Masarykova univerzita
Fakulta informatiky

Extrakcia informácií z burzových správ

Diplomová práca

Jozef Štyrák

Brno, 2014

Prehlásenie

Prehlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Jozef Štyrák

Vedúci práce: Mgr. Marek Grác, Ph.D.

Poďakovanie

Rád by som na tomto mieste poďakoval vedúcemu mojej diplomovej práce, Mgr. Marekovi Grácovi, Ph.D., za odbornú pomoc a mojej rodine a priateľom za podporu.

Zhrnutie

Táto práca sa venuje návrhu a vývoju systému pre extrakciu informácií z textov burzových správ. Úlohou vyvíjaného systému je automatická identifikácia a extrakcia nákupných doporučení z voľného textu správ. Na danej úlohe je ilustrovaná problematika počítačového spracovania českého jazyka. Postupne sú na každej jazykovej úrovni popísané úlohy a problémy, ktoré je potrebné riešiť. Práca sa okrem toho špeciálne zameriava na extrakciu informácie ako aplikáciu počítačového spracovania prirodzeného jazyka, pričom je v texte uvedených niekoľko príkladov systémov extrakcie informácií pre český jazyk. Priestor je venovaný tiež nástrojom pre spracovanie češtiny vyvíjaným v Centre spracovania prirodzeného jazyka (CZPJ), ktoré boli využité pri vývoji. Cieľom je preskúmať ich praktické využitie pri tvorbe spomínaného systému.

Kľúčové slová

extrakcia informácií, named entity recognition, burzové správy, morfologická analýza, syntaktická analýza, nlp, rezolúcia anafory, valenčné rámce

Obsah

1 Úvod

Český jazyk patrí medzi flektívne jazyky a vďaka bohatej morfológií a pomerne voľnému slovosledu je vývoj nástrojov a aplikácií preň často komplikovaný. Kým v anglickej literatúre je morfologickej analýze venovaná minimálna pozornosť, pri čestine ide o komplexný problém, ktorý ovplyvňuje aj ďalšie roviny spracovania jazyka. Nástroje vyvinuté pre anglický jazyk (alebo ostatné svetové jazyky) nie sú vo väčšine prípadov kompatibilné s češtinou, rovnako ani jazykové modely a nástroje pre češtinu nie sú štandardnou súčasťou populárnych knižníc pre spracovanie prirodzeného jazyka. Vývoj nástrojov sa aj preto často odohráva vo vedeckom prostredí na oddeleniach univerzít. Už dlhodobo je Centrum zpracování přirozeného jazyka (ďalej CZPJ) Fakulty informatiky Masarykovej univerzity v popredí štúdia a výskumu počítačovej lingvistiky. Každý rok sa tu napíše viacero vedeckých publikácií a vytvorí niekoľko nástrojov.

V práci sa budeme venovať problematike českého jazyka z pohľadu aplikačného programátora. Preskúmame, aké nároky kladie čeština na návrh systému a aké nástroje je potrebné využiť pri jeho implementácii. Priestor v textovej časti bude venovaný nástrojom vyvíjaným v CZPJ a zároveň ich funkcionalita bude otestovaná pri návrhu a implementácii vlastného systému. Systém nebude určený priamo pre koncového uživateľa (vývoj uživateľského rozhrania nie je súčasťou diplomovej práce), bude skôr vzorovým príkladom aplikácie pracujúcej s českým jazykom.

Aplikácie v oblasti spracovania prirodzeného jazyka sú rôznorodé vzhľadom na predmet skúmania či prístup k riešeniu. Medzi príklady patria gramatické korektory, full-text vyhľadávače, dotazovacie systémy, systémy pre rozpoznávanie hovorenej reči alebo strojové prekladače. Jednou z aplikácií je aj extrakcia informácií z neštruktúrovaného textu, pri ktorej je potrebné pracovať s jazykom na viacerých rovinách, a preto je aj vhodnou voľbou pre ukážkovú aplikáciu.

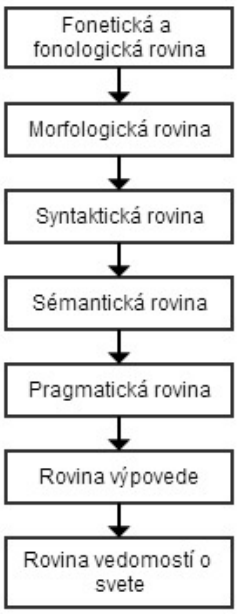
Systémy extrakcie informácie môžu byť zamerané na extrakciu špecifickej informácie, alebo môžu z textu extrahovať všetky informácie, ktore sa im podarí detekovať. Nami vytvorený systém bude zameraný na extrakciu špecifickej informácie o zmene nákupných doporučení pre trhové entity z textov burzových správ. Text burzových správ je charakteristický vecnosťou a stručnosťou, zároveň však obsahuje vyšší počet skratiek, menných entít a numerických hodnôt, doménovo špecifické výrazy a bezslovesné vety. Tieto charakteristiky robia danú úlohu zaujímavou.

V prvej kapitole sa budeme venovať problematike spracovania prirodze-

ného jazyka s dôrazom na lingvistické vlastnosti češtiny. Okrem všeobecnej informácie bude hlavnou náplňou kapitoly prehľad viacerých nástrojov vyvíjaných v CZPJ.
Témou druhej kapitoly bude popis extrakcie informácie ako aplikácie spracovania prirodzeného jazyka. Okrem popisu úlohy, jednotlivých fáz extrakcie a rôznych prístupov k nej sa zameriame aj na jej aplikácie pre český jazyk a uvedieme niekoľko príkladov.
V tretej kapitole sa budeme zaoberať systémom, ktorý bol navrhnutý na základe nadobudnutých poznatkov. Podrobnejšie rozoberieme doménu textov, nad ktorými systém pracuje, a upresníme jeho úlohy a informáciu, ktorú bude extrahovať. Postupne budú na rôznych jazykových rovinách rozobrané problémy, na ktoré sme pri analýze a návrhu narazili, a akým spôsobom bolo potrebné dané problémy riešiť. Posledná kapitola obsahuje dosiahnuté výsledky a analýzu chýb. Na záver budú zhodnotené určené ciele a vlastný prínos.

2 Nástroje pre spracovanie českého jazyka

Cieľom nasledujúcej kapitoly je podať stručný prehľad vybraných tém zo spracovania prirodzeného jazyka s dôrazom na špecifiká češtiny. Postupovať pri tom budeme na základe rovinového modelu[13], ktorý je zobrazený na obrázku 2.1. Spracovanie prirodzeného jazyka (NLP) je však komplexná úloha, priestor jednej kapitoly nepostačuje ani na rozbor jednej samostatnej roviny. Preto sa budeme venovať skôr praktickým aspektom analýzy a oblastiam, ktoré budú neskôr využité v implementačnej časti, či už ide o teóriu alebo konkrétne nástroje. Vytvorený systém využíva viaceré nástroje z CZPJ, ktorým bude tiež venovaný priestor v kapitole. Uvedieme ich stručný prehľad a tiež prípadné porovnanie s inými českými nástrojmi, najmä nástrojmi z Ústavu formálnej a aplikovanej lingvistiky (ďalej ÚFAL) Matematicko-fyzikálnej fakulty Karlovej Univerzity. Praktická skúsenosť s využitými nástrojmi bude navyše rozobraná v kapitole 4.



Obr. 2.1: Roviny spracovania prirodzeného jazyka

2.1 Roviny spracovania prirodzeného jazyka

Typický systém v oblasti spracovania prirodzeného jazyka pracuje s komplexnými dátami. Aplikácia môže prijímať text v písanej alebo v hovorenej podobe, v rôznych jazykoch, dialektoch, písaný text môže pozostávať z písmen rôznych abeced, pri hovorenom slove je kľúčova tiež osoba rečníka, každé slovo môže nadobúdať viacero tvarov, môže byť použité v rôznych kontextoch. Ako bolo spomenuté v úvode kapitoly, obrázok 2.1 ilustruje roviny vedomostí o jazyku, ktoré sú relevantné pre tvorbu systému NLP.
Na úrovni fonetiky a fonológie sa zaoberáme tým, ako sa slová vzťahujú na zvuky, ktoré ich identifikujú. Tieto vedomosti sú kritické pre aplikácie pracujúce s hovorenou rečou.
Morfológia sa zaoberá spôsobom, akým sa morfémy spájajú do slov. Morféma je najmenšou lingvistickou jednotkou, ktorá môže niesť význam. Problematika morfológie je obzvlášť dôležitá pre štúdium flektívnych jazykov, medzi ktoré patrí aj čeština.
Jazyková syntax študuje to, akým spôsobom sa slová navzájom viažu a tvoria korektné frázy a vety.
Sémantika sa zaoberá významom slov a tým, ako sa tieto významy kombinujú a tvoria význam vety. Tento význam je kontextovo nezávislý, tzv. význam sa nemení v závislosti od kontextu, v ktorom sa veta nachádza. Na druhej strane, pragmatika študuje, ako môže byť tá istá veta použitá v rôznych situáciach a ako to ovplyvňuje jej interpretáciu.
Analýza výpovede sa zaoberá vetou ako časťou textu. Skúma, ako predchádzajúce vety ovplyvňujú interpretáciu danej vety. Dôležitým pojmom jej štúdia je anafora, vzťah antecedenta a koreferenta a temporálny aspekt informácie obsiahnutej vo vete.
Úroveň vedomostí o svete obsahuje všeobecné vedomosti o štruktúre sveta, ktoré je nutné ovládať, aby človek mohol napr. viesť konverzáciu.

2.2 Morfologická analýza

Čeština sa vyznačuje bohatou morfológiou, čo je jeden z dôvodov, prečo úlohu automatickej morfologickej analýzy nie je možné aj napriek neustálemu výzkumu považovať za vyriešenú. Proces morfologickej analýzy pozostáva z dvoch úloh:

- morfologické značkovanie
- desambiguácia na morfologickej úrovni

Pri jazykoch ako je angličtina sa desambiguácia uskutočnuje priamo počas morfologického značkovania (Part-of-Speech tagging), avšak v kontexte českého jazyka je táto činnosť komplexnejšia, a preto je potrebné ju vyčleniť samostatne.
Úlohou morfologickej analýzy je typicky pre každé vstupné slovo vrátiť jeho základný tvar (lemma) a gramatické kategórie pre daný tvar. Gramatické kategórie sú zakódované do morfologickej značky. Pre češtinu sú najčastejšie používané dva formáty. Prvým je pozičný systém, ktorý bol vytvorený v ÚFAL [28]. Každá gramatická kategória je jednoznačne určená svojou pozíciou v značke, pričom celkový počet pozícií je 15. Napríklad, značka *NNMS1−−−−−A−−−−* popisuje obyčajné substantívum mužského rodu v nominatíve jednotného čísla a je afirmáciou.
Druhým systémom je brnenský atribútový systém [1]. Značka pozostáva z dvojíc atribút–hodnota, pričom nezáleží na poradí dvojice v značke. Gramatická kategória je identifikovaná malým písmenom a jej hodnota písmenom veľkým alebo číslicou. Predchádzajúca pozičná značka by vyzerala v atribútovom systéme takto: *k1gMnSc1*. V atribútovej značke podstatných mien nie je zachytená informácia o špecifikácii a afirmácii. Tento systém je predvolený pre nástroje vyvíjané v CZPJ.

2.2.1 Morfologické analyzátory

V nasledujúcej kapitole sa budeme venovať dvom morfologickým analyzátorom využívaným v CZPJ, analyzátorom Ajka [45] a Majka [32].
Vo svojej práci [45] sa autori vyjadrujú k problematike českej morfológie a jej algoritmickému popisu. Ako základné slovotvorné procesy uvádzajú inflekciu (skloňovanie a časovanie), deriváciu (odvodzovanie) a skladanie slov. Slovo sa skladá zo základných jednotiek – morfém. Morféma je najmenšia stavebná jednotka, ktorá môže niesť význam, pričom pre gramatické kategórie je typicky najdôležitejšia morfologická koncovka. Základným pojmom pre algoritmický popis je morfologický vzor. Vzor reprezentuje generalizáciu skloňovacích (v prípade slovies časovacích) koncoviek, na základe vedomostí o inflekcii vzoru vieme určiť inflekčné koncovky každého slova danej triedy. Okrem koncoviek musí analyzátor brať do úvahy aj častú zmenu intersegmentu (napr. ho-r-a, ho-ř-e). Čeština obsahuje vysoký počet slov s nepravidelnou inflekciou. Ajka tento problém rieši vytvorením nového vzoru, pričom počet vzorov v úplnom systéme (zahrnuté sú aj archaizmy a iné málo sa vyskytujúce tvary) dosahuje číslo 1500.
Ako vyplýva z predchádzajúceho odstavcu, zvolená dátová štruktúra je kritická pre efektívnosť analyzátora. Autori využili dátovú štruktúru trie

implementovanú vo forme minimálneho konečného automatu [16], z dôvodu minimalizácie pamäťových nárokov. Pri morfologickej analýze vybraného slova algoritmus segmentuje slovo na jednotlivé morfémy a na ich základe určí gramatické kategórie.
Základným kritériom pre evaluáciu analyzátora je počet slov, ktoré rozoznáva. Ich počet závisí na kvalite slovníka, autori v texte uvádzajú počet kmeňov na 223600 a celkový počet rozoznávaných slov na 5678122.
Ajka sa však ukázala v praxi ako príliš pomalá. Tento problém sa snaží odstrániť analyzátor Majka, ktorý je implementáciou prístupu navrhnutého v [33]. Prístup je založený na využití algoritmu Jana Daciuka [15] – na vytvorení minimálneho deterministického acyklického konečného automatu (DAFSA) a jeho využiti pri morfologickej analýze. Základná myšlienka je jednoduchá. Každý konečný zoznam unikátnych reťazcov je možné považovať za konečný jazyk, a tak môže byť reprezentovaný DAFSA. Ak je minimálny, tak existuje jediná cesta zodpovedajúca určitému podreťazcu nejakej podmnožiny modelovaných reťazcov. Samotná analýza spočíva v rýchlom vyhľadávaní v automate.
Dáta sú uložené vo forme zoznamu všetkých kombinácií rozoznávaného vstupu a zodpovedajúceho výstupu. Dané dve slová sú zakódované vo forme páru tvoreného prvým slovom a rozdielom medzi oboma slovami. Napríklad:

```
klouček:A,k1gMnSc1
kloučka:Cek,k1gMnSc2
kloučka:Cek,k1gMnSc4
```

Dvojbodka oddeľuje vstup od výstupu, za ňou nasleduje postup, ako vytvoriť k danemú tvaru lemma. Písmená A a C určujú na základe svojho poradia v abecede, koľko znakov je potrebné vymazať z pravej strany (0 a 2 v danom príklade), znaky na konci sa následne pridajú na koniec slova. Za čiarkou nasleduje morfologická značka. Samotná analýza je jednoduchá a rýchla. Ak sa analyzovaný vstup zreťazený s dvojbodkou nachádza v automate, potom každá cesta do koncového stavu reprezentuje možnú analýzu daného slova. Na výstupe analýzy vstupu *kloučka* bude teda lemma *klouček* a značky *k1gMnSc2* a *k1gMnSc4*. Obdobne je potrebné vytvoriť dáta v špeciálnom formáte pre generovanie tvarov či segmentáciu.
Je možné si všimnúť, že pri analýze sa neuplatňuje žiaden sofistikovaný algoritmus s gramatickým modelom, analýza pozostáva z rýchleho slovníkového vyhľadávania. Analyzátor Majka je vďaka tomu v porovnaní s Ajkou niekoľkonásobne rýchlejší.

