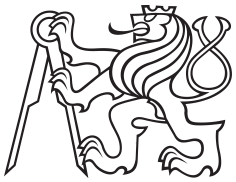
Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra telekomunikační techniky

Analýza dat pomocí znalostních sítí

Bc. Arina Lebedeva

Vedoucí práce: Ing. Radek Mařík, CSc.
Obor: Elektronika a komunikace
Studijní program: Komunikační sítě a internet
Květen 2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Lebedeva** Jméno: **Arina** Osobní číslo: **495518**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra telekomunikační techniky**

Studijní program: **Elektronika a komunikace**

Specializace: **Komunikační sítě a internet**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Analýza dat pomocí znalostních sítí

Název diplomové práce anglicky:

Data Analysis using Knowledge Networks

Pokyny pro vypracování:

1. Vytvořte přehled metod vytváření znalostních sítí.
2. Navrhněte postup identifikující relace mezi vybranými objekty a aspekty textových dokumentů.
3. Vyberte vhodnou sestavu metod řešící zadaný cíl, případně je modifikujte a implementujte.
4. Na experimentálních datech ověřte vlastnosti metody.
5. Proveďte diskusi získaných výsledků.

Seznam doporučené literatury:

[1] Newman, M.: Networks: an introduction, 2010, Oxford University Press, Inc., ISBN: 978-0-19920-665-0.
[2] Kejriwal, M. - Knoblock, C. A. - Szekely, P.: Knowledge Graphs: Fundamentals, Techniques, and Applications, 2021, Adaptive Computation and Machine Learning series, MIT Press. 560 pp. ISBN: 978-0-26204-509-4.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Radek Mařík, CSc. katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **03.02.2022** Termín odevzdání diplomové práce: **20.05.2022**

Platnost zadání diplomové práce: **30.09.2023**

Ing. Radek Mařík, CSc. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

4.3.2022
Datum převzetí zadání

Podpis studentky

Poděkování

Děkuji vedoucímu práce Ing. Radkovi Maříkovi, CSc. za neocenitelné rady a pomoc při tvorbě diplomové práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 20. května 2023

Abstrakt

Tato práce se zaobírá problematikou tvorby znalostních sítí, konkrétně tvorbou znalostních grafů, jež jsou digitální reprezentací entit reálného světa a vztahů mezi nimi. Teoretická část práce se skládá z popisu procesu tvorby znalostních grafů. V ní jsou rozebrány takové pojmy, jako jsou NLP pipelines, tokenization, POS tagging, NER (Named-entity recognition) a RE (Relation Extraction). Praktická část ukazuje proces tvorby znalostní sítě na reálních datech a demonstruje znalostní graf vygenerovaný z několika tisíc novinářských článků dodaných ústavem formální a aplikované lingvistiky.

Klíčová slova: strukturalizace dat, znalostní síť, znalostní graf, Python, Neo4j

Vedoucí práce: Ing. Radek Mařík, CSc.

Abstract

This thesis concerns itself with the issue of knowledge networks, specifically the creation of knowledge graphs, which are a digital representation of real-world entities and the relationships between them. The theoretical part consists of a description of the process of creating knowledge graphs. It discusses concepts such as NLP pipelines, tokenization, POS tagging, NER (Named-entity recognition) and RE (Relation Extraction). The practical part shows the process of creating a knowledge network from real data and demonstrates a knowledge graph generated from several thousand journalistic articles supplied by the Institute of Formal and Applied Linguistics.

x

Keywords: data structuring, knowledge network, knowledge graph, Python, Neo4j

Title translation: Data Analysis using Knowledge Networks

Obsah

Obrázky

Tabulky

Kapitola 1

Úvod

S růstem objemu produkovaných dat a vývojem sociálních médií zvyšuje se poptávka po analýze textových dokumentů a vizualizaci dat. S analýzou textových dokumentů těsně souvisí jejích strukturalizace, která hraje zásadní roli v procesu extrakce užitečných informací a je významném pomocníkem při identifikaci vztahů mezi různými objekty. Jedním z nejlepších způsobu strukturalizace dat je využití znalostní sítě neboli znalostního grafu.

Znalostní graf je grafické znázornění entit reálného světa a vztahů mezi nimi. Tento termín se často používá k označení rozsáhlých strukturovaných databází informací, které se využívají například při vyhledávání informací v prohlížečích nebo při zpracování úloh přirozeného jazyku.

Cílem diplomové práce je navržení metody identifikující relace mezi vybranými objekty a aspekty textových dokumentů a následné ukládání získaných užitečných informací do znalostního grafu.

Práce je rozdělena na teoretickou a praktickou části. V první kapitole teoretické části je popsán algoritmus vytvoření znalostních grafů a je uveden stručný přehled metod, umožnujících získávání užitečných informací z textových dokumentů. Ve druhé kapitole je provedena analýza datového korpusu zvoleného jako zdroj informací znalostního grafu.

Praktická část diplomové práce se skládá z kapitoly popisující implementaci zvolené metody pro identifikaci entit a relací a kapitoly s experimentálními dotazy nad grafovou databází Neo4j.

Kapitola 2

Metodologie tvorby znalostních grafů

Pojem *znalostí graf* byl poprvé zaveden v roce 1972 Edwardem W. Schneiderem v jeho článku „The Interface System and Its Implications For Sequence Control and Data Analysis“, kde autor diskutoval o možnostech ukládání velkého množství dat a jejích reprezentace pomocí zjednodušené grafové struktury neboli znalostního grafu.[3]

Navzdory tomu, že se termín objevil ve 20. století, k jeho popularizaci došlo až v roce 2012, kdy společnost Google prezentovala svůj znalostní graf využívající tzv. *knowledge base* - databázi, ve které informace jsou uložené ve formě fragmentů znalostí (faktů) vzájemně propojených na několika úrovních.[3]

V moderním znění pojem *znalostní graf*, známý také jako sémantická síť, lze chápat jako síť uloženou v podobě orientovaného grafu a ilustrující vztah mezi objekty neboli entity reálného světa. Hlavními komponenty znalostního grafů jsou uzly, hrany a popisky (labels). Uzly reprezentují objekty, jako jsou například lidé, společnosti, města, planety, atd. Hrany spojují dvojice uzlů a definují vztah neboli relaci mezi nimi. Popisky vystihují význam relací.[1]

Tak na obrázku 2.1 je zobrazen graf se třemi uzly a dvěma hranami. Každý uzel zde reprezentuje svou třídu objektů, a proto uzly se liší barvou. První třída reprezentuje entitu *"Lidé"*, druhá reprezentuje entitu *"Země"*, třetí reprezentuje entitu *"Mezinárodní organizace"*. Všechny hrany mají svůj popisek, který říká, jakým způsobem reprezentant každé entity se vztahuje k ostatním objektům. Konkretně v níže uvedeném příkladu vyskytují se relace *"je prezidentem"* a *"je členem"*.

Obrázek 2.1: Jednoduchý znalostní graf

Jak je zřejmě z výše uvedeného příkladu, pro tvorbu znalostního grafu je potřeba mít jasně definované entity a relace, což znamená, že data, ze kterých se tvoří graf, musí mít určitou strukturu. V anglické literatuře procesu

strukturování dat se říká *information extraction*. Do češtiny tento pojem lze přeložit jako *extrakce informací*.

Ve skutečnosti *extrakce informací* i je hlavním kamenem úrazu v procesu vytváření znalostního grafu. Způsobeno to tím, že každý jazyk je jedinečný, a proto nelze napsat univerzální aplikaci, která by stejně dobře prováděla strukturování dat ve všech jazycích.

Nicméně díky tomu, že úloha *extrakci informací* patří do oboru NLP (Natural Language Processing), při strukturování dat lze postupovat podle algoritmů uvedených v knize *"Natural Language Processing for the Semantic Web"*[7].

Finální algoritmus reprezentující proces tvorby znalostních grafů lze definovat následovně:

1. Provést extrakce informací:
   a. provést lingvistické předzpracování (linguistic pre-processing);
   b. provést rozpoznávaní pojmenovaných entit (NER - Named Entity Recognition);
   c. provést extrakci vztahů (RE - Relation Extraction).
2. Uložit extrahované informace do databázi.

2.1 Lingvistické předzpracování

Metody *lingvistického předzpracování* textu lze rozdělit na:

1. metody založené na **znalostech** či **pravidlech**;
2. metody založené na **strojovém učení**[7].

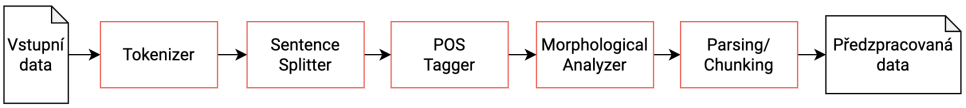
Metody založené na **znalostech** či **pravidlech** jsou tradičnějšími metodami strukturování dat, které s příchodem výkonných počítačů utrácí svou popularitu. Tyto metody používají ručně definovaná pravidla a spoléhají na znalost gramatiky jazyku, což je dělá flexibilnějšími, nicméně v případě existence velkého množství výjimek či nedodržení pravidel, tyto metody přestávají být účinné. Například v sociálních sítích lidé často nepoužívají velká písmena pro vlastní podstatná jména, což dělá pravidlo "vlastní podstatné jméno vždy začíná velkým písmenem"neplatným.

Nicméně velkou výhodou znalostních metod je snadné porozumění výsledkům. Když program nesprávně identifikuje slovo, vývojář může zkontrolovat pravidla, zjistit, proč došlo k chybě, a následovně upravit pravidlo či přidat nové. Psaní pravidel však může být poměrně časově náročným procesem, a pokud se specifikace úlohy mění, nastává riziko, že vývojář bude muset přepsat velké množství pravidel.

