škôl pracuje práve s týmto korpusom. Pre nás najzaujímavejším bol článok ***A case study of a free word order*** od autorov *Kuboň V., Lopatková M. a Mírovský J.*, v ktorom bol spomenutý koncept *redukčnej analýzy*. Autorský kolektív opäť používa pri práci s textom *PDT* a teda má k dispozícii detailnejšie informácie o slovách vo vete. V článku sa sústreďujú na spracovanie *projektívnych viet*. Na začiatku našej práce sme sa teda aj my sústredili na projektívne vety .

# Literatúra

[1] Projekt Tvaroslovník [online] Dostupné na internete: <http://tvaroslovnik.ics.upjs.sk/>

[2] Krátky slovník slovenského jazyka. Red. J. Kačala – M. Pisárčiková – M. Považaj. 4. dopl. a upr. vyd. Bratislava: Veda 2003. 985 s. ISBN 80-224-0750-X

[3] A case study of a free word order 2012 Kuboň V., Lopatková M. a Mírovský J.

[4] Konferencia SLOVKO [online] Dostupné na internete: <http://korpus.sk/ slovko/2015/>