2.2.2 Desambiguácia na morfologickej úrovni

Bežným javom na morfologickej úrovni je morfologická homonymia, kedy jeden slovný tvar je možné interpretovať rôznymi gramatickými kategóriami. Z hľadiska analýzy to znamená, že pre daný vstup analyzátor dá na vystup viac rôznych výstupov, či už v podobe rôznych lemmat alebo morfologických značiek. Známym príkladom je tvar *ženu*. Tento tvar môže byť interpretovaný ako akuzatív feminina *žena* a zároveň ako prvá osoba singuláru slovesa *hnát*. V tabuľke 2.1 je možné vidieť homonymiu slov v jednoduchej vete.

Slovo	**Značky**	**Význam**
Správny	k2eAgMnPc4d1wH, k2eAgInPc1d1wH, k2eAgInPc4d1wH, k2eAgInPc5d1wH a 11 ďalších	prídavné meno
postup	k5eAaPmRp2nS, k5eAaPmRp2nS, k1gInSc1, k1gInSc4	podstatné meno, sloveso
je	k5eAaImIp3nS, k3p3gMnPc4, k3p3gInPc4, k0 a 3 ďalšie	sloveso, zámeno, citoslovcia
důležitý	k2eAgMnSc1d1, k2eAgMnSc5d1, k2eAgMnPc4d1wH, k2eAgInPc1d1wH a 19 ďalších	prídavné meno

Tabuľka 2.1: Morfologická homonymia v jednoduchej vete

Prístupy k desambiguácií je možné rozdeliť na pravidlové a štatistické, alebo ako je to v prípade [21], oba tieto prístupy je možné skombinovať navyše ešte so skrytými Markovovými modelmi. Nevýhody manuálnej tvorby pravidiel (časová náročnosť, nutnosť lingvistických vedomostí, údržba) sa pokúšajú odstrániť autori systému Desamb [31] pomocou metódy učenia sa pravidiel z neanotovaných dát. Systém vznikol v CZPJ.

Ako základnú myšlienku autori uvádzajú náhodnosť povahy mnohých homonymií. V predchádzajúcom príklade má síce akuzatív feminina rovnaký tvar ako sloveso, avšak je veľa feminin, ktorých akuzatív sa nezhoduje s prvou osobou singuláru nejakého slovesa. Autori predpokladajú, že morfologická značka vyjadruje funkciu slova vo vybranom kontexte a túto funkciu bude pravdepodobne spĺňať aj v ďalších kontextoch, kde sa vyskytuje. Pri desambiguácií medzi značkami X a Y je potrebné nájsť v korpuse jednoznačný výskyt oboch značiek. Na základe nájdených kontextov pre značky X a Y a porovnania ich vlastností s kontextom aktuálne desambiguovaného slova by malo byť možné určiť, ktorú značku vymazať.
Ako učiaci sa algoritmus nad trénovacími dátami pre generovanie pravidiel autori zvolili algoritmus induktívneho logického programovania [35]. Na kódovanie kontextu boli využité prologovské fakty, pričom samotné slová v kontexte dôležité nie sú, podstatná je morfologická značka slov v kontexte. Algoritmus následne konštruuje množinu pravidiel popisujúcich pozitívne príklady pre danú značku.
Ďalšie informácie týkajúce sa samotného algoritmu a spôsobu evaluácie je možné nájsť v [31].

```
Správny správný k2eAgInPc1d1wH
postup  postup  k1gInSc1
je      být     k5eAaImIp3nS
důležitý        důležitý        k2eAgInSc1d1
```

Obr. 2.2: Desambiguovaný výstup programu Desamb

2.3 Syntaktická analýza

Úlohou syntaktickej analýzy je rozoznať vzťahy medzi slovami, a to, ako sa slová spájajú do fráz a viet. Čeština je jazyk s pomerne voľným slovosledom, čo vplýva na náročnosť analýzy.
Pri syntaktickej analýze, podobne ako pri morfologickej, môžeme tiež hovoriť o mnohoznačnosti. Ako príklad môže poslúžiť veta z [29]: *Karel pronásledoval muže na kole*. Z takto fomulovanej vety bez znalosti kontextu nie je možné určiť, kto *na kole* sedel, či *Karel* alebo *muž*.
V súčasnosti existujú tri hlavné prístupy k syntaktickej analýze češtiny:

- závislostný prístup
- zložkový prístup

- metóda postupnej segmentácie vety

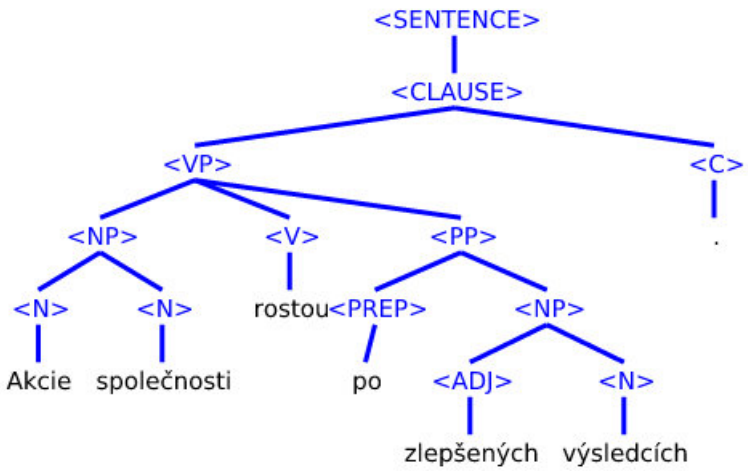
Závislostný prístup bol vyvinutý (a je stále rozvíjaný) v ÚFAL. Za výstup analýzy je považovaný koreňový strom (orientovaný acyklický graf s vyznačeným koreňom). Každému uzlu v grafe zodpovedá jedno slovo zo vstupnej vety (koreň je vyčlenený samostatne). Vzťahy medzi slovami sú reprezentované hranami. Každy uzol je navyše ohodnotený syntaktickou funkciou, ktorá určuje typ závislosti na slove, s ktorým je daný uzol spojený hranou (v smere ku koreňu). V ÚFAL bolo vyvinutých mnoho analyzátorov, pričom väčšina z nich pracuje na podobnom princípe. V prvej, trénovacej fáze sa učia pravidlá na dátach Pražského závislostného korpusu (PDT, z anglického Prague Dependency Treebank) [20], ktoré sú následne použité pri analýze viet, tzv. v druhej fáze [29].
Zložkový prístup je rozvíjaný najmä v CZPJ. Počas analýzy identifikuje zložky (konštituenty), ktoré, narozdiel od závislostných analyzátorov, môžu predstavovať aj väčšie vetné štruktúry (frázy) ako slová, a snaží sa určiť, ako sa tieto zložky spájajú do vetného celku. Výsledkom analýzy je zložkový strom, v ktorom vnútorné uzly predstavujú frázové zložky a listy jednotlivé slová vety. Príkladom zložkového analyzátora je analyzátor synt, ktorý pri analýze využíva meta-gramatiku, ktorá pokrýva viac ako 90% českých viet [25].

2.3.1 Metóda postupnej segmentácie a analyzátor SET

Metódu postupnej segmentácie predstavujú autori analyzátora SET vo svojej práci [30] ako alternatívu k starším prístupom (viď vyššie), vhodnú najmä pre jazyky s voľnejším slovosledom. Ich prístup je založený na postupnej aplikácií pravidiel na analyzovanú vetu a jej časti. Pod pravidlom je možné rozumieť sekvenciu slovných druhov a vetných prvkov (napr. čiarka), ktoré analyzátor hľadá v danej vete. Výber pravidiel je determinovaný viacerými faktormi, ako je pravdepodobnosť výskytu daného pravidla, jeho pozícia vo vete alebo kolokačné štatistiky pre slová, ktoré má pravidlo popisovať. Na základe množiny pravidiel sa algoritmus pokúša identifikovať najlepší subor pravidiel pre danú vetu.
Parsovací algoritmus predpokladá, že každá vstupná veta je gramatická, a preto je ho možné implementovať ako parser regulárnych výrazov, a tým sa vyhnúť tomu, že prirodzený jazyk je z hľadiska Chomského hierarchie minimálne bezkontextový.
Pravidlá sú usporiadané do šiestich vrstiev, ktoré sú na vstup aplikované postupne. Najskôr sú použité pravidlá identifikujúce väčšie vetné celky, vedľajšie vety a pod. Posledná vrstva určuje frázy, pri ktorých je možné

určovať závislosti, ako sú menné alebo slovesné frázy. Príklad výstupu je možné si pozrieť na obrázku 2.3.

Pre vývojárov analyzátora SET poskytuje prehľadný formát pre definíciu



Obr. 2.3: Frázový výstup programu SET z webového rozhrania [9]

pravidiel. Každé pravidlo sa skladá z dvoch častí, zo vzoru (template) a súboru akcií, ktoré sa vykonajú, pokiaľ je možné daný vzor aplikovať na niektorú sekvenciu slov vo vete. Vzor pozostáva zo sekvencie slovných druhov, ktoré tvoria dané pravidlo. Akcie predstavujú inštrukcie pre analyzátor, ako má interpretovať dané pravidlo (priradenie frázovej značky, vyznačenie závislosti medzi slovami, a pod.). Na obrázku 2.4 je znázornené pravidlo identifikujúce frázu pozostávajúcu z číslovky a podstatného mena. Časť „...“ v danom vzore reprezentuje voľné miesto, na ktoré je možné dosadiť jedno alebo viac ľubovoľných slov. Druhý riadok predstavuje súbor akcií. Konkrétne toto pravidlo vyžaduje gramatickú zhodu čislovky a podstatného mena v páde, rode a čísle, a navyše analyzátor vyznačí závislosť číslovky na podstatnom mene. Príkladom vyhovujúcej frázy môžu byť frázy *tri psy* alebo *tri velké psy*.

SET je implementovaný v jazyku Python, ako vstup vyžaduje označkovaný

```
TMPL: numeral ... noun
        AGREE 0 2 cgn MARK 0 DEP 2
```

Obr. 2.4: Príklad pravidla analyzátora SET

text vo formáte, kde každé slovo je na osobitnom riadku. Webové rozhranie, dokumentáciu a ďalšie rozširujúce informácie je možné nájsť na webových stránkach projektu [9], prípadne v práci [29].
Aj napriek neustálemu vyvoju automatická syntaktická analýza češtiny stále nie je vyriešený problém. Závislostné analyzátori dosahujú presnosť do 84% [24], synt medzi 70–90% [26] a set 76–95% [30].

2.4 Sémantická analýza

Na sémantickej rovine sa analýza zaoberá významom slov a fráz, a tým, ako sa spájajú a tvoria význam vety. Oblasť sémantiky je veľmi široká, môžeme pri nej hovoriť o lexikálnom význame alebo o logických reprezentáciach vety, môžeme sa zároveň zaoberať rôznymi pojmami, ako sú ontológia, slovník, rôzne druhy logík a podobne. V nasledujúcej kapitole sa budeme zaoberať iba jedným pojmom, ktorý je dôležitý pre implementačnú časť. Týmto pojmom sú valenčné rámce slovies.

2.4.1 Valenčné rámce slovies

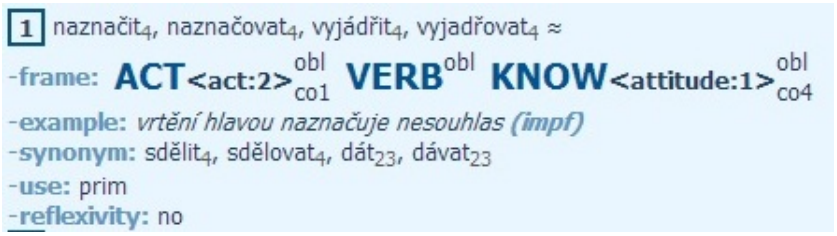
Valenčné rámce sú javom na rozhraní syntaxe a sémantiky. Sloveso sa viaže vo vete pomocou morfologických prostriedkov, ako sú pády a predložkové spojenia. Avšak slovesnú valenciu môžeme tiež chápať ako významom ovplyvnenú schopnosť slovesa viazať sa s ďalšími slovami. Slovesné valencie preto budeme chápať ako jav najmä sémantický [38].
Sloveso je najčastejšie centrom vety, a preto je štúdium jeho syntaktického správania dôležité pre viaceré oblasti spracovania jazyka, ako sú morfologická analýza, desambiguácia, syntaktická analýza a v neposlednom rade sémantická analýza [48]. Na sloveso sa môžeme pozerať vo vete ako na predikát a na ostatné vetné členy ako na jeho argumenty. Napríklad vetu *Agentura zvyšuje doporučení* môžeme zapísať ako predikát *zvyšovat(agens:agentura, paciens:doporučení)*.
Informácie o valenčných rámcoch slovies sú zaznamenávané v špeciálnych slovníkoch. Pre češtinu ich vzniklo (a stále vzniká) viacero. Ako príklady môžu poslúžiť slovník Brief [39] vyvinutý na Fakulte Informatiky Masarykovej Univerzity a slovník Vallex [48] vyvinutý v ÚFAL. Výrazný vplyv na vývoj v oblasti valenčných rámcov mala aj práca na českom WordNete [40] počas projektu Balkanet [3], kedy bol český WordNet obohatený o slovník valenčných rámcov. Ďalším slovníkom, ktorý vznikol v CZPJ je slovník VerbaLex [22], ktorému sa budeme venovať v samostatnej podkapitole. Ešte

predtým však stručne popíšeme valenčný rámec ako taký.
Formát valenčného rámca pre určité sloveso sa líši v závislosti od vybraného slovníka, avšak niektoré informácie sa nachádzajú vo väčšine slovníkov. Rámec nesie informáciu o slovese, ktoré je možné dosadiť do daného rámca, a zároveň aj zoznam k nemu synonymických slov, pre ktoré daný rámec platí tiež. Rámec často obsahuje rozširujúce informácie, ako je napríklad dokonavosť alebo sémantická trieda slovesa.
Rámec v užšom slova zmysle (konkrétne spojenie argumentov a slovesa) pozostáva z argumentov a popisu toho, ako sa viažu na dané sloveso. Argumenty sú najčastejšie reprezentované sémantickou rolou a viažu sa na sloveso pomocou pádov, predložiek, pronominálnych výrazov, infinitívu, atď. Jednotlivé doplnenia (argumenty) môžeme rozdeliť na základe vzťahu k danému rámcu na integrálne, obligatórne, fakultatívne, stredné, voľné a periférne [46]. Na obrázku 2.5 je znázornený vybraný rámec zo slovníka VerbaLex pre sloveso *naznačit*. Upresnenie niektorých informácií je uvedené v nasledujúcej podkapitole.