Metody založené na **strojovém učení** jsou modernějšími metodami strukturování dat. Funkčnost takových metod závisí na počtu předpracovaných neboli trénovacích dat, které vstupují do programu.

V případě existence dostačujícího množství předzpracovaných dat účinnost takových metod roste, což umožňuje získat rozumné výsledky s velmi malým úsilím, avšak získání či vytvoření dostatečného množství trénovacích dat je často extrémně problematickým a zdlouhavým procesem. Tato závislost na předzpracovaných datech také znamená, že adaptace na nové typy textu je nákladná, protože vyžaduje značné množství nových trénovacích dat. Lidsky čitelná pravidla se proto obvykle snáze přizpůsobují novým jazykům a typům textu, než pravidla vytvořená ze statistických modelů.

Všechny úkoly *lingvistického předzpracování* textu lze rozdělit na několik dílčích úkolů, každý ze kterých se zpracovává pomocí svého programů neboli komponenty. Seznam všech komponent je zobrazen na obrázku 2.2.

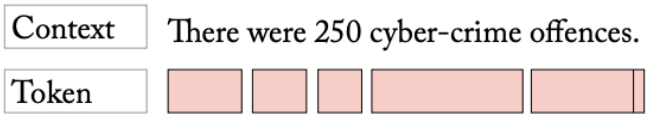


Obrázek 2.2: Pipeline lingvistického předzpracování textu[7]

Jak lze vidět, první fází je typicky tokenizace, během které dochází k rozdělení textu na jednotlivá slova (tokeny). Dále nasleduje fáze dělení textu na věty, po které dochází k určení slovních druhu tokenů. Následně probíhá morfologická analýza slov, hlavním úkolem které je nalezení kořenů tokenů. Poslední fází je syntakticky rozbor jednotlivých vět.

2.1.1 Tokenizace

Tokenizace je proces rozdělení textu na jednotlivé částí nazývané **tokeny**. V závislosti na vlastnostech tokenizeru token může reprezentovat různé objekty, nicméně obvykle tokeny reprezentují slova, čísla, interpunkční znaménka či symboly. Za účelem zkvalitnění výstupních dat během procesu tokenizace může taky docházet k určení třídy tokenu (slovo či číslo), délky slova, které reprezentuje token, či zjištění informace, zdá-li token reprezentuje slovo, které je napsáno s velkým písmenem, atd.[7] Příklad tonizované věty je zobrazen na obrázku 2.3.



Obrázek 2.3: Reprezentace tokenizované věty.[7]

Zde je vidět, že tokenizer rozložil text na tokeny na základě přítomností mezer, a proto tečka nebyla identifikována jako samostatný token.

Nejznámějšími tokenizery pro anglické texty jsou tokenizer maximální entropie OpenNLP TokenizerME, tokenizer PTBTokenizer, který je součastí sady nástrojů Stanford CoreNLP a tokenizer GATE Tokenizer založený na Unicode, což při úpravě jeho pravidel umožňuje jeho použití pro tokenizaci

textu ve všech evropských jazycích (tokenizer používá rozdělení na tokeny na základě přítomností mezery).[7]

Pro tokenizaci českých textu lze využít nástroje NameTag od ÚFALu či MorphoDiTa (Morphological Dictionary and Tagger).[11]

2.1.2 Rozdělení na věty

Rozdělení na věty je proces analýzy textu, při kterým na základě přítomnosti interpunkčních znamének dochází k rozdělení textů na jednotlivé části. Během tohoto procesu často dochází k problémům s určením začátků a konců vět, protože v textech se hodně často používají zkratky (např. či tzn.), přímá řeč, trojtečky, tabulky a vzorečky, a tak při návrhu algoritmu či úpravě již existujícího řešení je potřeba toto všechno brát v úvahu.[7]

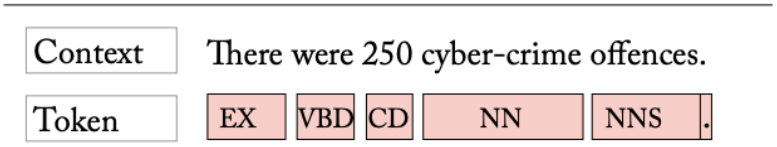
Nejznámějšími nástroji pro rozdělení anglických textů na věty jsou OpenNLP a GATE.[7] Pro texty v češtině lze použit nástroje NameTag a MorphoDiTa využívané taky pro tokenizaci textu.[11]

2.1.3 Tagování

Tagování či anglicky POS tagging je proces analýzy všech slov v textu. Během tohoto procesu každému slovu se přiřazuje tag, jednoznačně určující do jaké třídy slov (podstatná jména, slovesa, přídavná jména, zájmena, číslovky atd.) spadá konkretní token. V závislosti na tom, v jakém jazyce je napsán text, může taky docházet například k určení rodů podstatných jmen.[7]

Nejběžnějšími nástroji pro provedení tagování anglických textů jsou GATE tagger, Brown corpus, LOB Corpus. Pro tagování českých textů lze využit NameTag a MorphoDiTa. Všechny tyto nástroje využívají strojové učení pro predikci správného POS tagu a liší se sadou tokenů.[7][11]

Příklad tagované věty je zobrazen na obrázku 2.4.



Obrázek 2.4: Příklad tagovaného textu[7]

2.1.4 Morfologická analýza

Morfologická analýza se zabývá lineárním rozkladem slov na morfémy. V oblasti zpracování dat morfologická analýza je proces přiřazování slovům morfologických údajů včetně slovního druhu. Tyto údaje jsou obvykle reprezentovány jedním řetězcem, formát kterého se odvíjí od složitosti jazykové gramatiky.

Například pro české sloveso *být* ve formátu *je* řetězec může mít následující vzhled:

```
je ---> VpS--3n
```

Význam jednotlivých hodnot v řetězci je uveden v tabulce 3.1.

Tabulka 2.1: Formát tagu pro texty v českém jazyce [8]

Pozice	Morfologický údaj	Hodnoty údaje
1.	slovní druh	V – sloveso, P – zájmeno, N – substantivum, R – předložka, Z – interpunkce.
2.	poddruh slovního druhu	p – prézens, P – osobní zájmeno, N – apelativum.
3.	číslo	S – singulár, P – plurál
4.	jmenný rod	M – maskulinum životné, I – maskulinum neživotné. F – femininum, N – neutrum.
5.	pád	1 – nominativ, 2 – genitiv, 4 – akuzativ, 5 – vokativ.
6.	osoba	1 – první, 2 - druhá, 3 – třetí.
7.	vid	d – dokonavý, n – nedokonavý.

Nejznámějšími nástroji pro provedení morfologické analýzy anglických textů jsou GATE a Stanford Morphology tool. Pro analýzu českých textů lze použít nástroje NameTag, MorphoDiTa a Ajka.[7][11]

2.1.5 Syntaktická analýza

Syntaktická analýza či anglicky parsing zabývá se analýzou vět za účelem odvození jejich syntaktické struktury. Analýza dodává informaci o tom, jak jednotlivá slova jsou spolu propojená (viz. obr 2.5).[7]

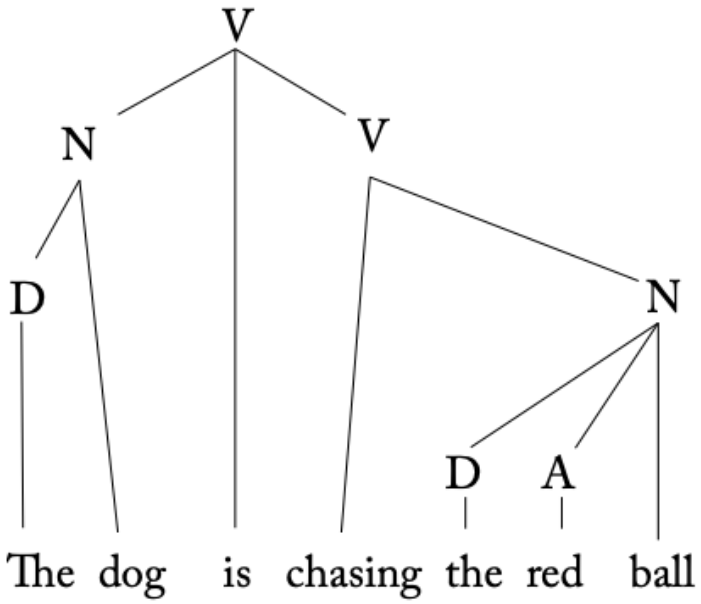
Na obrázku 2.5 je vidět, že predikátem je kombinace slov *is chaing*, předmětem je kombinace slov *the dog* reprezentovaná podstatným jménem (vyplývá to z toho, že v angličtině předmět se uvádí před predikátem). Pro větu v českém jazyce rozklad by samozřejmě nebyl tak jednoduchý, protože čeština nemá striktně definované pořadí slov, ale i přesto existují nástroje založené na strojovém učení, které umožňují rozložit věty do stromové struktury.

Nejznámějšími nástroji pro provedení syntaktické analýzy anglických textu jsou Minipar, RASP a Stanford statistical parser. Nástrojem pro provedení syntaktické analýzy českých textů je Treex od ÚFALu.[7][11]

2.2 Rozpoznávaní pojmenovaných entit

Procesu rozpoznávání pojmenovaných entit (NER) se rozumí identifikace a extrahování objektů reálného světa (např. lidí, organizací, míst).

Metody *rozpoznávaní pojmenovaných entit* textu lze rozdělit na:



Obrázek 2.5: Příklad syntaktického rozboru věty[7]

1. metody založené na **pravidlech**;
2. **slovníkové** metody;
3. metody založené na **strojovém učení**;
4. **hybridní** metody[7].