2.4.2 VerbaLex

Slovník VerbaLex čerpá zo zdrojov spomenutých v predchádzajúcom odstavci, na ktorých základe autori navrhli nový formát pre zápis valenčných rámcov – komplexný valenčný rámec (ďalej CVF, z anglického Complex Valency Frame). V nasledujúcej kapitole uvediem rozdiely prístupu VerbaLexu k slovesným valenciám.
Lexikálne jednotky vo WordNete sú usporiadané do synsetov. VerbaLex sa od slovníku VALLEX odlišuje tým, že synset obsahuje slovesá (v tvare lemmy) s ich významovým číslom. V kontexte synsetu pod synonymiou rozumieme veľmi úzky vzťah medzi jednotlivými slovami. Nie vždy je možné nahradiť každé slovo jeho synonymom, a preto do daného synsetu patrí práve sloveso s významom, ktorý je identifikovaným jeho významovým číslom.
V synsete sa nachádza tiež informácia o dokonavosti každého jeho slovesa (imperfektum, perfektum, alebo oba). Synset ďalej osahuje informáciu o použití daného slovesa, či ide o doslovný význam, prenesený význam alebo frazeologizmus. VerbaLex ďalej člení slovesá do viacerých sémantických tried.
Najpodstatnejší rozdiel oproti ostatným slovníkom je v prístupe k sémantickým rolám. Verbalex predstavuje dvojúrovňové sémantické role. Prvá úroveň obsahuje hlavné sémantické role na základe EuroWordNet Top Ontology [49]. Na druhej úrovni sa nachádzajú špecifické role s vyznačeným

číslom významu. Vďaka tomu je možné špecifikovať slová (v hyponymickom vzťahu), ktoré je možné vložiť na daný slot v slovesnej valencii. Ako príklad poslúži nasledujúci sémantický popis subjektu pre ľubovoľné sloveso. Subjekt (slot na ľavej strane od slovesa vo valenčnom rámci) má najčastejšie sémantickú rolu agenta (AG). Táto rola je však príliš obecná. Tento nedostatočný popis rieši sémantická rola druhej úrovne, ktorá môže rozšíriť význam subjektu, či ide o osobu (*AG<person:1>*), zviera (*AG<animal:1>*) alebo napríklad o skupinu ľudí *AG<group:1>*. Tento prístup umožňuje bohatú diferenciáciu medzi rolami a významami, ktoré prislúchajú jednotlivým slotom vo valenčnom rámci.
VerbaLex v popise rámca uvádza tiež pozíciu slovesa pomocou špeciálnej role VERB. Ďalšou špeciálnou rolou je rola ISUB pre inexplicitný subjekt, ako napr. vo vete *Daří se mu*.
Pre potreby editácie a exportu slovníka boli vytvorené viaceré interak-



Obr. 2.5: Príklad valenčného rámca zo slovníka VerbaLex

tívne nástroje. Hlavným nástrojom pre používateľov je verbalex.sh, založený na multi-platformovom editore VIM. Verbalex.sh poskytuje v súčasnosti viacero funkcií, ako editáciu záznamov v slovníku, vyhľadávanie v slovníku na základe regulárnych výrazov, pridávanie nových sémantických rolí, export do XML, atď.

2.5 Analýza výpovede

Na úrovni výpovede analyzujeme vetu ako súčasť väčšieho celku – textu. Skúmame, ako predchádzajúce a nasledujúce vety ovplyvňujú jej interpretáciu. Najdôležitejšími pojmami na tejto úrovni sú koreferencia a anafora.
Pod pojmom rezolúcia koreferencie budeme rozumieť identifikáciu fráz, ktoré označujú tú istú entitu. Na druhej strane pod rezolúciou anafory budeme rozumieť identifikáciu konkrétnej frázy (antecedent), na ktorú anaforický výraz (anafora) odkazuje [50]. Anaforu v užšom zmysle slova chápeme ako vzťah k antecedentovi, ktorý sa vyskytuje v texte pred korefe-

rentom, katafora označuje vzťah smerom k nasledujúcim vetám a endofora (vnútorná referencia) zahŕňa oba tieto javy. Na druhej strane pojem exofora (vonkajšia referencia) označuje entitu v určitom svete priamo (napr. Vltava, Brno, Václav Klaus,...). V ďalšom textu budeme využívať pojem anafora v širšom zmysle slova, ktorý sa zhoduje s obsahom pojmu endofora.
Anafora môže byť v texte vyjadrená rôznymi spôsobmi, v závislosti od anaforického výrazu môžeme hovoriť o viacerých druhoch, podľa [17] rozlišujeme:

- pronominálna anafora – anaforický výraz je zámeno
- koreferencia vlastného mena – Prezident ČR ... Miloš Zeman
- apozícia - Miloš Zeman, prezident ČR, ...
- ordinálna anafora, časť – celok anafora a mnohé ďalšie

Ako jednoduchý príklad pronominálnej anafory môže poslúžiť veta „*Petr* naléhal, aby *mu* řekli pravdu“, v ktorej funkciu anafory má zámeno *mu* a antecedentom je vlastné meno *Petr*. Za antecedenta najčastejšie považujeme frázu s podstatným menom, často však v roli antecedenta môže vystupovať slovesná fráza, veta alebo celý odstavec.
Existuje viacero prístupov k rezolúcii anafory. Väčšina z nich na začiatku identifikuje množinu kandidátov na antecedenta zo zvoleného kontextu. Následne sa na základe určitých faktorov (závislé na zvolenom prístupe, viď nižšie) niektorí kandidáti odstránia, a naopak niektorí uprednostia [34]. Literatúra poskytuje viacero klasifikácií pre konkrétne prístupy. Jedným z nich je delenie na základe množstva vedomostí o kontexte, ktoré sa využijú pri rezolúcii, na knowledge-rich a knowledge-poor [41]. Medzi knowledge-rich metódy je možné zaradiť syntaktické metódy, napríklad Hobbsov algoritmus [23], ktorý predstavuje jeden z najstarších algoritmov pre rezolúciu anafory (1977) a zároveň aj jeden z najúspešnejších. Sémantické metódy (metódy analýzy výpovede) je tiež možné považovať za knowledge-rich, ako napríklad metóda Centering Theory [34], ktorá sa pokúša nájsť v rámci výpovede „najnápadnejšieho“ kandidáta. Na druhej strane, medzi knowledge-poor prístupy patria najmä algoritmy na báze strojového učenia alebo algoritmy založené na povrchovej syntaktickej analýze a pod.
Existuje viacero systémov pre rezolúciu anafory v českých textov, spomeniem [37] vyvíjaný v ÚFAL a systém Saara [36] vyvíjaný v CZPJ.

3 Extrakcia informácií

Pod extrakciou informácií (d'alej IE, z anglického Information Extraction) budeme rozumieť automatickú extrakciu informácií z neštruktúrovaného (prípadne semi-štruktúrovaného) textu. Úlohou je transformovať túto neštruktúrovanú informáciu do štruktúrovanej podoby. Systém IE funguje najčastejšie nad určitou sémantickou doménou a extrahuje špecifickú informáciu, na rozdiel od komplexnejších systémov pre úplné porozumenie textom. V nasledujúcej kapitole bude najskôr zmienený úvod do problematiky a následne budú prebraté fázy IE, pričom sa budeme zameriavať na informácie relevantné pre návrh a implementáciu vlastného systému. Posledná podkapitola obsahuje stručný popis systému pre IE, ktorý je vyvíjaný v CZPJ.

3.1 Úvod do problematiky IE

Výzkum v oblasti IE bol v 80-tych rokoch výrazne ovplyvnený konferenciami pre Message Understanding (MUC). MUC sa konala každoročne a mala najmä kompetitívny charakter. Domény úloh MUC boli rôznorodé, od teroristických útokov v latinskej amerike po správy o akvizíciach obchodných spoločností a zmenách v ich vedení [12]. Podobne ani v súčasnosti sa vývoj v IE neobmedzuje na extrakciu informácií z jedného typu textov, systémy extrahujú informácie z novinových článkoch, webových stránok a blogových príspevkov, farmaceutických textov či vedeckých publikácií.
Adekvátne k rôznorodosti aplikácií IE sa postupne vyvíjali aj technológie a prístupy k IE. Najstaršie systémy boli pravidlové. Tie sa však pri narastajúcich požiadavkách stávali náročné na vytvorenie a údržbu, a tak vzniklo viacero algoritmov pre automatické učenie sa pravidiel. Z oblasti štatistického učenia sú populárne najmä skryté Markovové modely, modely maximálnej entropie a mnohé iné, pričom ich počet sa stále rozširuje. V praxi nie je možné označiť jeden prístup za najlepší, jednotlivé postupy sa často kombinujú a vznikajú rôzne hybridné modely [44].
Pri úlohach IE je potrebné, ako u väčšiny aplikácií spracovania prirodzeného jazyka, pracovať so vstupným textom na rôznych úrovniach, ktoré boli stručne opísané v druhej kapitole. Okrem týchto procesov je možné úlohu IE rozdeliť na dve podstatné časti:

- extrakcia menných entít

- extrakcia vzťahov

Jedným z trendov pri IE je vynechávanie niektorých krokov pri analýze vstupného textu. Konkrétne v [44][19] polemizujú autori o vplyve syntaktickej analýzy na úlohu IE. Výpočtová náročnosť a častokrát nízka presnosť analýzy sú dva dôvody, ktoré autori uvádzajú, a pre ktoré je podľa nich vhodnejšie vykonať povrchovú analýzu (shallow parsing), resp. identifikáciu iba menných a slovesných fráz, alebo v niektorých prípadoch syntaktickú analýzu vynechať úplne.
Pri evaluácií výsledkov IE (ako pri extrakcii menných entít, tak aj pri extrakcii vzťahov) sa najčastejšie využívajú metriky precision, recall a F-skóre.

$$Precision = \frac{correct}{correct + spurious}$$

$$Recall = \frac{correct}{correct + missing}$$

$$F = \frac{2 * precision * recall}{precision + recall}$$

Pod premennou *correct* rozumieme počet správne identifikovaných entít, resp. vzťahov, pod *spurious* počet nesprávne identifikovaných a pod *missing* počet chýbajúcich (neidentifikovaných) entít [18].

3.2 Extrakcia menných entít

Extrakcia menných entít (ďalej NER, z anglického Named Entity Recognition) je základnou fázou IE. Pod NER rozumieme identifikáciu menných skupín v texte, ako vlastné mená, skratky, dátumy alebo monetárne jednotky. Majme napríklad vetu *Orange Polska po výsledcích roste o 0,5 PLN*. Úlohou NER je v danej vete identifikovať slovné spojenie *Orange Polska* ako vlastné meno (prípadne mu priradiť inú značku, v závislosti od úlohy a systému ho môžeme označiť ako obchodnú spoločnosť, organizáciu, a pod.) a spojenie *0,5 PLN* ako napr. numerickú hodnotu alebo cenu. Voľba entít, ktoré z textu extrahujeme, a to, ako ich označíme, závisí od daného systému. Systém IE v oblasti medicínskych textov väčšinou extrahuje inú informáciu, ako systém pracujúci nad doménou publicistických článkov.
Systémy NER môžeme vo všeobecnosti rozdeliť na pravidlové a štatistické. Pravidlové systémy sú starším prístupom, no stále sú predmetom skúmania a v praxi dosahujú dobré výsledky. Pokiaľ extrahujeme z textu entity

ako sú emailové adresy alebo telefónne čísla, vytvorenie pravidiel je efektívnym riešením. Klasický pravidlový systém sa skladá z dvoch častí, súboru vlastných pravidiel a kontroly prípadu, kedy danému tokenu vyhovuje viac pravidiel [44]. Spôsoby reprezentácie pravidiel sa líšia, od regulárnych výrazov, napr. pre numerické hodnoty, po pravidlá berúce do úvahy zvolené vlastnosti klasifikovaných tokenov. Pri voľbe vlastností môžeme brať do úvahy textovú reprezentáciu (veľké písmená, prítomnosť špecifických písmen, slabík, číslic, dĺžku tokenu,...), lingvistické vlastnosti (značky získané počas morfologickej analýzy, závislostný strom,...) alebo tiež kontext. Pravidlá môžu byť aplikované na celé vety, frázy či slová.
Pri štatistických metódach hovoríme najčastejšie o tokenovom modele [44]. V texte sa tieto metódy pokúšajú priradiť tokenom značky podľa toho, akú mennú entitu reprezentujú. Ako pri každej klasifikačnej úlohe výber charakteristík (feature extraction) je kľúčový. Mnohé metódy priraďujú značku vybranému tokenu nezávisle od značiek v kontexte, ktoré sú priradené okolitým tokenom [44]. Na druhej strane sekvenčné modely uvažujú závislosť medzi značkami okolitých tokenov využívajúc pohyblivé okno. Ako príklad týchto metód uvediem skryté Markovové modely (HMM) alebo Markovové modely maximálnej entropie (MEMM) [27].
Príkladom systému NER pre český jazyk je nástroj dostupný na adrese [7], ktorý je v súčasnosti vyvíjaný na CZPJ (momentálne neexistujú žiadne oficiálne zdroje, ktoré by bolo možné citovať). V texte identifikuje rôzne entity, od číšiel a skratiek po napr. odkazy na zákony.
Veľmi úspešným systémom je [47], ktorý je vyvíjaný na ÚFAL. Na extrakciu entít využíva Markovové modely maximálnej entropie, pričom k typickým klasifikačným črtám (tvar tokenu, lemma, tagy v okolí tokenu, regulárne výrazy,...) autori pridávajú tzv. globálne charakteristiky, založené napr. na dvojfázovej klasifikácii alebo dátach z českej wikipédie. Výsledky pre češtinu dosahujú F-skóre 79,23. Ich systém funguje aj nad doménou anglických textov, pri ktorých je F-skóre na úrovni 89,16.
Úloha NER je pomerne úspešne riešiteľná a pri všeobecných typoch menných entít aj veľmi dobre prenositeľná medzi rôznymi doménami aplikácií.

3.3 Extrakcia vzťahov

V mnohých aplikáciách samotná extrakcia menných entít nie je postačujúca a úloha si vyžaduje identifikáciu vzťahov medzi nimi. V tomto prípade hovoríme o extrakcii vzťahov (z anglického Relation Extraction). Najčastejšie

uvažujeme binárne vzťahy, tzv. vzťahy medzi dvomi entitami, ale úlohu je možné ľubovoľne rozšíriť.
Úlohu extrakcie vzťahov môžeme ilustrovať na príklade z predchádzajúcej kapitoly. V procese NER sme zistili, že *Orange Polska* je obchodná spoločnosť a *0,5 PLN* numerická hodnota. Táto informácia nám však nič nehovorí o tom, v akom vzťahu sú jedna k druhej. Vzťah, ktorý chceme získať, môžeme zapísať ako predikát *rásť(Orange Polska; 0,5PLN)*. Pod vzťahom môžeme teda rozumieť predikát dvoch argumentov, entít [18].
V predchádzajúcom príklade máme identifikované dve entity a pokúšame sa nájsť vzťah medzi nimi. Medzi časté úlohy extrakcie vzťahov patria ešte dva prípady. V prvom máme danú jednu mennú entitu e a typ vzťahu v, pričom úlohou je nájsť všetky entity, ktoré sú s entitou e vo vzťahu v. V druhom prípade je daný vzťah v a úlohou je nájsť v texte všetky páry entít, ktoré sa v danom vzťahu v nachádzajú [44].
Podobne ako pri úlohe NER, aj prístupy k extrakcii vzťahov je možné rozdeliť na pravidlové a štatistické. Tvorba pravidiel je vhodná pre špecifické domény s nižším počtom vzťahov, pri komplexnejších systémoch s vyšším počtom vzťahov sú pravidlové systémy náročné na tvorbu a údržbu. Pri štatistický systémoch s učiteľom je úloha extrakcie vzťahu často rozdelená do dvoch fáz [27] [18]. V prvej fáze klasifikátor pre každé dve entity v rámci napr. jednej vety rozhodne, či je medzi nimi vzťah alebo nie. V druhej fáze sa pokúša následne určiť typ tohto vzťahu. Ako črty pri klasifikácií je možné využiť napríklad typy daných entít, vzdialenosť v texte medzi nimi, sekvencie slov medzi nimi, závislostné stromy a pod. Špeciálnym prípadom metód s učiteľom sú kernelové metódy [18], ktoré sa za pomoci kernelovej funkcie (v tomto kontexte určitá miera podobnosti) snažia určiť podobnosť medzi označkovaným reprezentantom vzťahu a inštanciou, ktorá je klasifikovaná. Najčastejšie ide o podobnosť závislostných stromov alebo sekvencií slov. Algoritmy ako SVM alebo algoritmus najbližšieho suseda môžu priamo pracovať s kernelovými funkciami, a preto sú vhodné pre tento prístup.
Nevýhodou metód s učiteľom je potreba korpusu s označkovanými dátami. Medzi alternatívy patrí napr. bootstraping, ktorý využíva na začiatku iba niekoľko označkovaných príkladov. Algoritmus následne označkované dvojice vyhľadáva v korpuse a generuje pravidlá pre extrakciu vzťahov na základe nového kontextu, v ktorom boli entity nájdené. Na tomto princípe fungujú algoritmy ako DIPRE alebo Snowball [12]. Podobným prístupom je metóda Distance Supervision, ktorá podobne využíva na začiatku označkovaný príklad dvoch entít a vzťahu medzi nimi. Následne v texte vyhľadá spoločné výskyty daných entít a ich kontext, na rozdiel od DIPRE alebo

Snowball metód, využije na extrakciu čŕt a ako pozitívne trénovacie príklady [18].
Medzi úlohy extrakcie informácií často patrí aj rezolúcia koreferencie alebo anafory, ku ktorým sme sa stručne vyjadrili v kapitole 2.5.
Aj napriek neustálemu výskumu nie je možné považovať extrakciu vzťahov za vyriešený problém, najúspešnejšie systémy dosahujú F-skóre 75 [18].

3.4 Extrakcia informácií pre češtinu

Príkladom IE systému pre češtinu je práca [14] Víta Baisu a Vojtěcha Kováře z CZPJ. Autori sa nezamerali na extrakciu informácií z jednej špecifickej domény, ale vytvorili systém pre extrakciu faktov a všeobecných informácií bez zamerania na určitý typ textov.
V publikácií autori opisujú postupnosť procesov, ktorými je potrebné spracovať vstupný text na rôznych jazykových rovinách, pričom využívajú nástroje CZPJ. Podstatnou časťou je klasifikácia slovesných, menných a predložkových fráz. Systém sa každej fráze pokúša priradiť jednu z predpripravených sémantických značiek (napr. subjekt, miesto, čas, spôsob,...). Pri priraďovaní značiek sa systém rozhoduje na základe vytvorených pravidiel s využitím dát českého WordNetu [40]. Systém dosahuje presnosť (F-skóre) 69,9%. Webové rozhranie je prístupné na adrese [2].

Ministr vnitra včera v Praze schválil dlouho očekávaný zákon				
SUBJECT	VP	WHEN	DIROBJ	WHERE
Ministr vnitra	schválil	včera	dlouho očekávaný zákon	v Praze

Tabuľka 3.1: Príklad extrakcie faktov

4 Analýza a návrh systému

V nasledujúcej kapitole sa venujeme analýze a návrhu vlastného systému. Uvedieme charakteristiku vstupných textov a popis nárokov, ktoré kladú tieto texty na systém. V popise návrhu systému budeme postupovať na základe rovín spracovania jazyka spomínaných v kapitole 2.1. Pri každej rovine budú uvedené problémy, na ktoré sme narazili, nedostatky použitých nástrojov a to, ako bolo potrebné tieto nedostatky riešiť. V kapitole tiež rozoberieme ako bola implementovaná úloha extrakcie informácií.

4.1 Charakteristika vstupného textu a úlohy

V našej práci sme sa zamerali na doménu z oblasti ekonomických textov, konkrétne na burzové správy.
Dianie na svetových či lokálnych finančných trhoch sleduje množstvo informačných agentúr. V závislosti od konkrétnej agentúry sa líši aj povaha článkov, ktoré sú uverejňované. Jednotlivé texty môžu obsahovať stručné informácie o udalostiach na trhu, raste či prepade akcií, ale tiež obšírnejšie analýzy alebo rozbory. Ako vstupné dáta pre systém sme použili správy uverejňované na portáli Fio banky [5].
Správa ako textový útvar patrí medzi útvary publicistického štýlu. Jej hlavnou funkciou je informovať, a tomu sa podriaďujú prostriedky, ktoré využíva. Charakteristiky sa líšia v závislosti od oblasti, z ktorej správa pochádza. Pre správy z burzy je typické stručné a vecné vyjadrovanie, bohaté používanie skratiek (skratky titulov na burze, skratky mien,...), názvov organizácií či doménovo špecifických výrazov (napr. názvy doporučení). Časté je tiež využívanie bezslovesných viet, napr. *BMW – cílová cena 100 euro a snížené doporučení market-perform*. Konkrétne príklady a ich rozbor budú uvedené v nasledujúcich podkapitolách.
Ako bolo spomenuté v úvode, cieľom práce je vytvoriť systém pre extrakciu informácie o zmene doporučenia pre trhový titul spomínaný v texte správy. Finančné doporučenie predstavuje názor analytika na to, či je vhodné akcie daného subjektu na trhu kupovať alebo nie [4]. V praxi ide najmä o analytikov investičných skupín, ktoré vystupujú na danej burze, pričom každá skupina využíva vlastnú terminológiu. Pre klienta to znamená náročnejšie porovnávanie doporučení medzi viacerými analytikmi, pre vývojára systému to predstavuje nejednotnú doménu s dátami, ktoré nie je vždy možné získať. Ako príklady názvov spomenieme *buy, sell, hold, outperform, equalweight* alebo na portáli [5] často používané české preklady *koupit, pro-*

dat, podvážit atď.
Pri extrakcii informácií nás zaujímajú odpovede na nasledujúce otázky:

- Komu bolo zmenené doporučenie?
- Kto zmenil toto doporučenie?
- Aká je hodnota nového doporučenia? Aké bolo pôvodné doporučenie?
- Nachádza sa v texte informácia o reakcii ceny na dané doporučenie?

Aké môžu byť hľadané informácie ilustrujeme na nasledujúcom príklade:

Barclays snižuje doporučení pro Deutsche Bank na „equalweight“ z „overweight“. Akcie (DBK) včera připsaly 0,59 % na 36,92 eur.

Ide o krátky a vecný text, ktorý bol skutočne uverejnení na spravodajskom webovom portáli [5]. Text obsahuje informácie o zmene doporučenia pre jeden trhový titul jednou agentúrou. Následne je v texte uvedená informácia o aktuálnej trhovej cene paciensa zmeny doporučenia. Konkrétne odpovede vyzerajú teda takto:

- Deutsche Bank (DBK)
- Barclays
- doporučenie equalweight, z pôvodného overweight
- nárast ceny o 0,59 % na 36,92 eur

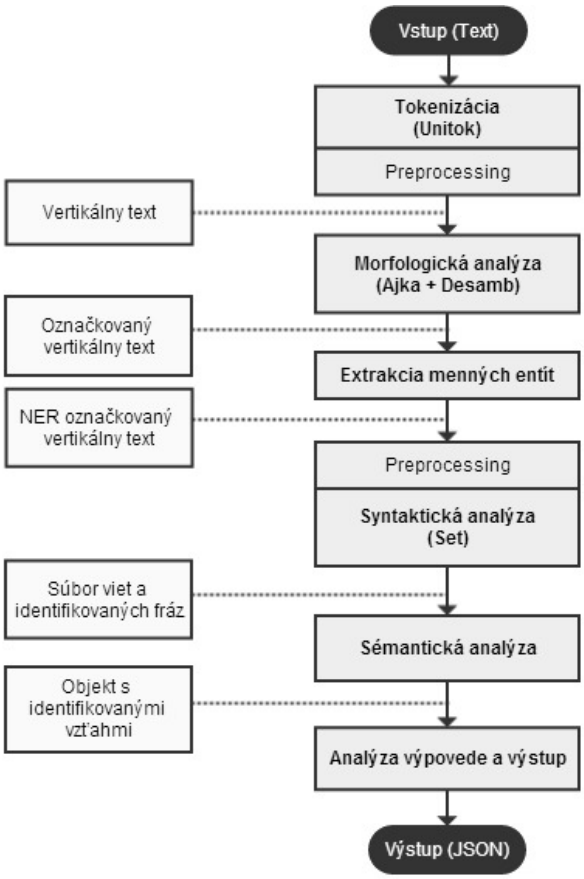
Výstupným objektom systému je záznam obsahujúci dané informácie. V tomto príklade je možné si tiež všimnúť, že hľadaný aktér je v texte označený dvomi rôznymi menami. Pre ilustráciu možného vyjadrenia zmeny doporučenia uvádzame niekoľko príkladov:

- Raymond James snížilo doporučení pro stavitele Lennar na outperform ze strong buy.
- Société Générale – cílová cena 575 USD s doporučením "Držet".
- Credit Suisse začalo pokrývat společnost Red Hat s doporučením „outperform".
- Poslední z jmenovaných posiluje po zvýšeném nákupním doporučení od Raiffeisen Bank.

Zaujímavým príkladom z viacerých dôvodov je druhá veta. Okrem toho, že sa jedná o bezslovesnú vetnú konštrukciu, je možné si všimnúť tiež nejednoznačnosť sémantickej role entity *Société Générale*. Pri izolovanom pohľade iba na danú vetu nevieme určiť, či *Société Générale* mení doporučenie, alebo či je jej doporučenie menené. Až na základe vedomosti o kontexte a vedomostí o svete, je možné určiť, že daná entita má rolu analytickej agentúry, ktorá doporučenie mení. Ďalším dôvodom je vo vete zachytený vzťah agentúry a ceny. Agentúra často okrem doporučenia uverejňuje aj odporúčanú kúpnu alebo predajnú cenu, pokiaľ je táto informácia v texte prítomná, tak sa ju systém snaží extrahovať tiež.
Na tomto mieste považujeme za dôležité podotknúť, že úlohou systému je extrahovať konkrétnu hodnotu doporučenia, prípadne ceny. Pomerne bežným typom vety je *Deutsche Bank klesá po sníženém doporučení*. Prídavné meno *snížený* nie je konkrétnym označením doporučenia a systém sa túto informáciu nesnaží extrahovať. Daná veta však môže byť dôležitá z hľadiska indikácie ďalšieho obsahu, no najmä je na jej základe možné určiť, že *Deutsche Bank* bude vystupovať v texte ako paciens zmeny doporučenia.
Na záver uvedieme krátku poznámku k formátu správ. Burzová správa, ktorá je zverejnená na webovom portáli [5], sa skladá z troch častí: titulok, perex (podtitulok) a samotný text správy. Pre extrakciu informácií sú použité všetky tri časti spojené do jedného celku, pričom perex alebo samotný text nie sú niekedy v správe vôbec prítomné. Ku každej časti tak pristupujeme rovnocenne bez prikladania dôrazu na informáciu obsiahnutú napr. v titulku. Nevylučujeme, že by prístup, ktorý by kládol rozdielny dôraz na každú časť správy, nemohol byť úspešný, avšak zároveň by bol pravdepodobne komplexnejší.