Metody založené na pravidlech používají sadu předdefinovaných pravidel k identifikaci pojmenovaných entit v textu. Tato pravidla můžou být založena na kontextu, ve kterém se entita vyskytuje nebo na vlastnostech jednotlivých tokenů (například velká písmena, pád, interpunkce). Systémy NER založené na pravidlech jsou relativně jednoduché na implementaci, ale můžou mít omezenou přesnost a pokrytí.

Slovníkové metody používají k identifikaci pojmenovaných entit předdefinovaný seznam pojmenovaných entit (například seznam názvů organizací). Tyto metody bohužel mají omezené pokrytí a můžou vyžadovat ruční úpravu slovníku.

Metody rozpoznávání pojmenovaných entit založené na strojovém učení provádí detekci a klasifikaci pojmenovaných entit na základě znalostí získaných z trénovacích dat. Samotný proces detekce NER lze popsat pomocí následujících kroků:

1. Příprava trénovacích dat, během které dochází k ručnímu zpracovaní textu, kdy všem pojmenovaným entitám se přiřazuje kategorie.
2. Extrakce lingvistických, morfologických a syntaktických charakteristik jednotlivých tokenů.

3. Trénování modelu pomocí algoritmů strojového učení, které je realizováno pomocí metod Support Vector Machine(SRV) nebo Conditional Random Field (CRF), nebo recurrent neural network (RMM), nebo Hidden Markov Model (HMM). Tyto metody na vstupu dostávají trénovací data a charakteristiky jednotlivých tokenů, a následně definují pravidla pro rozpoznávání pojmenovaných entit.

Výsledkem trénování je funkční model, který lze použit pro definování entit u textů, syntakticky a lingvisticky podobných trénovacím datům (podobných zde znamená, že pokud model se učí na anglických textech, to neznamená, že on bude fungovat u německých nebo českých textů, a to hlavně kvůli odlišným gramatickým pravidlům).

Hybridní metody využívají několik různých metod pro detekci pojmenovaných entit, což umožňuje přesnější a efektivnější kategorizací entit. Nejvyužívanějšími metody zde metody založené na strojovém učení, do kterých jsou přidány manuálně definované pravidla. Výhodou takových metod je to, že v případě nepřesnosti metody strojového učení, pro špatně označené entity lze vytvořit pravidlo a není třeba při tom připravovat další trénovací data.

2.3 Extrakce vztahů

Extrakce vztahů (RE) je proces identifikace a extrahování vztahů mezi entitami v textu.

Metody *extrakce vztahů* textu lze rozdělit na:

1. metody založené na **pravidlech**;
2. metody založené na **strojovém učení s učitelem**;
3. metody založené na **strojovém učení bez učitele**;
4. **hybridní** metody[7].

Metody založené na **pravidlech** umožňují identifikaci konkrétních vzorů v textu, které naznačují vztah mezi objekty. V takových metodách vyhledání konkrétních sekvencí je realizováno pomocí regulárních výrazů , které označují vztah (například „A je otcem B“). Tyto metody jsou snadně implementovatelný a využívají se pro práci s malými soubory dat.

Metody založené na **strojovém učení s učitelem** umožňují trénování modelu strojového učení na předzpracovaném datovém souboru textu, kde jsou vztahy mezi entitami již známé. Získaný model se následně používá k extrahování vztahů z nestrukturovaných textových dokumentů. Metody učení s učitelem jsou velmi účinným nástrojem, ale pro trénování modelu vyžadují velké množství trénovacích dat, jejichž získání může být obtížné a časově náročné.

Metody založené na **strojovém učení bez učitele** používají techniky shlukování a modelování témat k identifikaci v textu relací, které naznačují

vztah mezi entitami. Výhodou těchto metod je to, že oni nevyžadují přípravu trénovacích dat.

Hybridní metody umožňují pro extrakci vztahu mezi entitami použití kombinace výše uvedených metod. Tak lze například použít metody založené na pravidlech k identifikaci potenciálních relací a poté použít strojové učení k potvrzení či vyvrácení těchto vztahů. Tento přístup je efektivnější než použití jakékoli samostatné metody, ale také je složitější na implementaci.

Kapitola 3

Analýza datového korpusu

Datový korpus je reprezentován souborem ve formátu *.zip* obsahujícím 104 676 novinářských článků, u kterých pomocí aplikace vyvinuté Ústavem teoretické a komputační lingvistiky Karlové univerzity bylo provedeno lingvistické předzpracování.

3.0.1 Struktura dat

Každý předzpracovány článek v datovém korpusu je uložen do samostatného *.xml* souboru s unikátním názvem, který je zároveň identifikátorem článku. Tak například článek *doc-8112658.xml* má identifikátor *doc-8112658*, a proto identifikátory všech tokenů v souboru taky začínají řetězcem *doc-8112658*.

Tokenem v datovém korpusu se rozumí jednotlivá slova či interpunkce, a proto slova typu *KDU-ČSL* či *Kasym-Žomart* jsou považována za 3 tokeny. Každý token má své *id*, *lemmu* - slovo v základním tvaru, *pos* - kategorii slovních druhů, do které token spadá, *msd* - řetězec s informacemi o morfologických charakteristikách příslušných slovnímu druhu tokenu a *ana* - řetězec s zakódovanými informacemi o všech morfologických charakteristikách tokenu.

Tak na níže uvedeném příkladu token je reprezentován slovem *kancelář*. Základní tvar slova je stejný s použitým v textu. Jedná se o podstatné jméno, ve 4. pádu, ženského rodu, v jednotném čísle, které nemá negativní podtext (například na rozdíl od slova **ne**štěstí).

```
<w xml:id="doc-5399285.p13.s1.w3" lemma="kancelář" pos="NOUN"
msd="UPosTag=NOUN|Case=Acc|Gender=Fem|Number=Sing|Polarity=Pos"
ana="pdt:NNFS4-----A----">kancelář</w>
```

Tag *w* tady říká, že token je reprezentován slovem. Tokeny reprezentované interpunkcí mají tag *pc* (viz. příklad níže).

```
<pc xml:id="doc-5344047.p26.s1.w22" lemma="." pos="PUNCT"
msd="UPosTag=PUNCT" ana="pdt:Z:-------------">.</pc>
```

Některé tokeny navíc mají tag *name*, který slouží pro označování vlastních jmen (viz. příklad níže).

```
<name ana="ne:ps" xml:id="corpus-856.ne386" type="PER">
<w xml:id="doc-5344047.p17.s2.w4" lemma="Kolář" pos="PROPN"
msd="UPosTag=PROPN|Animacy=Anim|Case=Gen|Gender=Masc|
NameType=Sur|Number=Sing|Polarity=Pos"
ana="pdt:NNMS2-----A----">Koláře
</w>
</name>
```

Hodnota *ana* zde říká, o jaký druh vlastního jména se jedna. Zkratka *ne* signalizuje o tom, že byla využita dvouúrovňová hierarchická klasifikace pojmenovaných entit, všechny možné klasifikace které jsou uvedeny na obrázku 3.1.

Nejpoužívanějšími hodnotami POS tagu neboli *pos* jsou:

- *ADJ* - přídavná jména;
- *ADV* - příslovce;
- *NOUN* - podstatná jména;
- *VERB* - slovesa;
- *PROPN* - vlastní jména;
- *NUM* - číslovky;
- *PUNCT* - interpunkce;
- *CCONJ* - spojky;
- *INTJ* - citoslovce;
- *SYM* - symboly;
- *SCONJ* - podřízené spojky;
- *DET* - determinátory;
- *AUX* - pomocná slovesa (například "být");
- *ADP* - adpozice.

Základní morfologická kriteria *msd* a jejích hodnoty jsou uvedeny v tabulce 3.1.

Tabulka 3.1: Základní morfologická kriteria[9]

Název kriteria	Popis kriteria	Hodnoty kriteria
UPosTag	Slovní druh tokenu.	Viz. hodnoty *POS tag*.

NameType	Druh vlastního jména.	*Com* - organizace, *Geo* - zeměpisný název, *Giv* - jméno, *Nat* - národnost, *Pro* - značka, *Sur* - příjmení, *Oth* - ostatní.
Animacy	Životnost tokenu.	*Anim*- životný, *Inan* - neživotný.
Case	Pád.	*Nom* - 1. pád, *Gen* - 2. pád, *Dat* - 3. pád, *Acc* - 4. pád, *Voc* - 5. pád, *Loc* - 6. pád, *Ins* - 7. pád.
Gender	Rod.	*Fem* - ženský, *Masc* - mužský, *Neut* - střední.
Number	Číslo.	*Sing* - jednotné číslo, *Plur* - množné číslo.
Polarity	Charakteristika tokenu z hlediska přítomnosti negace.	*Pos* - bez negativního podtextu. *Ne* - s negativním podtextem.
Aspect	Slovesný vid.	*Imp* - nedokonavý vid, *Perf* - dokonavý vid.
Person	Osoba.	*1* - 1. osoba, *2* - 2. osoba, *3* - 3. osoba.
Abbr	Je-li token zkratkou nečeho.	*Yes* - je zkratkou, *No* - není zkratkou.