4.2 Návrh systému

V kapitole 2.1 sme rozprávali o úrovniach, na ktorých je možné jazyk spracovávať. Systémy pre spracovanie prirodzeného jazyka najčastejšie preto pozostávajú zo série procesov, v ktorých výstup jedného procesu je vstupom pre nasledujúci. Každý proces vykonáva analýzu na určitej úrovni, pričom pri vývoji samostatnej aplikácie je kvôli náročnosti úlohy na danej úrovni vhodné využiť externé nástroje. Príkladom pre anglický jazyk sú anotátory v Stanford CoreNLP knižnici [10]. Pre potreby českého jazyka sú však podobné knižnice nevyhovujúce, a preto je potrebné využiť nástroje špeciálne vyvinuté pre češtinu.
Na obrázku 4.1 je možné vidieť jednoduchý diagram systému, na ktorom



Obr. 4.1: Diagram navrhnutého systému

sú znázornené niektoré roviny jazyka a zároveň tiež procesy, ktoré bolo nutné pridať k jednotlivým fázam. V diagrame sú okrem týchto základných úloh spracovania jazyka vyznačené aj procesy, ktoré priamo patria k extrakcii informácií. Vyznačené v zátvorkách sú externé nástroje, ktoré boli použité na podstatné úlohy v rámci analýzy na danej úrovni.
Ako programovací jazyk pre implementáciu sme zvolili jazyk Python [8]. V Pythone je možné programovať na základe viacerých programovacích paradigiem, je vhodný ako skriptovací jazyk a tiež ako jazyk pre implementáciu komplexnejšich systémov. V porovnaní s jazykmi ako C, C++ a Java je síce pomalší, ale na druhej strane v kontexte akademickej činnosti je cenená lepšia čitateľnosť, univerzálnosť, rýchlejší vývoj aplikácií alebo dostupnosť knižníc.

4.3 Spracovanie vstupu

Vstupom systému je neštruktúrovaný text, intuitívne chápaný ako súbor slov alebo viet, avšak na elementárnej úrovni ide o súbor znakov v určitom kódovaní. Cieľom analýzy je podať na vstup pre morfologickú analýzu organizovaný (podľa poradia vo vete, v texte) súbor slov a viet vo vertikálnom formáte, tzv. každé slovo na osobitnom riadku. Pri tom musíme uvažovať nasledujúce úlohy:

- jazykovo kompatibilné kódovanie znakov
- tokenizácia na úrovni slov
- tokenizácia na úrovni viet

Pre reprezentáciu textu v počítači je možné využiť jedno z mnohých kódovaní. Pre potreby českého jazyka a aplikácií, s ktorými sme pracovali je vhodné využiť nasledujúce sady znakov:

- ISO/IEC 8859–2 (Latin–2)
- UTF–8

Sada znakov Latin–2 je vo všeobecnosti určená pre jazyky východnej európy a obsahuje v sebe aj znaky českej abecedy. Kódovanie UTF–8 obsahuje všetky znaky sady Unicode, ide o veľmi často používané a obľúbené kódovanie, ktoré je aj využité vo väčšine aplikácií a utilít, ktoré boli pri implementácii použité. Kedže nástroje ako sú Ajka alebo Desamb využívajú kódovanie Latin–2, bolo potrebné niekoľkokrát text konvertovať z jedného formátu na druhý.

Tokenizácia na úrovni slov je proces segmentácie textového vstupu na jednotlivé slová alebo tokeny (bodka, čiarka, otáznik a pod.). Pod tokenizáciou na úrovni viet budeme rozumieť na druhej strane segmentáciu textu na jednotlivé vety. Najmä segmentácia textu na vety nie je úplne triviálny problém. Bodka je častokrát používaná nielen ako oddeľovač viet, ale aj v rámci skratiek (J. Novák, tis., atď.).

Pre tokenizáciu vstupu bol použitý program Unitok [43], vytvorený Janom Pomikálkom. Program slúži najmä na segmentáciu na úrovni slov, pričom do výstupu vkladá tagy, ktoré označujú možný koniec vety. Utilita je inšpirovaná tokenizérom TreeTagger Wrapper [42] vyvinutým Laurent Pointalom. Na segmentáciu slov využiva regulárne výrazy, ktoré popisujú rôzne

entity, ktoré sa môžu v texte vyskytovať, od IP adries po rôzne číselné výrazy. Unitok je veľmi užitočný nástroj, avšak nutné je ďalšie spracovanie výstupu.
Pre potreby ďalšieho spracovania bolo potrebné upraviť výstup programu Unitok nasledujúco. Pre jednoduchšiu morfologickú analýzu je vhodnejší token v tvare *WIG20*, pričom Unitok daný token rozdelí na dva: *WIG* a *20*. Unitok rozdeľuje tiež napríklad numerické hodnoty, ako sú cieľové ceny. Podobné prípady bolo nutné ošetriť a niektoré tokeny, ktoré boli Unitokom rozdelené, opäť spojiť.
Pre identifikáciu hraníc viet bolo potrebné vytvoriť vlastný modul. O konci vety sa v ňom rozhoduje na základe jednoduchých pravidiel, v ktorých sa uvažujú niektoré bežné bodkou ukončené skratky vyskytujúce sa v textoch správ a tokeny, ktoré za bodkou nasledujú.

4.4 Problémy morfologickej analýzy

V kapitole 2.2 sme sa venovali nástrojom pre analýzu češtiny na morfologickej úrovni. Tiež bolo uvedené, že analýza pozostáva z dvoch fáz: morfologického značkovania a desambiguácie. Ako súčasť systému boli využité na tieto úlohy analyzátor Ajka a desambiguátor Desamb. Vstup pre morfologickú analýzu je forme vertikálneho textu, tzv. text je vo forme zoznamu, kde každé slovo je na osobitnom riadku a v texte sú vyznačené hranice viet. Napriek komplexnej morfológií a homonymii na morfologickej úrovni, kvalita výstupu dvojice Ajka a Desamb je na praktickej úrovni veľmi dobrá. Výraznejšie problémy nastávajú pri textoch z rôznych domén s vlastnými špecifikami, ako napr. pri textoch burzových správ. Tieto problémy je možné si všimnúť pri analýze vety *Operátor Orange Polska roste poté, co Erste Group zvyšuje cílovou cenu akcie na 10,7 PLN z původních 8 PLN a ponechává doporučení držet*. Podobná veta pochádza z korpusu textov, ktoré boli použité pri analýze a testovaní, pre účely ilustrácie problematiky morfologickej analýzy bola však mierne skrátená.
Na obrázku 4.2 je zobrazený štandardný výstup programu Desamb, tzv. označkovaný a desambiguovaný vertikálny text, ktorý ilustruje viacero problémov, ktoré môžu nastať pri podobných vetách. Jedným z problémov je značkovanie desatinných čísiel (v tomto príklade sú číslice oddelené desatinnou čiarkou), kedy analyzátor nevie priradiť vhodnú značku. Pre ďalší postup je však najkritickejšie značkovanie menných entít alebo skratiek. Zaujímavým príkladom je skratka PLN pre poľskú monetárnu jednotku,

```
<s desamb="1">
Operátor        operátor        k1gInSc1
Orange  Orange  k1gNnSc2
Polska  Polsko  k1gNnSc2
roste   růst    k5eAaImIp3nS
poté    poté    k6eAd1
,       ,       kIx,
co      co      k3yQnSc4,k3yRnSc4,k3yInSc4
Erste   Erst    k1gMnSc5
Group   Group   k1gInSc1
zvyšuje zvyšovat        k5eAaImIp3nS
cílovou cílový  k2eAgFnSc4d1
cenu    cena    k1gFnSc4
akcie   akcie   k1gFnSc2
na      na      k7c4
10,7    10,7    k?
PLN     pln     k2eAgInSc1d1
z       z       k7c2
původních       původní k2eAgNnPc2d1
8       #num#   k4
PLN     plno    k1gNnPc2
a       a       k8xC
ponechává       ponechávat      k5eAaImIp3nS
doporučení      doporučení      k1gNnSc4
držet   držet   k5eAaImF
.       .       kIx.
</s>
```

Obr. 4.2: Štandardný výstup programu Desamb

ktorú analyzátor identifikoval ako mužské neživotné prídavné meno *pln* v nominatíve, resp. mužské neživotné podstatné meno *plno* v genitíve. Ďalším príkladom je napr. analýza názvu *Erste Group*. Vo všeobecnosti problém nastáva ak niektoré vlastné meno alebo skratka majú formu určitého českého slova a analyzátor potom danému vlastnému menu alebo skratke priradí nesprávnu značku. Tieto problémy budú adresované v d'alších kapitolách pri identifikácií menných entít a úprave vstupu pre syntaktickú analýzu.

4.5 Extrakcia menných entít

V kapitole 3.2 sme sa venovali špecifickej fáze úlohy extrakcie informácií z textu – extrakcii menných entít (NER). Fáza NER je nevyhnutná aj v našom systéme pre d'alšiu fázu extrakcie konkrétnych vzťahov medzi nimi. Navyše, takto identifikované entity bude možné použiť pre úpravu výstupu

morfologickej analýzy, a tak odstrániť problémy spomenuté v predchádzajúcej kapitole.
Systém rozlišuje tri druhy menných entít:

- identifikátor trhového aktéra, napr. Twitter, TWTR,...
- cena, napr. 0,5 USD; +1,3 %,...
- názov doporučenia, napr. kupovat, sector perform,...

Tvar a typ entít (najmä doporučení) je často odlišný od entít, ktoré extrahujú bežné NER nástroje, a to je hlavný dôvod, prečo sme sa rozhodli pre vlastnú implementáciu NER.
Úlohou modulu pre NER je ich identifikácia v texte a priradenie NER značiek (konkrétne ACTOR, PRICE, STATE). Proces pozostáva z troch fáz:

1. úprava vstupu
2. vlastná NER
3. úprava výstupu

Vstupom pre modul je označkovaný vertikálny text. Na základe pozorovaní textov správ je vhodné spojiť niektoré skupiny tokenov. Prvou z nich je obsah zátvoriek, ktorý najčastejšie reprezentuje samotnú mennú entitu reprezentovanú trhovou skratkou. Ak by sa v texte vyskytovala skupina *(OPL +2,23%)*, tokenizátor by ju rozdelil na 5 samostatných tokenov („*(*“, „*OPL*“, „*+2,23*“, „*%*“, „*)*“). V tomto tvare by však komplikovala úlohu NER a najmä syntaktickej analýzy, a preto je vhodné túto skupinu reprezentovať jedným tokenom. Modul tiež spája obsah úvodzoviek. Vo väčšine textov úvodzovky vymedzujú priamu reč, avšak v burzových správach úvodzovky najčastejšie ohraničujú názov doporučenia. Ich spájanie má preto zmysel najmä pri viacslovných názvoch doporučení (market perform, strong buy,...).
Počas vlastnej NER sú na identifikáciu menných entít použité vlastné vytvorené pravidlá. Úloha je najjednoduchšia v prípade identifikácie cien v texte, ktoré sú rozoznávané pomocou regulárneho výrazu a tokenov v bezprostrednom kontexte daného číselného výrazu. Napríklad, za cenu často nie je možné považovať celočíselny výraz, ktorému predchádza názov kalendárneho mesiaca.
Komplikovanejšia je identifikácia doporučení a názvov trhových aktérov. Pre rozpoznávanie názvov trhových aktérov bolo vytvorených viacero pravidiel, ktoré sú založené na tvare názvov. Konkrétne, pokiaľ sa vyskytuje

token s veľkým začiatočným písmenom uprostred vety, je možné predpokladať, že ide o názov. Podobne je to s tokenmi, ktoré sa skladajú iba z veľkých písmen, zátvoriek a čísiel (napr. *(TWTR)*), majú počiatočné písmeno malé, a za ním nasleduje veľké písmeno (napr. *mBank, eBay*) a podobne. Tieto pravidlá však nie sú vždy aplikovateľné, napr. každá veta začína tokenom s veľkým začiatočným písmenom alebo nie každá entita pozostávajúca iba z veľkých písmen označuje trhového aktéra (napr. *USD*). Za účelom odstránenia týchto problémov boli vytvorené, na základe automaticky získaného korpusu textov správ, dva zoznamy. Prvý obsahuje názvy aktérov, ktoré sa často vyskytujú v textoch správ, a druhý skratky, ktoré nie je možné považovať za trhových aktérov. Kombinácia zoznamov a pravidiel sa ukázala ako pomerne dobré riešenie.
Podobný prístup bol použitý pri extrakcii doporučení. Vytvorenie zoznamov s názvami doporučení je oproti zoznamom mien trhových aktérov menej náročné, názvy všetkých možných označení doporučení nie je síce možné získať, ale množina v texte najčastejšie sa vyskytujúcich doporučení je pomerne obmedzená. Zoznamy sú preto hlavným prostriedkom pre identifikáciu názvov doporučení v texte. Doplnené sú menším počtom pravidiel pre identifikáciu niektorých výnimiek.
V takto označkovanom texte NER značkami sú ešte počas úpravy výstupu spojené niektoré tokeny. Konkrétne, token označený ako cena je spojený s nasledujúcim tokenom, pokiaľ ide o identifikátor meny alebo napríklad znak „%“. Tiež sú spojené všetky za sebou idúce tokeny, ktoré sú označené tou istou NER značkou. To sa týka najmä tokenom so značkou ACTOR. V príklade z obrázka 4.2 tak vzniknú spojené tokeny *Orange_Polska* a *Erste_Group*. Spájanie má však tiež za následok, že program nie je schopný rozlíšiť prípad, kedy za sebou nasledujú dva rôzne názvy. Pri analýze vety *Po zvýšenom doporučení od Barclays Twitter posiluje* tak bude vytvorená jedna menná entita *Barclays_Twitter*. Tento jav je však veľmi vzácny (predchádzajúci príklad nepochádza z textov správ) a tento postup je často využívaný pri značkovaní menných entít v korpusoch, resp. pri samotnej NER [27].
Príklad označkovaného textu je možné nájsť na obrázku 4.3, ktorý už obsahuje aj informáciu zo syntaktického preprocessingu.