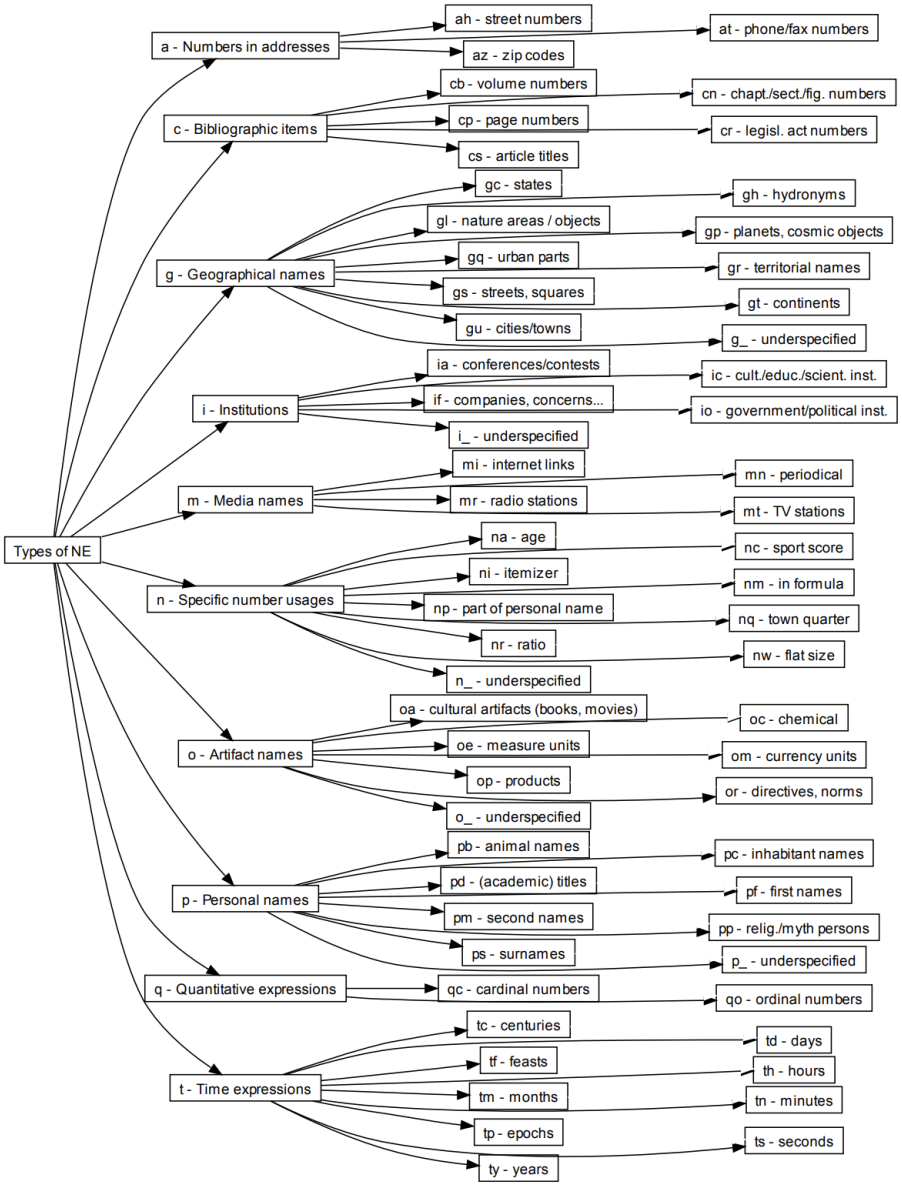
Popis jednotlivých pozic 15-mistrného morfologického řetězce *ana* je zobrazen na obrázku 3.2. Detailnější informaci ohledně každé morfologické charakteristiky lze nalézt na oficiálních stránkách ÚFALu [2]. Nicméně je vidět, že tento řetězec obsahuje stejné informace, jako *msd*, přitom forma zápisu informace v *msd* je mnohem čitelnější.

Syntaktické charakteristiky vět jsou uloženy v tagu *link*, kde řetězec *ana* jednoznačně určuje druh syntaktické relace a řetězec *target* identifikuje tokeny, ke kterým se relace vztahuje. Tak na příkladu níže tokeny *doc-5344047.p1.s2.w6* (politik) a *doc-5344047.p1.s2.w5* (český) jsou svázany relací *amod*, která slouží k dodání podrobnější informace o podstatném jméně.

```
<link ana="ud-syn:amod"
target="#doc-5344047.p1.s2.w6 #doc-5344047.p1.s2.w5"/>
```

Nejvýznamnějšími syntaktickými relacemi jsou:

- *nsubj* - relace mezi podmětem a přísudkem;
- *obj* - relace mezi přísudkem a předmětem;



Obrázek 3.1: Dvouúrovňová hierarchická klasifikace pojmenovaných entit[4]

- *ccomp* - relace, která ukazuje na předmět slovesa či přídavného jména;
- *nmod* - relace mezi podstatnými jmény;
- *root* - kořen věty;
- *flat* - relace mezi vlastními jmény či vlastními a podstatnými jmény;
- *conj* - vztah mezi dvěma slovy spojenými koordinační spojkou;
- *cc* - konjunkce dvou slov;
- *aux* - relace mezi pomocným slovesem a slovesem.

Position	Name	Description
1	POS	Part of speech
2	SubPOS	Detailed part of speech
3	Gender	Gender
4	Number	Number
5	Case	Case
6	PossGender	Possessor's gender
7	PossNumber	Possessor's number
8	Person	Person
9	Tense	Tense
10	Grade	Degree of comparison
11	Negation	Negation
12	Voice	Voice
13	Reserve1	Reserve
14	Reserve2	Reserve
15	Var	Variant, style

Obrázek 3.2: Seznam morfologických kriterií řetězce *ana*[9]

- *amod* - relace mezi podstatnými a přídavnými jmény.
- *appos* - relace, která slouží k definování, úpravě, pojmenování či popisu podstatného jména.

Ostatní hodnoty a jejich význam lze najít na oficiálních stránkách ÚFALu [10]

3.0.2 Definice entit a relací

Jak bylo uvedeno ve druhé kapitole, pro tvorbu znalostního grafu je potřeba definovat seznam entit a relací, a proto bylo rozhodnuto definovat 4 entity, pro něž je požadováno identifikovat 3 druhy relace.

První entitou je skupina reprezentující lidí. Do ní spadají všechna jména, která zazněla ve článcích. Druhou entitou je skupina všech profesí, které mají osoby z první entity. Třetí entitou je skupina všech oborů či organizací, ve kterých působí osoby z první entity. Poslední entitou je seznam všech citací, autory kterých jsou osoby z první entity.

Finální seznam všech entit vypadá následovné:

1. entita: Person
2. entita: Occupation
3. entita: Organization or Field of work
4. entita: Quote

Seznam relací vypadá následovné:

1. Person **JE** Occupation

```
    Jan "je" doktor
```

2. Occupation **PRACUJE V/NA** Organization or Field of work

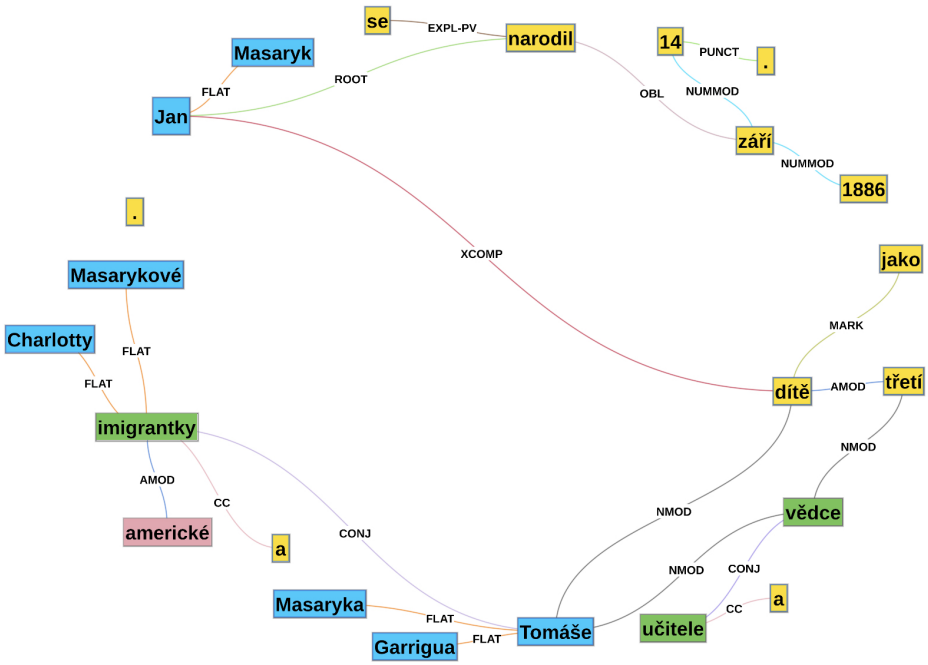
```
doktor "pracuje v" nemocnice
```

3. Person **PŘÍSUDEK Z VETY** Quote

```
Jan "řekl" "Rakovina je diagnózou"
```

3.0.3 Návrh postupu zpracování dat

Abychom mohli navrhnout postup zpracování dat a následně jejích vizualizaci, musíme detailněji prozkoumat strukturu článků a podívat se na struktury jednotlivých vět. Tak na obrázku 3.3 je ilustrován syntaktický rozbor věty z příspěvku o prezidentovi J. Masarikovi.



Obrázek 3.3: Syntaktický rozbor věty

Jelikož nás zajímají pouze objekty spadající do čtyrech entit, slova ve vetě byly obarveny podle toho, do jaké entity by se je dalo zařadit. Modrou barvou

byla označena vlastní jména spadající do entity *Person*. Zelenou barvou byla označena slova identifikující potenciální povolání a tím pádem spadající do entity *Occupation*. Růžovou barvou byla označena slova charakterizující potenciální povolaní a taky spadající do entity *Occupation*. Ostatní slova nespadající do žádné ze čtyřech entit byla obarvena žlutou barvou.