4.6 Problémy syntaktickej analýzy

Na syntaktickej úrovni sa zaoberáme spájaním slov do fráz a fráz do viet. Správna identifikácia fráz a určenie závislosti medzi nimi je nevyhnutným predpokladom pre úspešnú sémantickú analýzu. Ako syntaktický analyzá-

tor bol použitý nástroj SET (popísaný v kapitole 2.3.1). Vstupom pre program SET je vertikálny text s priradenými morfologickými, a v našom prípade aj NER, značkami. Výstup, ktorého formát je v prípade programu SET možné nastaviť pomocou prepínačov, je vo forme zoznamu fráz s vyznačenými závislosťami medzi nimi.
V predchádzajúcich dvoch kapitolách boli spomenuté problémy, ktoré nastávajú pri morfologickom značkovaní textov burzových správ. Tento problém sa prenáša aj ďalej na syntaktickú analýzu a program SET tak nedokáže správne identifikovať frázy vo vete. Potrebné je preto vo fáze preprocessingu upraviť morfologické značky niektorých tokenov, konkrétne tokenov, ktoré reprezentujú aktérov, doporučenia, ceny, ale tiež ostatné skratky. Najpodstatnejšou gramatickou kategóriou pre SET, či už sa to týka zaradenia slov do rovnakej frázy alebo identifikácie subjektu a objektu, je kategória pádu (vychádzame z vlastných pozorovaní správania sa nástroja). Cieľom spracovania vstupu je preto identifikácia správneho pádu pre spomínané entity (samozrejme, vytvorená je celá značka aj s ostatnými kategóriami, aby bol vstup pre SET validný). Správny pád sa určuje na základe pravidiel, ktoré berú do úvahy kontext daného tokenu a jeho prípadnú NER značku. Určiť pád po predložke alebo prídavnom mene nie je komplikované, úloha sa však stáva náročnou, ak ani jeden tento slovný druh nie je v kontexte slova, ktorému určujeme pád. Ak sa v kontexte vyskytuje sloveso, je potrebné prejsť okolité tokeny a určiť, či danej slovesnej valencii chýba subjekt alebo objekt, pokiaľ tokenu predchádza spojka alebo čiarka, pád závisí od toho, či je spojka podraďovacia alebo priraďovacia a podobne.
Na obrázku 4.3 je zobrazený upravený vertikálny text, ktorý bude vstupom pre program Set. Je možné si všimnúť druhý pád pri entite *Orange Polska* a pri doporučení *držet*. Program SET tak prinútime spojiť túto entitu do jednej frázy s predchádzajúcim podstatným menom.

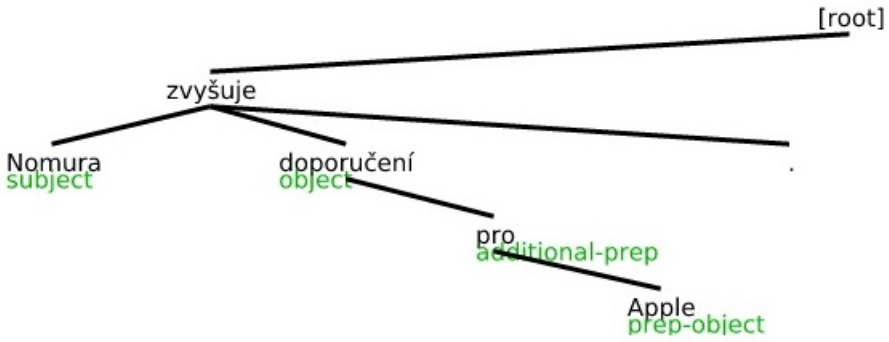
4.7 Sémantická analýza

Hlavným cieľom sémantickej analýzy je vytvorenie formálnej sémantickej reprezentácie pre danú vetu. V našom prípade je touto reprezentáciou vzťah určený predikátom a rolami, ktoré sú obsadené vetnými frázami. V systéme túto funkciu vykonáva modul pre identifikáciu slovesných a menných rámcov. Hlavnou funkciou analýzy na sémantickej úrovni v kontexte nášho systému je identifikácia vzťahov medzi mennými entitami a špecifikácia sémantických rolí jednotlivých entít v týchto vzťahoch. Napríklad,

```
<s desamb=" 1 ">
Operátor        operátor        k1gInSc1
Orange_Polska_ACTOR     Orange_Polska   k1nPgIc1
roste   růst    k5eAaImIp3nS
poté    poté    k6eAd1
,       ,       kIx,
co      co      k3yQnSc4,k3yRnSc4,k3yInSc4
Erste_Group_ACTOR       Erste_Group     k1nPgIc1
zvyšuje zvyšovat        k5eAaImIp3nS
cílovou cílový  k2eAgFnSc4d1
cenu    cena    k1gFnSc4
akcie   akcie   k1gFnSc2
na      na      k7c4
10,7_PLN_PRICE  10,7_PLN        k1nPgIc4
z       z       k7c2
původních       původní k2eAgNnPc2d1
8_PLN_PRICE     8_PLN   k1nPgIc2
a       a       k8xC
ponechává       ponechávat      k5eAaImIp3nS
doporučení      doporučení      k1gNnSc4
držet_STATE     držet   k1nPgIc2
.       .       kIx.
</s>
```

Obr. 4.3: Upravený výstup programu Desamb

jednoduchú vetu *Nomura zvyšuje doporučení* je možné zapísať ako predikát *zvyšovať(Nomura, doporučení)*. Tento predikát tak identifikuje rolu agensa a objektu, a zároveň špecifikuje úlohu aktéra *Nomura*, ktorý v tomto vzťahu vystupuje ako entita „určujúca“ doporučenie inému trhovému aktérovi. Konkrétny popis sémantických rolí sa nachádza v kapitole 4.7.1.
V úvode by sme chceli uviesť dôvody, prečo je nutné vytvoriť rámce aj pre menné skupiny. Veľmi dôležitým dôvodom je súvislosť identifikácie slovesných rámcov a závislostného stromu. Na obrázku 4.4 je zobrazený závislostný strom pre vetu *Nomura zvyšuje doporučení pro Apple*. Subjekt *Nomura* a objekt *doporučení* sú závislé na slovese *zvyšuje*, avšak fráza *pro Apple* je komplementom podstatného mena *doporučení*. Slovesné valenčné rámce identifikujú objekty závislé na slovese, preto bolo potrebné tento mechanizmus rozšíriť. Podobná situácia nastáva aj pri identifikácií rámcov pri pasívnom tvare slovies. V tomto prípade objekt obsadzuje prvý slot v rámci a subjekt často nie je ani prítomný. Zároveň sa tiež mení závislosť niektorých fráz na objekte, ktoré sa stávajú závislými na slovese (vychádzame z pozorovaní výstupu programu SET).
Ďalším dôvodom je prítomnosť bezslovesných viet, ktoré sú častým javom



Obr. 4.4: Závislostný strom vygenerovaný programom SET

v textoch burzových správ. Je evidentné, že pre dané vety nie je možné aplikovať slovesné rámce. Tieto vety však často obsahujú informáciu, ktorú chceme z textu získať, a preto je potrebné na ne uplatniť mechanizmus rozširujúci slovesné rámce.

4.7.1 Valenčné rámce slovies

Pri návrhu slovesných rámcov sme vychádzali najmä zo slovníka VerbaLex (viď kapitola 2.4.2). Na konkrétnych príkladoch je možné si všimnúť určité podobnosti. Avšak valenčné rámce v navrhnutom systéme boli použité so špecifickým zámerom extrakcie informácie, preto obsahujú viacero odlišností a tiež zjednodušení oproti rámcom slovníka VerbaLex. V zozname slovies sa nachádzajú iba slovesá a ich valenčné rámce, ktoré sú relevantné pre skúmanú problematiku. V čase návrhu sme mali prístup iba k demoverzii web rozhrania VerbaLexu [11], v ktorom sa mnohé slovesá nenachádzajú, bolo preto nutné valenčné rámce vytvárať na základe pozorovaní textov správ.
Medzi podporované spôsoby viazania sa slov na sloveso patrí spojenie cez pád, predložku alebo infinitív d'alšieho slovesa. Špecifikácia argumentu vychádza priamo z VerbaLexu, ale navyše môže obsahovať atribút lemma. Tento atribút predstavuje obmedzenie, ktoré povolí programu obsadiť daný slot iba frázou, ktorá obsahuje lemma definované atribútom. Toto rozšírenie je možné chápať ako určitú implementáciu DPHR značky štandardne používanej vo VerbaLexe.
Návrh sémantických rolí vychádza z dvojúrovňovej hierarchie VerbaLexu. Podstatnou informáciou pre extrakciu informácie je názov sekundárnej role. Sekundárne role sme rozdelili do troch kategórií:

```
* reagovat

AG(kdo1;<actor_agency:1>;obl)+++VERB+++OBJ(čím7;<state_current:1>;obl;{lemma:doporučení})
# example: Nomura reaguje doporučením Neutral
```

Obr. 4.5: Príklad rámca pre sloveso „reagovat“

- role aktérov: actor, actor_agency, actor_stock
- role pre doporučenia: state, state_current, state_past
- role pre ceny: price, price_change, price_current, price_past

V rámci každej kategórie bola vytvorená jednoduchá hierarchia so spoločnou všeobecnou rolou (actor, state, price), ktorú je možné použiť v nejednoznačných prípadoch, a špeciálnymi rolami, ktoré identifikujú jednoznačne hľadanú informáciu. Role obsahujú tiež číslo významu, avšak toto číslo nie je prakticky využívané. Hierarchiu medzi rolami je možné považovať za určitú implementáciu tohto čísla.
Algoritmus priradenia rolí prechádza postupne všetky vety a ich časti (clauses) a hľadá v nich sloveso, ktoré zodpovedá niektorému slovesu zo slovies s rámcami. Ak dôjde k zhode, program sa pokúša aplikovať postupne všetky valenčné rámce daného slovesa na vetu, jedna fráza tak môže mať jednu a viac rôznych rolí. Častým javom v češtine je tzv. zamlčaný podmet, kedy podmet nie je vo vete prítomný. Z praktického hľadiska to znamená, že pri identifikácií rámcov môže vzniknúť neobsadená rola, ktorú je potrebné pri analýze výpovede obsadiť.

4.7.2 Valenčné rámce podstatných mien a menných entít

V úvode kapitoly sú uvedené dôvody, prečo je nutné vytvoriť rozširujúci mechanizmus k valenčným slovesným rámcom. Ako vhodné rozšírenie sa ukázala implementácia rámcov pre menné skupiny. Podstatné mená (a frázy, v ktorých sa viažu) je možné, podobne ako slovesá, rozšíriť pomocou komplementov, ktoré sú na podstatných menách závislé a viažu sa na ne najčastejšie cez predložkové väzby. Napríklad v spojení *doporučení pro BMW* je *BMW* v roli komplementa, ktorý sa viaže na *doporučení* cez predložkovú väzbu *pro + 4.pád*. Významovo je zároveň možné identifikovať *BMW* ako entitu, ktorej bolo zmenené (určené, zvýšené,...) doporučenie, v hierarchii sekundárnych rolí jej patrí rola actor_stock.
Identifikácia menných rámcov nasleduje za identifikáciou slovesných rám-

```
* cena; cíl
+ role:<price:1>
PAT(pro co4|u čeho2;<actor_stock:1>;obl)
AG(od čeho2;<actor_agency:1>;opt)
ATTR(z čeho2;<price_past:1>;opt)
ATTR(na co4;<price_current:1>;opt)
ATTR(co4;<price:1>;opt)
ATTR(pri čem6|s čím7;<state_current:1>;opt)
ATTR(o co4;<price_change:1>;opt)
```

Obr. 4.6: Príklad rámca pre podstatné mená „cena“ a „cíl“

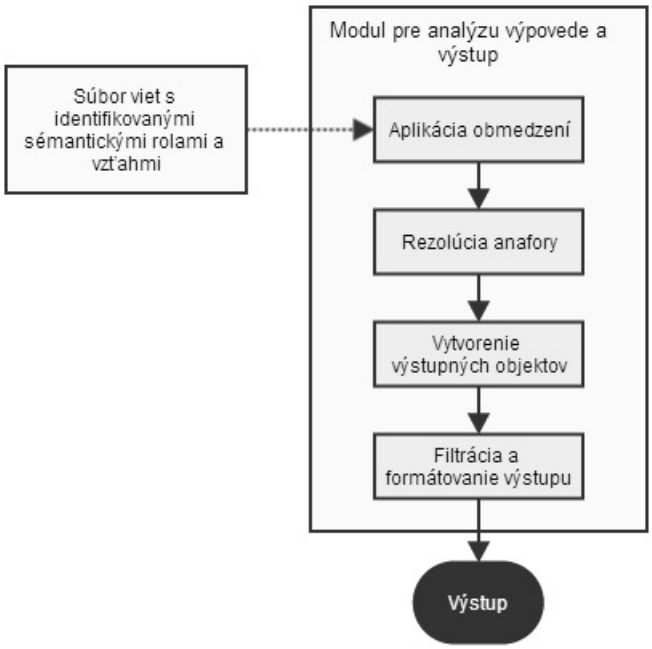
cov. Vo vetách, ktoré obsahujú sloveso s rámcom, a vo vetách, ktoré sú bezslovesné, sa postupne prechádzajú jednotlivé frázy a tokeny a hľadá sa zhoda s niektorým podstatným meno zo súboru. Ak sa zhoda nájde, program danej fráze priradí rolu, ktorá zodpovedá nájdenému podstatnému menu a pokúsi sa role priradiť aj rozšíreniam, ktoré sú definované v rámci nájdeného podstatného mena, a nachádzajú sa v závislostnom strome danej frázy. Z toho vyplýva, že dané rozšírenie nemusí byť priamo závislé na nájdenom podstatnom mene. Dôvodom je tvar závislostného stromu pre vety, ktoré obsahujú frázy ako „zvyšuje doporučení pro *Stock* z *B* na *C*“, kedy jedna fráza s doporučením sa vždy viaže na druhú. Algoritmus na rovnakom princípe vyhľadáva rozšírenia tiež pre identifikované menné skupiny (aktér, cena, doporučenie). Na obrázku 4.6 je možné vidieť príklad menného rámca.

4.8 Analýza výpovede a výstup

Na diagrame na obrázku 4.7 je zobrazený modul pre analýzu výpovede a výstup. Vstupom zo sémantickej analýzy je súbor vetných štruktúr s identifikovanými rolami a vzťahmi. V tejto fáze je potrebné dané role a vzťahy spracovať, vybrať tie, ktoré obsahujú relevantnú informáciu, a vo vhodnom formáte dať túto informáciu na výstup.