Na základě rozboru věty a kontroly slovních druhů lze udělat předpoklad, že:

1. Vlastní jména lze identifikovat pomocí relace *flat*, která existuje mezi:
   a. Jménem a příjmením:

```
1) Jan   Masaryk
   PROPN PROPN
```

   b. Jménem a druhým jménem či jménem a příjmením:

```
1) Tomáš Garrigue
   PROPN PROPN
2) Tomáš Masaryk
   PROPN PROPN
```

   c. Jménem a potenciálním povoláním:

```
1) imigrantka Charlotta
   NOUN       PROPN
2) imigrantka Masaryková
   PROPN      PROPN
```

2. Povolaní lze identifikovat pomocí relací *nmod*, *conj* a *amod*, které existují mezi:
   a. Vlastními a podstatnými jmény (přitom slova musí být ve stejném pádu a mít stejný rod):

```
1) Relace nmod:
   Tomáš vědec
   PROPN NOUN
```

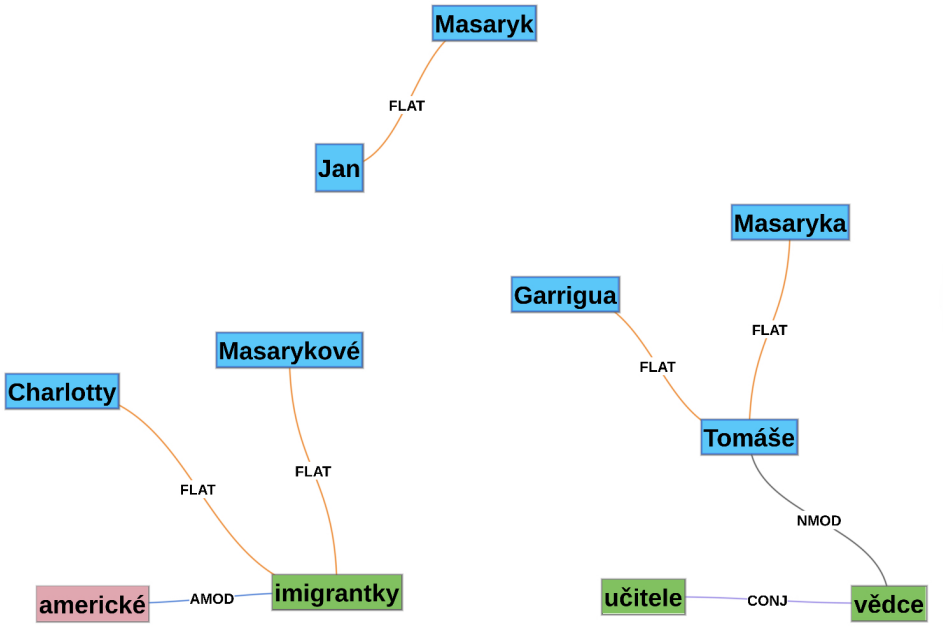
   b. Podstatnými jmény (přitom slova musí být ve stejném pádu a mít stejný rod):

```
1) Relace conj:
   vědec učitel
   NOUN  NOUN
```

c. Podstatnými a přídavnými jmény:

```
1) Relace amod:
   americká imigrantka
   ADJ     NOUN
```

Graf, ve kterým by se vyskytovala jenom slova z požadovaných entit, která jsou mezi sebou spojena pomocí relací *flat*, *nmod*, *conj* a *amod*, je uveden na obrázku 3.4.

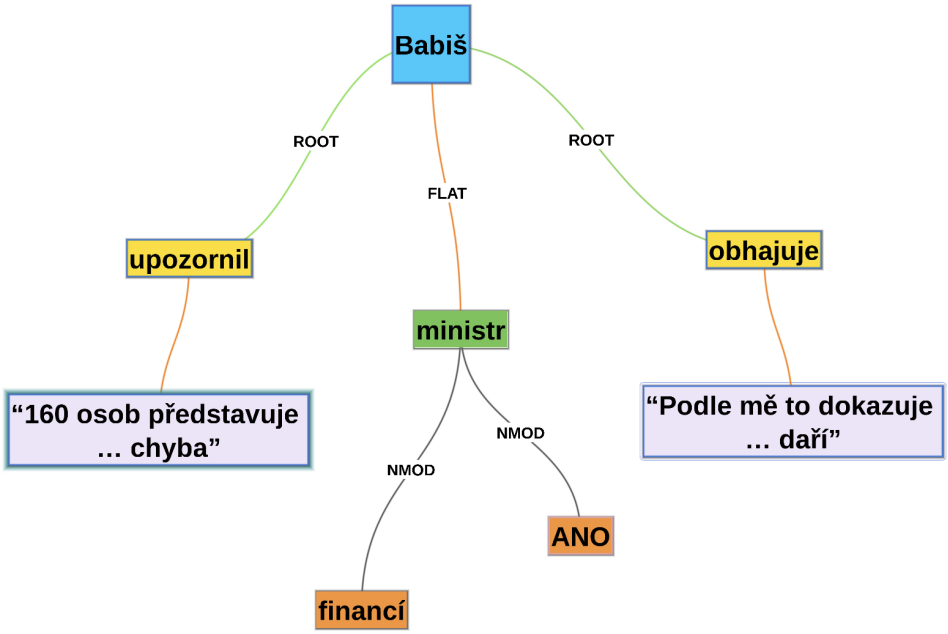


Obrázek 3.4: Redukovaný graf věty

Abychom se ujistili v předpokladu, že pro identifikaci entit je potřeba dohledávat relace *flat*, *nmod*, *conj* a *amod*, rozebereme ještě pár vět. Na obrázku 3.5 je vidět syntaktický rozbor věty o A. Babišovi. Pro lepší orientaci z grafu byla vymazaná slova, která nemají žádný vztah ke slovům patřícím do definovaných dříve entit.

Jak je zřejmě z obrázku 3.5:

1. Vlastní jméno a povolaní lze identifikovat pomocí relace *flat*, která existuje mezi jménem a povoláním:



Obrázek 3.5: Syntaktický rozbor věty o A. Babišovi

```
ministr Babiš
NOUN    PROPN
```

2. Dodatečnou charakteristiku povolaní lze získat pomocí relace *nmod*, která existuje mezi povoláním a podstatným jménem ve 2. pádu:

```
ministr financí
NOUN    NOUN
```

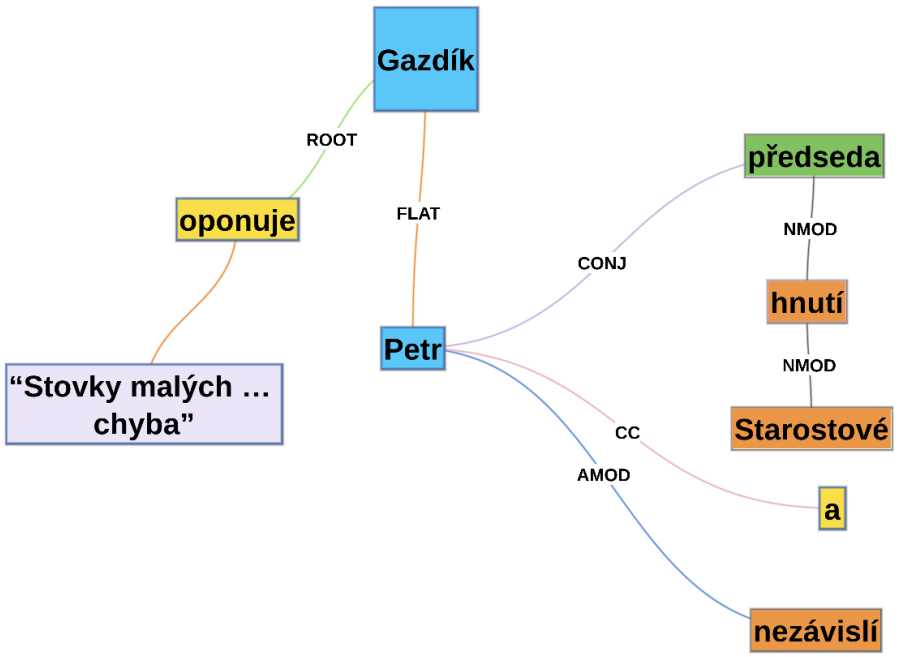
3. Organizaci neboli směr působení lze získat pomocí relace *nmod*, která existuje mezi povoláním a zkratkou se slovním druhem *PROPN* neboli podstatným jménem ve 2. pádu:

```
ministr ANO
NOUN    PROPN
```

4. Přímou řeč neboli citace lze identifikovat pomocí počítaní interpunkčních znamének. Všechno, co se nachází mezi uvozovky, je citací. Autora citace lze stanovit pomocí relace *root*. To znamená, že stačí najít kořenové sloveso věty, ověřit existenci jakékoliv relace mezi slovesem a slovy v

citaci, a následně zkontrolovat, vyskytuje-li se ve větě nějaké vlastní jméno vztahující se ke kořenovému slovesu.

Syntaktický rozbor ještě jedné věty je uveden na obrázku 3.6.



Obrázek 3.6: Syntaktický rozbor věty o P. Gazdíkovi

1. Zde vlastní jména lze identifikovat pomocí relace *flat*, která existuje mezi jménem a příjmením:

```
Pert  Gazdík
PROPN PROPN
```

2. Povolaní lze získat pomocí relace *conj*, která existuje mezi podstatným jménem a jménem osoby:

```
předseda Petr
NOUN     PROPN
```

3. Organizaci neboli směr působení lze získat pomocí relace *nmod*, která existuje mezi povoláním a podstatným jménem ve 2. pádu:

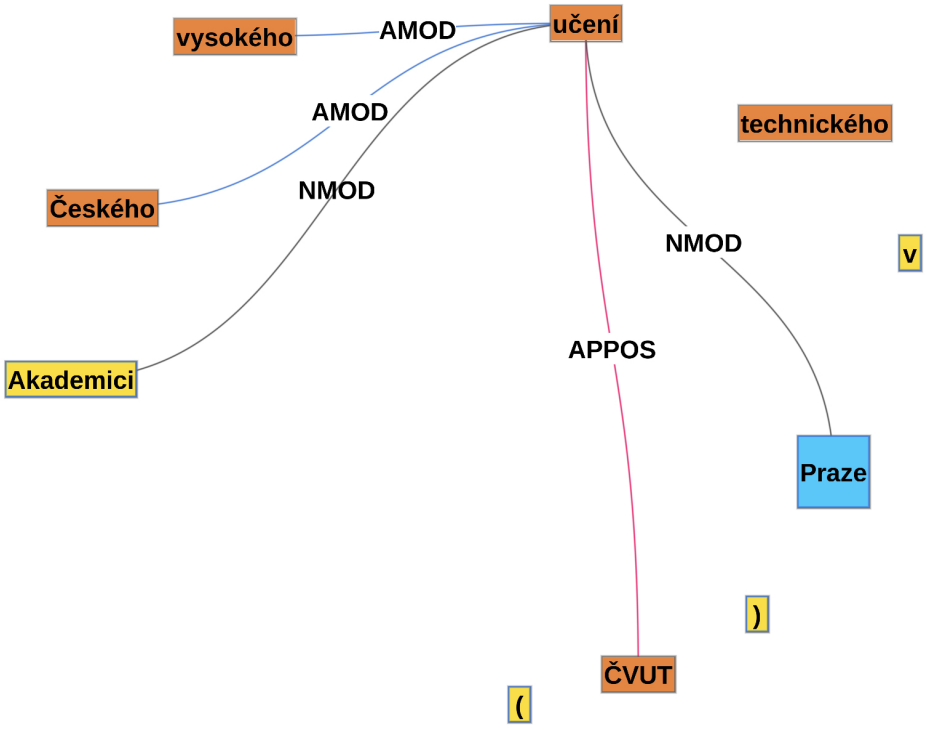
```
   předseda hnutí
   NOUN     NOUN
```

4. Dodatečnou informaci o organizace neboli směru působení lze získat pomocí relací *nmod* a *amod*, které existuji mezi podstatnými jmény, jedno ze kterých je ve 2. pádu, či jménem osoby a podstatným jménem ve 2. pádu:

```
1) hnutí Starostové
   NOUN  NOUN
2) Petr   nezávislí
   PROPN  NOUN
```

5. Citace lze identifikovat stejným způsobem, jako u předchozí věty.

Syntaktický rozbor části věty, ve které se vyskytuje zkratka je uveden na obrázku 3.7.



Obrázek 3.7: Syntaktický rozbor věty se zkratkou

Z analýzy grafu věty vyplývá, že pokud se ve větě vyskytuje název organizace a zároveň zkratka názvu, je jednodušší dohledat název organizace pomocí relace *nmod* a následně skrz relaci *appos*:

```
1) Relace nmod:
   akademici učení
   NOUN      NOUN
2) Relace appos:
   učení  ČVUT
   NOUN   PROPN
```

Jak je vidět z rozboru vět, slova spadající do definovaných entit lze identifikovat pomocí relací *flat*, *nmod*, *amod*, *conj* a *appos*, a proto získat entity a relace lze pomocí pravidel založených na morfologických a syntaktických znalostech o slovech a vetách. Definice a implementace pravidel i byly provedený v následující kapitole.

Kapitola 4

Implementace metody pro vytvoření znalostních grafů

4.0.1 Volba software

Pro návrh metod NER a RE založených na pravidlech byl zvolen programovací jazýk Python. Python je dynamický, objektově orientovaný, interpretovaný programovací jazyk, který se aktivně používá pro tvorbu rozsáhlých, plnohodnotných aplikací. Python je vyvíjen jako open source projekt a poskytuje bezplatné instalační balíčky pro většinu běžných platforem. Charakteristickým rysem jazyka Python je produktivita z hlediska rychlosti psaní programů a snadné použití široké škály knihovních modulů, které usnadňují řešení různých úloh.[6]

Pro vizualizaci znalostního grafu byla zvolena grafová NoSQL databáze Neo4j Desktop. Grafová databáze je systém na ukládání a zpracování dat v podobě grafu, kde graf je datovou strukturou skládající se z vrcholů, hran a vlastností. "Dotazování v Neo4j je uskutečňováno prostřednictvím procházení grafu od uzlu, který je předem určen jako startovní. Neo4j má tři mechanismy na procházení grafů, a to jsou dotazovací jazyky Cypher, Gremlin a vestavěné rozhraní Java API."[5]

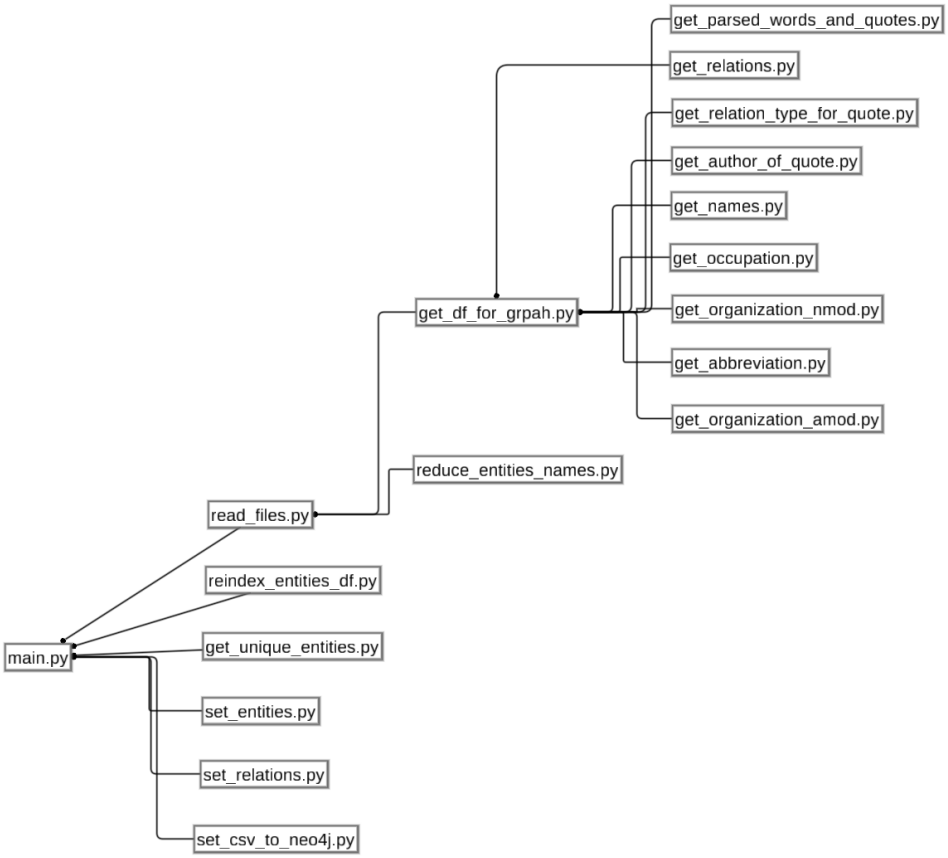
4.0.2 Popis navržené implementace

Hlavní program je reprezentován souborem *main.py*. Schematické znázornění struktury programu je uvedeno na obrázku 4.1.

Zpracování souboru začíná v metodě *get_parsed_words_and_quotes*, která:

1. Prochází soubor token po tokenu a u toho počítá, kolik krát došlo k vyskytnutí uvozovek.

2. Pokud uvozovky se vyskytly n krát, kde n je sudé číslo, program zpracovává každý token a ukládá do slovníku informace o ID slova, jeho základní formě a morfolofických vlastnostech. Pokud token je zároveň vlastním jménem či podstatným jménem s hodnotou *Sur* (příjmení) v atributu *NameType*, token se ukládá do tabulky entit.



Obrázek 4.1: Schematické znázornění struktury programu

3. Pokud uvozovky se vyskytly k krát, kde k je liché číslo, program začíná ukládat tokeny do jednoho řetězce a následně ukládá získaný řetězec do slovníku s citacemi, do kterého taky ukládá nově vygenerovaný index citace.

4. Na výstupu metoda vrací slovník citací, slovník s informacemi o všech tokenech, které nejsou součástí citací, a tabulku s informacemi o všech vlastních jménech, které taky nejsou součástí citací.

Tabulka entit má sloupce *Name_nominative (x2), Name_id (x2), Surname_nominative (x1), Surname_id (x1), Gender (x1), Occupation_nominative (x3), Org_nominative (x3), Org_amod_nominative (x9), , Occupation_id (x3), Org_id (x3), Org_amod_id (x9)*. Taková volba počtu sloupců umožňuje u jedné osoby uložit 3 záznamu o povolaní a 3 záznamu o oboru působení či organizaci.

Zpracovaní dat dále pokračuje v metodě *get_relations*, která prochází všechny relace a ukládá informaci o nich do slovníku.

Následně se spouští metoda *get_relation_type_for_quote*, která prochází všechny věty, v nichž se vyskytly citace a hledá hlavní sloveso ve větě pomocí

relace *ROOT*. Dělá se to protože ve vetech s citacemi autorem citace je podmět, který je spojen s kořenovým slovesem věty. A proto nalezení kořenového slovesa umožňuje následné dohledaní podmětu pomocí metody *get_author_of_quote* a relace *nsubj*. V případě, že ve větě chybí relace *nsubj*, metoda hledá podstatné jméno v 1. pádu. Pokud ve větě není podstatné jméno v 1. pádu či jejích je několik a není možní určit, komu přesně patří citace, informace o autorovi v tabulce zůstává prázdna.