4.8.1 Aplikácia obmedzení

Prvou funkciou, ktorú modul vykonáva, je aplikáci obmedzení na role. Na každý typ role sa viaže určité obmedzenie, pokiaľ daná fráza majúca danú rolu toto obmedzenie nespĺňa, daná rola nie je validná. Obmedzenie zodpovedá typu mennej entity, ktorou by daná rola mala byť obsadená. Tento



Obr. 4.7: Modul pre analýzu výpovede a výstup

proces je možné chápať ako určitú nadstavbu NER.
Počas aplikácie slovesných rámcov môže dôjsť k nesprávnemu priradeniu role, či už z dôvodu nejednoznačnosti slovesného rámca alebo chyby vo výstupe syntaktickej analýzy. Algoritmus sa tiež snaží odstrániť role, ktoré neobsahujú konkrétne informácie. Napríklad, majme dvojicu viet *Nomura dáva cenu 21,5 USD* a *Nomura dáva nejvyšší cenu*. Sémantická analýza neobsahuje kontrolu konkrétnej informácie, ktorú daná rola obsahuje, a tak rámec slovesa *dávat* bude aplikovaný na obe vety. Avšak druhá veta neobsahuje potrebnú informáciu o cene a počas fázy aplikácie obmedzení bude rola pre frázu *nejvyšší cenu* označená ako invalidná.
Tento modul sa zároveň snaží špecifikovať všeobecné role. Napríklad, ak fráza *0,5 %* má rolu *price*, modul túto rolu môže špecifikovať na rolu *price_change* na základe znaku „%“ prítomnom vo fráze. Špecifikáciá rolí je však najdôležitejšia v prípade aktérov. Nie vždy je možné pri identifikácií rámcov rozhodnúť, či daná fráza je aktérom typu *agency* alebo *stock*, čo môže spôsobiť problémy pri rezolúcii anafory. Evidentné to je najmä pri bezslovesných vetách. Vo vete *Jefferies - cílová cena 625 USD s doporučením Koupit* nie je bez znalosti kontextu jasná rola pre entitu *Jefferies*, entita môže byť subjektom

pre zmenu doporučenia, ale zároveň aj jej objektom. Bolo preto nutné vytvoriť pravidlá, na základe ktorých je možné niektoré role špecifikovať. V prípade, ak fráza obsahuje okrem mennej entity aj slovo napr. „agentura“, rozhodovanie nie je náročné. Avšak v mnohých prípadoch je nutné nájsť rovnomenné entity v celkom texte a skontrolovať ich role. Aj napriek implementovaným pravidlám nie je vždy možné jednoznačne rozhodnúť o správnej role. Konkrétne pravidlá je možné nájsť v zdrojových kódoch.

4.8.2 Rezolúcia anafory

V zmysle analýzy výpovede je rezolúcia anafory hlavnou funkciou modulu. V našom prípade sa anafora týka najmä aktérov. Role z hierarchie aktérov môžu byť naplnené jedným z nasledujúcich objektov:

- menná entita, napr. Nomura, Apple,...,
- všeobecné podstatné meno, napr. akcie, společnost,...,
- zámeno, napr. která, on,...,
- zamlčaný podmet.

Pre účely výstupu je vhodný jedine objekt prvého typu, tzv. konkrétna menná entita. V ostatných prípadoch je potrebné určiť antecedenta pre daný objekt, pričom vedomosti získané iba pohľadom na danú vetu nie sú postačujúce, frázu identifikujúcu antecedenta je potrebné nájsť v texte. Je tu teda možné vidieť rozdiel oproti klasickej rezolúcií anafory, pri ktorej by antecedentom mohlo byť aj všeobecné podstatné meno, avšak v našom prípade tento typ frázy neposkytuje jednoznačnú identifikáciu konkrétneho aktéra. Pri hľadaní antecedenta algoritmus hľadá frázu, ktorá má rovnakú rolu ako má antecedent, prípadne základnu rolu, tzv. rolu actor, a zároveň daná fráza obsahuje mennú entitu. Algoritmus najskôr prezrie vetu, v ktorej sa anaforický výraz nachádza (to má zmysel najmä pri vedľajších vetách so vzťažnými zámenami), následne frázu s antecedentom hľadá v predchádzajúcich vetách, ktoré prezerá v reverznom poradí. Ak algoritmus vyhovujúcu frázu ani tak nenašiel, prezrú sa aj vety, ktoré nasledujú za anaforickým výrazom.
Takto upravené sémantické role a vzťahy nesú informáciu obsiahnutú v danej vete, resp. jej časti. Jeden aktér sa však môže nachádzať vo viacerých vetách s novou informáciou, a preto pre účely vystupu je potrebné zlúčiť túto informáciu do jedného objektu, ktorý je identifikovaný spoločným kľúčom. Ako vhodný kľúč poslúži menná entita, ktorá má priradenú

rolu *actor_stock*. Úloha nájsť aktéra vo viacerých vetách nie je zložitá, pokiaľ je daný aktér označený rovnakým menom, problém nastáva, ak je aktér označený iným menom, napríklad trhovou skratkou (príklad v kapitole 4.1) alebo inou formou mena (napríklad „Orange Polska“ a „Polský Orange“). Z tohto dôvodu bolo potrebné vytvoriť v module funkciu, ktorá dokáže na základe dvoch názvov určiť, či reprezentujú toho istého aktéra. Primárne algoritmus kontroluje zhodu celých názvov a podreťazcov. Pri skratkách je jednou možnosťou vytvoriť zoznam, ktorý by obsahoval názov entity a trhovú skratku, a táto informácia by bola použitá pri mapovaní. Pri tomto spôsobe riešenia by bol problém s vytváraním týchto zoznamov. Pri spracovaní výstupu morfologickej analýzy je použitý semi-automaticky vytvorený zoznam skratiek a názvov, vytvoriť však duálny zoznam názvov a prislúchajúcich skratiek je komplexnejší problém. Na základe vlastností textov sme sa rozhodoli pre jednoduchšie riešenie. Relevantné správy najčastejšie obsahujú informáciu o zmene doporučenia pre jedného trhové aktéra, preto je často možné automaticky jedinú v texte nájdenú trhovú skratku priradiť jedinému aktérovi s rolou *actor_stock*. V prípade ak ide o komplikovanejší text (napr. súhrn trhových udalostí), názvy aktérov sú zvyčajne vo formáte *názov (skratka ; cena)* a postačuje pri ich identifikácií kontrola celého názvu, prípadne podreťazca. Ak algoritmus mapovania potvrdí, že dané dva názvy označujú toho istého aktéra, informáciu z oboch viet zlúči do jedného objektu. V prípade nezhody sú objekty vytvorené dva.

4.8.3 Výstup

V kapitole 4.1 sme špecifikovali informáciu, ktorú v textoch hľadáme. Avšak z popisu slovesných a menných rámcov vyplýva, že identifikujú aj samotné zmeny cien bez informácie o zmene doporučenia. Najmä pri komplikovaných správach je preto nutné filtrovať iba objekty, ktoré obsahujú okrem informácie o zmene ceny aj informáciu o zmene doporučenia.
Poslednou činnosťou modulu je formátovanie výstupu a samotný výstup. Kedže vytvorený program nemal ambície stať sa aplikáciou, ktorú by ľudia mohli priamo používať cez webové rozhranie alebo ako mobilnú aplikáciu, rozhodli sme sa pre formálnejší zápis výstupu vo formáte JSON [6]. JSON (Javascript Object Notation) je otvorený štandard pre výmenú dátových objektov v tvare atribút–hodnota, čo ho robí vhodnou voľbou pre reprezentáciu výstupnej štruktúry. JSON predstavuje alternatívu k XML pre prenos dát.
Pre potreby dokumentácie bola vytvorená JSON schema, ktorá môže neskôr poslúžiť pre ďalší vyvoj aplikácie. Samotná štruktúra je pomerne jed-

noduchá, avšak obsahuje väčší počet atribútov, a tak z dôvodu väčšieho rozmeru je umiestnená do elektronických príloh. Jej aplikáciu uvedieme na tomto mieste na nasledujúcom príklade. Majme jednoduchý text:

Commerzbank snižuje doporučení pro BMW, cíl ale navýšila. Commerzbank snižuje doporučení pro automobilku BMW z "přidat"na "držet". Cílovou cenu ale navýšila z 87 EUR na 92 EUR. Akcie BMW (BMW) posilují o 0,80 % na 88,00 EUR.

Na obrázku 4.8 je zobrazený vygenerovaný výstupný JSON objekt.


```
{
    "stock abbreviation": "BMW",
    "stock name": "BMW",
    "price change": "0,80 %",
    "current price": "88,00 EUR",
    "agencies": [
        {
            "agency name": "Commerzbank",
            "current price": "92 EUR",
            "current recommendation": "držet",
            "past recommendation": "přidat",
            "past price": "87 EUR"
        }
    ]
}
```


Obr. 4.8: Príklad výstupného JSON objektu

5 Evaluácia systému

V úvode kapitoly 3 sme uviedli najznámejšie a najvyužívanejšie metriky pre evaluáciu IE. Pre evaluáciu je ďalej nutné zvoliť, čo konkrétne budeme vyhodnocovať. V nasledujúcej kapitole sú postupne uvedené spôsob testovania, dosiahnuté výsledky a na záver ich zhodnotie a prípadne zdôvodnenie chýb.

5.1 Metodika testovania

Ako bolo viackrát v texte spomenuté, výstup systému je možné považovať za záznamovú štruktúru, ktorá nesie informáciu o zmene doporučenia pre jedného trhového aktéra jednou, alebo viacerými agentúrami, a tak je možné sa pozerať na tento záznam ako na objekt evaluácie. Záznam je možné považovať za správny (correct), pokiaľ sú správne extrahované všetky položky záznamu – aktéri, doporučenia a cieľové ceny. Akonáhle je aspoň jedna položka extrahovaná nesprávne (prípadne nie je extrahovaná), záznam je označený celý ako nesprávny (spurious). Pokiaľ sa v texte nachádza informácia o zmene doporučenia, a táto informácia nie je extrahovaná, daný neextrahovaný záznam sa označí ako chýbajúci (missing).
Kompletnosť a správnosť celého záznamu sú však pomerne prísne kritéria, ktoré síce môžu byť indikátorom celkovej funkčnosti systému, avšak iba na ich základe je náročné zistiť efektívnosť, úspešnosť a chybovosť extrakcie jednotlivých informácií, ktoré celkový záznam vytvárajú. Preto sme sa rozhodli evaluovať aj binárne a ternárne vzťahy, resp. vzťahy medzi dvomi alebo tromi entitami, ktoré tvoria výstupný objekt.
Na ľubovoľnú n–ticu entít v texte je možné sa pozerať ako na n–árny vzťah, ktorý, pokiaľ sa v texte nachádza, je možné klasifikovať určitým typom. V zmysle definície úlohy systému a nami zvolenej hierarchie potencionálnych vzťahov je spoločnou entitou všetkých vzťahov aktér, ktorému bolo zmenené doporučenie, a ktorý môže byť vzťahu. Jeho vzťah k druhej entite je následne označený rolou, ktorú má daná entita vo výstupnom zázname. Vo vete *Coca-cola roste o 0,5%* je možné označiť vzťah medzi *Coca-cola* a *0,5%* napr. predikátom *price_change(Coca-cola, 0,5%)*. Ak je takto označený vzťah aj vo výstupe, je možné ho označiť ako správny.Na druhej strane, v ternárnom vzťahu figuruje vždy dvojica aktérov a entita, ktorá reprezentuje doporučenie alebo cenu. Vzťah vo vete *Nomura dáva Coca-cole doporučení Hold* môžeme v tomto zmysle zapísať ako *current_recommendation(Nomura,*

Coca-cola, Hold).
Pre potreby evaluácie sú tieto vzťahy rozdelené do dvoch skupín, vzťahy týkajúce sa cien a vzťahy týkajúce sa doporučení. Obe tieto skupiny sú vyhodnocované osobitne a na záver tiež spolu. Vzťah *agentúra – akcia* je súčasťou ternárnych vzťahov a nie je vyhodnocovaný osobitne.
Na tomto mieste je vhodné spomenúť, že samostatné vyhodnocovanie NER nie je súčasťou evaluácie.
Texty použité pri evaluácií boli zozbierané medzi ferbuárom 2014 a májom 2014. Spolu bolo vyhodnocovaných 61 textov správ, z ktorých väčšina obsahovala jeden alebo viac výstupných záznamov. Jednotlivé výstupné objekty bolo nutné manuálne vyhodnotiť a zapísať do výsledkovej tabuľky.

5.2 Výsledky

V tabuľke 5.1 sú zobrazené výsledky pre jednotlivé kategórie vzťahov (vzťahy týkajúce sa doporučení a cien, celkový výstupný objekt, resp. záznam), ktoré boli definované v predchádzajúcej podkapitole. Tabuľka s výsledkami pre jednotlivé texty správ je súčasťou elektronických príloh. Na prvý pohľad

Typ vzťahu	**Precision**	**Recall**	**F–skóre**
Záznam	53,2%	76,7%	62,9%
„Doporučenie“	77,6%	81,3%	79,4%
„Cena“	74,7%	75,2%	74,9%
„Doporučenie“ + „Cena“	75,9%	77,7%	76,8%

Tabuľka 5.1: Výsledky evaluácie

je možné si všimnúť nízku hodnotu metriky precision pre extrakciu celého vzťahu. Je potrebné si uvedomiť, že celý záznam najčastejšie pozostáva zo 4 až 6 informácií, ktoré musia byť všetky určené správne a žiadna nemôže chýbať. Častým javom je napr. nesprávne priradenie ceny, kedy namiesto vzťahu *current_price(stock, price)* (tzv. akcie sa obchodujú na danej cene) je daný vzťah označený ako ternárny vzťah *current_price(agency, stock, price)* (tzv. agentúra doporučuje obchodovať akcie na danej cene). Vďaka tomu sú aj výsledky vzťahov, ktoré sa týkajú cien, nižšie ako výsledky pre vzťahy doporučení.
Na úrovni jednotlivých vzťahov sú však výsledky pomerne dobré. Pre porovnanie systém [14] (spomínaný v kapitole 3.4) dosahuje F-skóre 69,9%. Úlohy oboch systémov sú samozrejme veľmi odlišné, a preto aj vyššie skóre navrhnutého systému je možné pripísať viacerým faktorom. Jedným z naj-

dôležitejších je fakt, že v našom prípade ide o systém špeciálne zameraný na jednu oblasť, čomu bol prispôsobený celý návrh a najmä spracovanie dát na rôznych úrovniach.
Nesprávne určené a chýbajúce vzťahy je možné vysvetliť viacerými spôsobmi. Už v predchádzajúcich odstavcov bol spomenutý problém s nesprávnym priradením cien. Okrem toho boli chyby spôsobené napr. nerozoznaným formátom názvu doporučenia alebo ceny. Tieto chyby najčastejšie súvisia s morfologickou analýzou a fázou NER. Vo všeobecnosti je však možné výstup týchto dvoch procesov považovať za pomerne dobrý a málo chybový.
Najväčší podiel na chybách mala však analýza výpovede, a to či už klasická rezolúcia anafory, ale tiež nesprávne priradenie role *agency* alebo *stock* alebo rezolúcia koreferencie v zmysle identifikácie všetkých textových entít, ktoré označujú tú istú entitu v reálnom svete. To sa prejavilo najmä v dlhších textoch a textoch s bezslovesnými vetami. Implementácia vhodnejšieho algoritmu by mohla výrazne zlepšiť vysledky.