Struktura tabulky citací je zobrazena na obrázku 4.2

Quotes_text	Quotes_id	Date	Relations_id	Relations_word	Authors_id
"Tyto materiály dokreslují atmosféru do...	doc-5347793.p18.s1_12	2017-04-04	doc-5347793.p18.s1.w24	argumentovat	doc-5347793.p13.s2.w17
"Rodinní příslušníci vesnických boháčů ...	doc-5347793.p22.s1_16	2017-04-04	doc-5347793.p22.s1.w49	argumentovat	doc-5347793.p13.s2.w17
"Vy jste ale udělala opak. Do země jste...	doc-5348677.p19.s1_5	2017-04-24	doc-5348677.p19.s1.w4	citelný	doc-5348677.p19.s1.w2
"Čestné členství zpravidla dostávají as...	doc-5371587.p4.s5_3	2021-01-18	doc-5371587.p4.s5.w17	dodat	doc-5371587.p4.s5.w18
"Možnost útěku podle soudu nebyla důvod...	doc-5371817.p4.s4_2	2017-04-11	doc-5371817.p4.s4.w40	dodat	doc-5371817.p7.s4.w16
"Paní ministryně Válková byla mezitím o...	doc-5347552.p5.s2_4	2017-04-06	doc-5347552.p5.s2.w15	dodat	doc-5347552.p5.s2.w16
"Nedovolíme útoky proti náboženskému cí...	doc-5346699.p6.s2_4	2017-04-07	doc-5346699.p6.s2.w11	dodat	doc-5346699.p6.s2.w12
"Je to pro mě prodloužený víkend,"	doc-5358317.p3.s3_2	2017-12-01	doc-5358317.p3.s3.w10	dodávat	doc-5358317.p12.s3.w16
"Jednou z prioritních oblastí bude i ot...	doc-5347419.p8.s2_1	2017-07-09	doc-5347419.p8.s2.w15	dodávat	doc-5347419.p8.s2.w16
"Statistiky ukazují, že rekreační běh a...	doc-5399159.p6.s2_2	2017-04-18	doc-5399159.p6.s2.w29	dodávat	doc-5399159.p6.s2.w30
"V tomhle případě to není žádná překážk...	doc-5348851.p15.s1_6	2017-04-15	doc-5348851.p15.s1.w5	dodávat	doc-5348851.p15.s1.w3
"Neexistuje nic jako data zadarmo. Jen ...	doc-5399403.p24.s3_10	2017-04-24	doc-5399403.p24.s3.w9	doplnit	doc-5399403.p30.s2.w19
"Je celkem žádoucí, aby v dozorčí rad...	doc-5348927.p30.s3_15	2017-04-26	doc-5348927.p30.s3.w22	doplnit	doc-5348927.p26.s3.w25

Obrázek 4.2: Tabulka citací

Po přípravě tabulky s informacemi o citacích začíná doplňovaní informací u tabulky entit. Pomocí metody *get_names* probíhá získávaní jmen pro všechna příjmení dohledaná v první metodě. Jelikož jména jsou spojená s příjmeními pomocí relace *flat*, která taky umožňuje dohledávaní povolaní, metoda *get_names*:

1. Na začátku získává všechny relace typu *Příjmení -> Jméno* či *Příjmení -> Povolaní*.
2. Následně metoda získává všechny relace typu *Jméno -> Povolaní* či *Jméno -> Druhé jméno* či příjmení.
3. Na konci metoda kontroluje, je-li pro povolaní ve vztahu *Příjmení -> Povolaní* nalezeném v prvním kroku existuje relace *Povolaní -> Jméno*.

Dále probíhá dohledávaní povolání pomocí metody *get_occupation*, která:

1. Za prvé, získává všechny relace typu *Jméno -> Povolaní* druhu *nmod*.
2. Následně metoda získává všechny relace typu *Jméno -> Povolaní* či *První povolaní -> Druhé povolaní* druhu *conj*, u kterých tokeny jsou ve stejném pádu.

Potom se spouští metoda *get_organization_nmod*, která slouží pro získávání všech pracovních oborů a organizací. Tato metoda dohledává všechny relace *Povolaní -> Společnost/obor* druhu *nmod*. Dohledaná data se zapisují do tabulky pouze v případě, že potenciální název společností je ve 2. pádu a není geografickým objektem.

Dále pomocí funkce *get_abbreviation* na základě existence relace druhu *appos* se ověřuje, jestli pro označení oboru/organizace neexistuje zkratka.

V případě existence zkratky název oboru se nahrazuje zkratkou. Následně pomocí metody *get_organization_amod* dohledávají se upřesňující informace o organizacích či oborech.

V dalším kroku v metodě *get_df_for_grpah* probíhá kontrola ID uvedených v tabulce citací. ID autorů citací jsou porovnány s ID jmen či druhých jmen osob vyskytujících se v tabulce entit. V případě, že ID je nalezeno (například autorem je osoba s ID xxx a jménem Jan), ono je nahrazeno ID příjmení autoru (autorem je osoba s ID yyy a příjmením Novák). Toto se dělá proto, aby později bylo možné propojit autory s citacemi.

Následně v metodě *reduce_entities_names* probíhá redukce nevyužitých sloupců a sjednocení osob na základě stejného jména, příjmení, profese a oblastí působení.

Dále pomocí metody *reindex_entities_df* se provádí spojení jména, druhého jména a příjmení, a taky reindexace všech povolání a oborů působení či organizací. Toto se dělá z toho důvodu, že stejné organizace a povolaní mají různé ID, které jim byly přidělány na základě ID článků, ve nichž se tyto tokeny vyskytovaly, a proto za účelem normalizace staré ID se nahrazují nově vygenerovanými.

Následně v metodě *get_unique_entities* dochází k reindexace osob. Účelem toho je to, že v tabulce entit se vyskytují osoby, pro které existuje několik záznamů. Tak například na obrázku 4.3 je vidět, že existují 3 záznamy pro osobu Abott neboli Abbott. Takové záznamy je potřeba sjednotit. Navíc v souvislostí s reindexací osob dochází taky v metodě k upravení ID autoru u citací.

Surname_nominative	Occupation_nominative1	Org_nominative1	Gender	Surname_id
Abandoned	None	None	Masc	doc-5347747.p3.s3.w11 doc-5347747.p3.s3.w11 doc-5347747.p3.s3.w11
Abbott	None	None	Masc	doc-5347395.p2.s2.w3 doc-5347395.p2.s4.w7 doc-5347395.p7.s5....
Abott	None	None	Masc	doc-5347395.p8.s1.w3 doc-5347395.p1.s1.w10 doc-5347395.p8.s1...
Abott	baseballista	None	Masc	doc-5347395.p1.s1.w10

Obrázek 4.3: Tabulka entit

Výstupem metody *get_unique_entities* je dvojice tabulek entit a citací, jejichž struktury jsou zobrazeny na obrázcích 4.4 a 4.5.

Dále pro vytvoření entit v grafové databázi používá se metoda *set_entities*, která na základě dat z tabulek entit a citací připravuje jednotlivé *.csv* soubory určené pro jejich následný import do Neo4j.

Vytvoření relací probíhá pomocí metody *set_relations*, která jako základ taky používá tabulky entit a citací. Citace, autora kterých nebylo možné stanovit, jsou taky nahrávány do databáze, ale žádné relace pro ně nejsou vytvořené, jelikož autorem je osoba, u které nebylo možné dohledat jméno.

Načítaní dat z jednotlivých *.csv* souborů do databáze Neo4j je realizováno pomocí metody *set_csv_to_neo4j*, ve které se provádí navázaní spojení s databází a pomocí funkcí knihovny APOC dochází k importu dat.

Surname_nominative	Occupation_nominative1	Org_nominative1	Gender	Surname_id	Occupation_id1	Org_id1
Pauline Grasová	mluvčí	ACM	Fem	per370	prof59	org49
Jaroslav Faltýnek	místopředseda	ANO	Masc	per270	prof16	org15
Andrej Babiš	lídr	ANO	Masc	per196	prof43	org15
Andrej Babiš	šéf	ANO	Masc	per196	prof26	org15
Brad Marchand	střelec	Bruin	Masc	per213	prof10	org19
Trump	šéf	CIA	Masc	per156	prof26	org9
Lea Michalová	analytička	Median	Fem	per313	prof91	org39
Miroslav Kalousek	šéf	TOP	Masc	per357	prof26	org46
Eibl	analytik	Transparency	Masc	per37	prof11	org2
Matthias Müller	ředitel	VW	Masc	per341	prof37	org45
Václav Hanzlík	mluvčí	Viktoria	Masc	per431	prof59	org64
Jiří Grunda	ředitel	asociace	Masc	per282	prof37	org10
Vildumetzová	předsedkyně	asociace	Fem	per165	prof38	org10
Pavel Suchan	tajemník	astronomický český společnost	Masc	per375	prof104	org51
Roman Havlín	ředitel	bezpečnost	Masc	per394	prof37	org56
Jiří Dušek	ředitel	brněnský hvězdárna	Masc	per280	prof37	org34
Filip Poňuchálek	mluvčí	brněnský magistrát	Masc	per244	prof59	org26
Sean Spicer	mluvčí	bílý dům	Masc	per404	prof59	org59
Jiří Rusnok	guvernér	centrální banka	Masc	per289	prof28	org35
Stanislav Zíma	mluvčí	cestovní kancelář	Masc	per413	prof59	org62
Christopher Stevens	pracovník	diplomatický mise	Masc	per218	prof57	org21
Roman Herden	mluvčí	dopravní společnost	Masc	per395	prof59	org57