6 Záver

Cieľom práce bolo navrhnúť a vytvoriť systém pre extrakciu nákupných doporučení z textov burzových správ. Ako podklady pre túto úlohu slúžili získané informácie o spracovaní prirodzeného jazyka a o oblasti extrakcie informácií vo všeobecnosti, ale tiež špeciálne zamerané na český jazyk. Cieľom bolo tiež overiť využitie nástrojov vyvíjaných v CZPJ na praktických úlohách počas implementácie navrhnutého systému.
V kapitole 2 sme sa venovali problematike spracovania českého jazyka. Postupovali sme na základe rovinového modelu, kde okrem popisu vybraných vlastností bol priestor venovaný prehľadu niektorých nástrojom z CZPJ, ktoré sú dostupné aplikačnému programátorovi. Tento všeobecný popis bol doplnený v kapitole 4 o popis praktických problémov, ktoré sú spôsobené charakterom vstupných textov a požiadavkami úlohy. Kapitola 3 slúži ako informačný podklad z oblasti extrakcie informácie pre návrh systému.
Na základe informácií spracovaných v kapitolách 2 a 3 bol navrhnutý a implementovaný vlastný systém. Charakteristika vstupných textov spôsobila, že nástroje morfologickej (ajka, desamb) a syntaktickej (SET) analýzy nebolo možné priamo použiť. Po dôkladnej analýze bol vytvorený modul pre NER, ktorý bol tiež využitý pri úprave výstupných údajov morfologickej analýzy pre potreby syntaktickej analýzy nástrojom SET. Pre extrakciu vzťahov vo vete bol vytvorený parser valenčných rámcov slovies a menných skupín a tiež modul pre rezolúciu anafory v texte. Takto extrahovaná informácia je daná na výstup vo formáte JSON.
Výsledky implementovaného systému pre binárne a ternárne vzťahy dosahujú F-skóre 76,8%. Okrem výsledkov pre rôzne typy vzťahov práca obsahuje aj zdôvodnenie nesprávnych a chýbajúcich inštancií vzťahov.
Ďalší vývoj sa môže uberať viacerými smermi. Tým prvým môže byť vytvorenie funkčnej aplikácie pre koncových užívateľov. Potrebné by bolo vytvoriť modul pre automatické získavanie článkov, pričom zdrojový kód už teraz obsahuje skripty pre automatickú extrakciu textov webových stránok. Ďalším krokom by bolo vytvorenie užívateľského rozhrania, či už vo forme webovej stránky, alebo mobilných či e-mailových notifikácií.
Druhý smer by sa mohol zamerať na odstránenie nedostatkov, ktoré boli naznačené v kapitole 5.2. Týka sa to najmä rezolúcie anafory a koreferencie, implementácia vhodnejšieho algoritmu by mohla zvýšiť efektivitu celého systému.

Literatúra

[1] *ajka tagset* [online]. Centrum zpracování přirodzeného jazyka. [cit. 2014-05-07]. Dostupné z: http://nlp.fi.muni.cz/projekty/ajka/tags.pdf.

[2] *Extraction of facts* [online]. Centrum zpracování přirodzeného jazyka. [cit. 2014-05-07]. Dostupné z: http://nlp.fi.muni.cz/projekty/set/efa/wwwefa.cgi/first_page.

[3] *Balkanet Project Home Page* [online]. Balkanet. [cit. 2014-05-07]. Dostupné z: http://www.dblab.upatras.gr/balkanet/.

[4] *What is a recommendation? definition and meaning* [online]. InvestorWords. [cit. 2014-05-07]. Dostupné z: http://www.investorwords.com/4090/recommendation.html.

[5] *Novinky z burzy, komentáře, zprávy, aktuality, akcie | Fio banka* [online]. Fio banka. [cit. 2014-05-07]. Dostupné z: http://www.fio.cz/zpravodajstvi/zpravy-z-burzy.

[6] *Introducing JSON* [online]. [cit. 2014-05-10]. Dostupné z: http://www.json.org/.

[7] *NER Demo* [online]. [cit. 2014-05-12]. Dostupné z: https://nlp.fi.muni.cz/projekty/ner/v2/.

[8] *Welcome to Python.org* [online]. 2014. [cit. 2014-05-21]. Dostupné z: https://www.python.org/.

[9] *SET* [online]. Centrum zpracování přirodzeného jazyka. [cit. 2014-05-07]. Dostupné z: http://nlp.fi.muni.cz/trac/set/.

[10] *The Stanford NLP (Natural Language Processing) Group* [online]. Stanford University. [cit. 2014-05-07]. Dostupné z: http://nlp.stanford.edu/software/corenlp.shtml.

[11] *VERBALEX* [online]. [cit. 2014-05-07]. Dostupné z: http://nlp.fi.muni.cz/verbalex/htmlDEMO/generated/alphabet/index.html.

[12] AGICHTEIN, Y. *Extracting Relations from Large Text Collections*. Ph.D. dissertation, Columbia University, 2005.

[13] ALLEN, J. *Natural Language Understanding*. Redwood City, CA: Benjamin/Cummings Pub., 2nd edition, 1995.

[14] BAISA, V. – KOVÁŘ, V. Information Extraction for Czech Based on Syntactic Analysis. In VETULANI, Z. (Ed.) *Human Language Technologies as a Challenge for Computer Science and Linguistics, Proceedings of 5th Language and Technology Conference*, s. 466–470, Poznań, 2011. Funcacja Universytetu im. A. Mickiewicza. ISBN 978-83-932640-1-8.

[15] DACIUK, J. *Incremental Construction of Finite-State Automata and Transducers, and their Use in the Natural Language Processing*. Ph.D. dissertation, Technical University of Gdańsk, Poland, 1998.

[16] DACIUK, J. – WATSON, B. W. – WATSON, R. E. Incremental Construction of Minimal Acyclic Finite State Automata and Transducers. In KARTTUNEN, L. (Ed.) *FSMNLP'98: International Workshop on Finite State Methods in Natural Language Processing*. Somerset, New Jersey: Association for Computational Linguistics, 1998. s. 48–55.

[17] FELDMAN, R. – SANGER, J. *The Text Minin Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007. ISBN 10-0-511-33507-5.

[18] GRISHMAN, R. Information Extraction: Capabilities and Challenges, 2012. Notes prepared for the 2012 International Winter School in Language and Speech Technologies.

[19] GRISHMAN, R. Information Extraction: Techniques and Challenges. In *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, SCIE '97, s. 10–27, London, UK, UK, 1997. Springer-Verlag. Dostupné z: `http://dl.acm.org/citation.cfm?id=645856.669801`. ISBN 3-540-63438-X.

[20] HAJIČ, J. *Complex Corpus Annotation: The Prague Dependency Treebank*, s. 54–73. Veda, Bratislava, Slovakia, Bratislava, Slovakia, 2006. ISBN 80-224-0880-8.

[21] HAJIČ, J. et al. Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In *Proceedings of ACL 2001*, Toulouse, France, 2001. Association for Computational Linguistics.

[22] HLAVÁČKOVÁ, D. – HORÁK, A. VerbaLex - New Comprehensive Lexicon of Verb Valencies for Czech. In *Computer Treatment of Slavic and East European Languages*, s. 107–115, Bratislava, Slovakia, 2006. Slovenský národný korpus. ISBN 80-224-0895-6.

[23] HOBBS, J. R. Pronoun Resolution. *SIGART Bull.* February 1977, , 61, s. 28–28. ISSN 0163-5719. doi: 10.1145/1045283.1045292. Dostupné z: http://doi.acm.org/10.1145/1045283.1045292.

[24] HOLAN, T. – ŽABOKRTSKÝ, Z. Combining Czech Dependency Parsers. In SOJKA, P. – KOPEČEK, I. – PALA, K. (Ed.) *Lecture Notes in Artificial Intelligence, Text, Speech and Dialogue. 9th International Conference, TSD 2006, Brno, Czech Republic, September 11–15, 2006, Proceedings*, 4188 / *Lecture Notes in Computer Science*, s. 95–102, Berlin / Heidelberg, 2006. Masarykova univerzita, Springer. ISBN 978-3-540-39090-9.

[25] HORÁK, A. *Computer Processing of Czech Syntax and Semantics*. Brno, Czech Republic : Librix.eu, 1st edition edition, 2008. ISBN 978-80-7399-375-7.

[26] HORÁK, A. et al. Dependency and Phrasal Parsers of the Czech Language: A Comparison. In *Proceedings of 10th International Conference on Text, Speech, and Dialogue (TSD 2007)*, s. 76–84, Berlin, Heidelberg, 2007. Springer. ISBN 978-3-540-74627-0.

[27] JURAFSKY, D. – MANNING, C. *Natural Language Processing | Coursera* [online]. Coursera. [cit. 2014-05-07]. Dostupné z: https://www.coursera.org/course/nlp.

[28] KOPŘIVOVÁ, M. – KOCEK, J. *Popis morfologických značek - poziční systém* [online]. Český národní korpus. [cit. 2014-05-07]. Dostupné z: http://ucnk.ff.cuni.cz/bonito/znacky.php.

[29] KOVÁŘ, V. Syntaktická analýza s využitím postupné segmentace věty. Diplomová práce, Masarykova univerzita, Fakulta informatiky, 2008.

[30] KOVÁŘ, V. – HORÁK, A. – JAKUBÍČEK, M. Syntactic Analysis Using Finite Patterns: A New Parsing System for Czech. In *Human Language Technology. Challenges for Computer Science and Linguistics*, s. 161–171, Berlin/Heidelberg, 2011. Springer. Dostupné z: http://dx.doi.org/10.1007/978-3-642-20095-3_15. ISBN 978-3-642-20094-6.

[31] ŠMERK, P. Unsupervised Learning of Rules for Morphological Disambiguation. *Lecture Notes in Computer Science*. 2004, 3206. ISSN 0302-9743.

[32] ŠMERK, P. Fast Morphological Analysis of Czech. In *Proceedings of the Raslan Workshop 2009*, Brno, 2009. Masarykova univerzita. ISBN 978-80-210-5048-8.

[33] ŠMERK, P. – SOJKA, P. – HORÁK, A. Morphemic Analysis: A Dictionary Lookup Instead of Real Analysis. In *First Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2007*, s. 77–85, Brno, 2007. Masaryk University. Dostupné z: `http://nlp.fi.muni.cz/raslan/2007/`. ISBN 978-80-210-4471-5.

[34] MITKOV, R. – SB, W. W. Anaphora Resolution: The State Of The Art. Technical report, University of Wolverhampton, 1999.

[35] MUGGLETON, S. – RAEDT, L. D. Inductive Logic Programming: Theory and Methods. *JOURNAL OF LOGIC PROGRAMMING*. 1994, 19, 20, s. 629–679.

[36] NĚMČÍK, V. Saara: Anaphora Resolution on Free Text in Czech. In ALEš HORáK, P. R. (Ed.) *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2012*, s. 3–8, Brno, 2012. Tribun EU. ISBN 978-80-263-0313-8.

[37] NOVÁK, M. – ŽABOKRTSKÝ, Z. Resolving Noun Phrase Coreference in Czech. *Lecture Notes in Computer Science*. 2011, 7099, s. 24–34. ISSN 0302-9743.

[38] PALA, K. Počítačové zpracování přirozeného jazyka, 2000. course material.

[39] PALA, K. – ŠEVEČEK, P. *Valence českých sloves*, s. 41–54. Masarykova univerzita Brno, Brno, 1997. ISBN 80-210-1606-x.

[40] PALA, K. – SMRŽ, P. Building Czech Wordnet. *Romanian Journal of Information Science and Technology*. 2004, 7. ISSN 1453-8245. Dostupné z: `http://nlp.fi.muni.cz/publications/romjist2004_pala_smrz/`.

[41] PANDEY, V. K. – SOLANKI, S. – SHARMA, K. Article: A Technique for Anaphora Resolution of Text. *International Journal of Applied Infor-*

mation Systems. January 2013, 5, 1, s. 28–33. Published by Foundation of Computer Science, New York, USA.

[42] POINTAL, L. *dev:treetaggerwrapper [LPointal]* [online]. [cit. 2014-05-21]. Dostupné z: `http://perso.limsi.fr/pointal/dev:treetaggerwrapper`.

[43] POMIKÁLEK, J. Unitok, 2009. Computer Program.

[44] SARAWAGI, S. Information Extraction. *Foundations and Trends in Databases*. 2008, 1, 3, s. 261–377.

[45] SEDLÁČEK, R. – SMRŽ, P. A New Czech Morphological Analyser ajka. In *Proceedings of the 4th International Conference TSD 2001*, s. 100–107, Berlin, 2001. Springer-Verlag. ISBN 3-540-42557-8.

[46] SOMERS, H. L. *Valency and Case in Computational Linguistics*. Edinburgh: Edinburgh University Press, 1987. ISBN 0852245181.

[47] STRAKOVÁ, J. – STRAKA, M. – HAJIČ, J. A New State-of-The-Art Czech Named Entity Recognizer. In HABERNAL, I. – MATOUŠEK, V. (Ed.) *Text, Speech and Dialogue: 16th International Conference, TSD 2013. Proceedings*, 8082 / *Lecture Notes in Computer Science*, s. 68–75, Berlin / Heidelberg, 2013. Západočeská univerzita v Plzni, Springer Verlag. ISBN 978-3-642-40584-6.

[48] STRAŇÁKOVÁ-LOPATKOVÁ, M. – ŽABOKRTSKÝ, Z. Valency Dictionary of Czech Verbs: Complex Tectogrammatical Annotation. In RODRÍGUEZ, M. G. – ARAUJO, C. P. S. (Ed.) *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, 3, s. 949–956. ELRA, 2002.

[49] VOSSEN, P. et al. The EuroWordNet Base Concepts and Top Ontology. Technical report, Paris, France, France, 1998.

[50] WANZARE, L. Coreference Resolution, 2010. Multomodal Ontology Based Dialogue Systems Seminar, Computational Linguistics Department, Saarland University.