Obrázek 4.4: Tabulka entit

Quotes_text	Quotes_id	Date	Relations_id	Relations_word	Authors_id
"Tyto materiály dokreslují atmosféru do...	doc-5347793.p18.s1_12	2017-04-04	doc-5347793.p18.s1.w24	argumentovat	per30
"Rodinní příslušníci vesnických boháčů ...	doc-5347793.p22.s1_16	2017-04-04	doc-5347793.p22.s1.w49	argumentovat	per30
"Vy jste ale udělala opak. Do země jste...	doc-5348677.p19.s1_5	2017-04-24	doc-5348677.p19.s1.w4	citelný	doc-5348677.p19.s1.w2
"Čestné členství zpravidla dostávají as...	doc-5371587.p4.s5_3	2021-01-18	doc-5371587.p4.s5.w17	dodat	per33
"Možnost útěku podle soudu nebyla důvod...	doc-5371817.p4.s4_2	2017-04-11	doc-5371817.p4.s4.w40	dodat	per46
"Paní ministryně Válková byla mezitím o...	doc-5347552.p5.s2_4	2017-04-06	doc-5347552.p5.s2.w15	dodat	per177
"Nedovolíme útoky proti náboženskému cí...	doc-5346699.p6.s2_4	2017-04-07	doc-5346699.p6.s2.w11	dodat	per114
"Je to pro mě prodloužený víkend,"	doc-5358317.p3.s3_2	2017-12-01	doc-5358317.p3.s3.w10	dodávat	per101
"Jednou z prioritních oblastí bude i ot...	doc-5347419.p8.s2_1	2017-07-09	doc-5347419.p8.s2.w15	dodávat	doc-5347419.p8.s2.w16
"Statistiky ukazují, že rekreační běh a...	doc-5399159.p6.s2_2	2017-04-18	doc-5399159.p6.s2.w29	dodávat	per5
"V tomhle případě to není žádná překážk...	doc-5348851.p15.s1_6	2017-04-15	doc-5348851.p15.s1.w5	dodávat	per424
"Neexistuje nic jako data zadarmo. Jen ...	doc-5399403.p24.s3_10	2017-04-24	doc-5399403.p24.s3.w9	doplnit	per282
"Je celkem žádoucí, aby v dozorčí rad...	doc-5348927.p30.s3_15	2017-04-26	doc-5348927.p30.s3.w22	doplnit	per37

Obrázek 4.5: Tabulka citací

Kapitola 5

Diskuze získaných výsledků

5.0.1 Analýza znalostního grafu

Po ukončení programu do znalostního grafu bylo importováno:

1. 149 913 objektů entity *Person*
2. 149 698 objektů entity *Person ID* (pomocná entita vyžitá pro vytvoření grafu v Neo4j)
3. 6 317 objektů entity *Occupation*
4. 4 586 objektů entity *Organization or Field of work*
5. 295 443 objektů entity *Quote*
6. 138 670 relací *JE*
7. 13 096 relací *PRACUJE V/NA*
8. 9 056 relací *MA* (pomocná relace mezi entity *Occupation* a *Organization or Field of work* vyžitá pro vytvoření grafu v Neo4j)
9. 254 630 relací propojujících citace a jejích autora

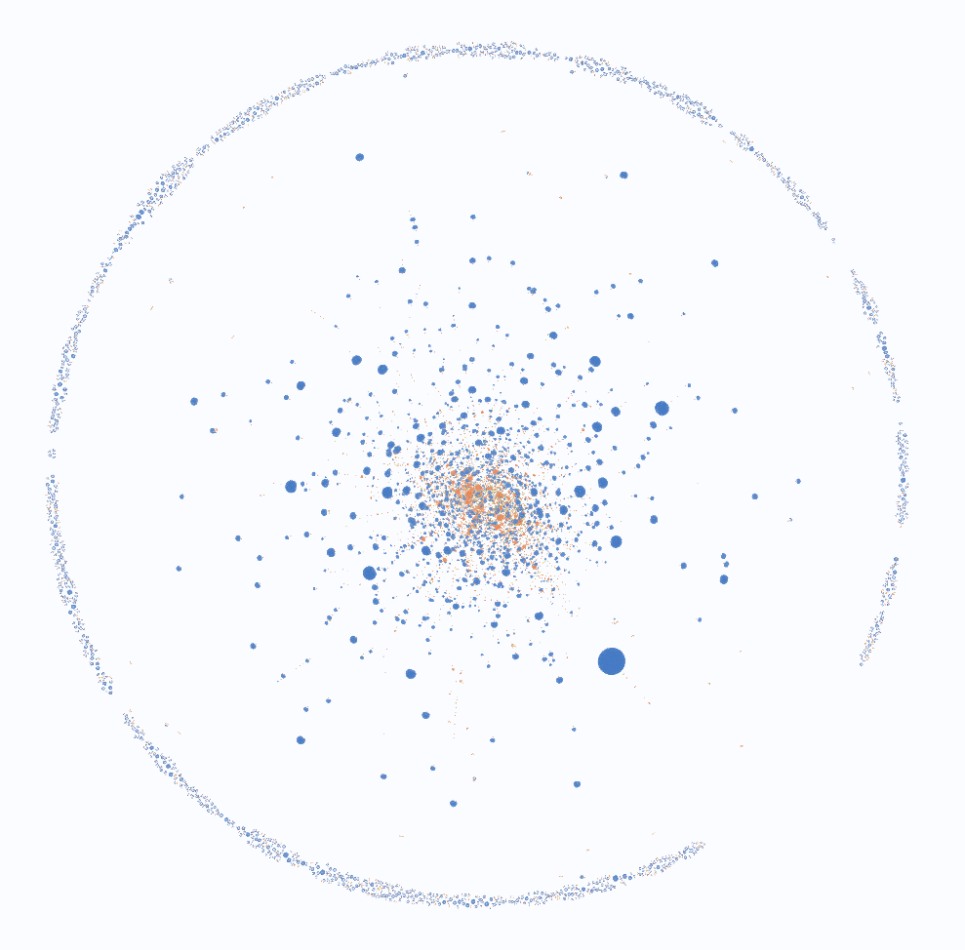
Vytvořený znalostní graf je zobrazen na obrázku5.1.

Jelikož graf obsahuje velké množství uzlů a relací, během jednotlivých dotazů zkusíme se zeptat databáze na konkretní věcí.

- Na začátku zkusíme zobrazit relaci *Person -> Occupation* pro osoby s příjmením "Hamáček".

Vygenerovaný graf zobrazený na obrázku 5.2 má 7 uzlů z entity *Person* a 84 uzlů z entity *Occupation*. Většina uzlů z entity *Occupation* opravdu definuje povolání, nicméně vyskytují se tam i uzly typu *třena* nebo *kamufláž*, které tam být nemají. Toto signalizuje o tom, že stanovené podmínky pro vztah *Person -> Occupation* nejsou dostatečně striktní.

- Dále zkusíme zobrazit relaci *Person -> Quotes* pro osoby s příjmením "Hamáček", u kterých není uvedeno jméno.

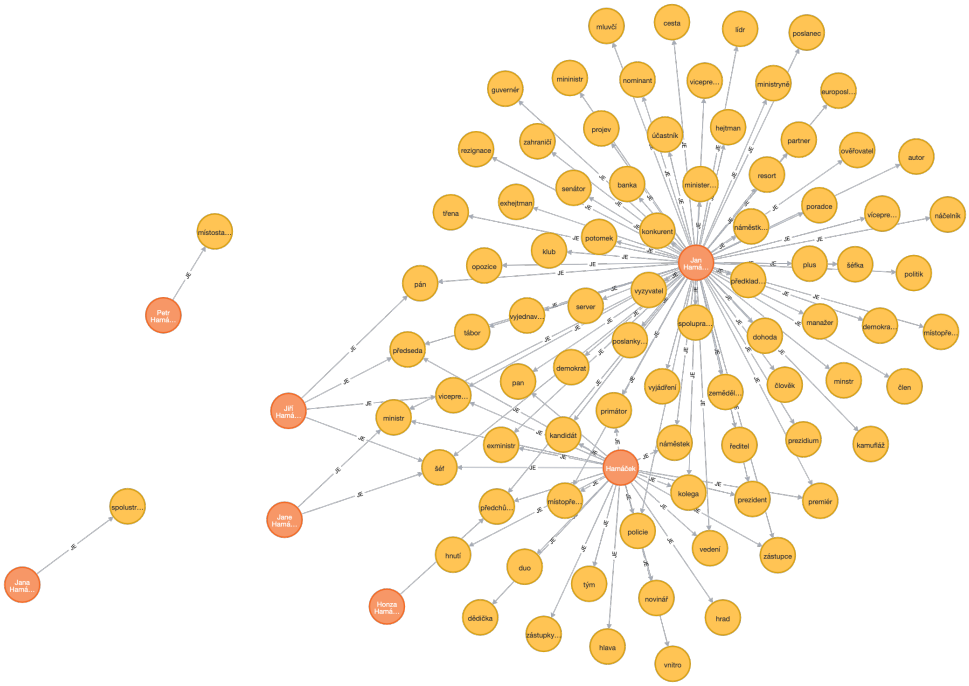


Obrázek 5.1: Vytvořený znalostní graf

Vygenerovaný graf zobrazený na obrázku 5.3 má 1 uzel z entity *Person* a 1424 uzlů z entity *Quotes*. To znamená, že v grafu existuje jenom jeden záznam o nějakém Hamáčkovi, jehož přesnou identitu není možné stanovit díku chybějícímu jménu. Jinak žádné problémy či nedostatky pro tento typ grafu se tu nevyskytují.

- Jako příklad grafu zobrazujícího několik druhu relací zobrazíme síť uzlů, kterou lze popsat vztahem *Person -> Organization -> Occupation*. Pro osoby v entitě *Person* znovu nastavíme podmínku, že jejích příjmení musí být "Hamáček", jméno nemusí být uvedeno.

Vygenerovaný graf zobrazený na obrázku 5.4 má 1 uzel z entity *Person*, 6 uzlů z entity *Organization or Field of work* a 15 uzlů z entity *Occupation*. Při jeho lepším zkoumání lze identifikovat 2 zbytečné uzly v entitě *Organization or Field of work* a 1 uzel v entitě *Occupation*. Toto poukazuje na to, že pravidla pro relaci *nmod* umožňují se do seznamu organizací dostat slovám, identifikujícím povolání.



Obrázek 5.2: Graf *Person -> Occupation* pro příjmení "Hamáček"

- Dále zobrazíme relaci *Person -> Occupation* pro entity s jménem "кирилл попутников".

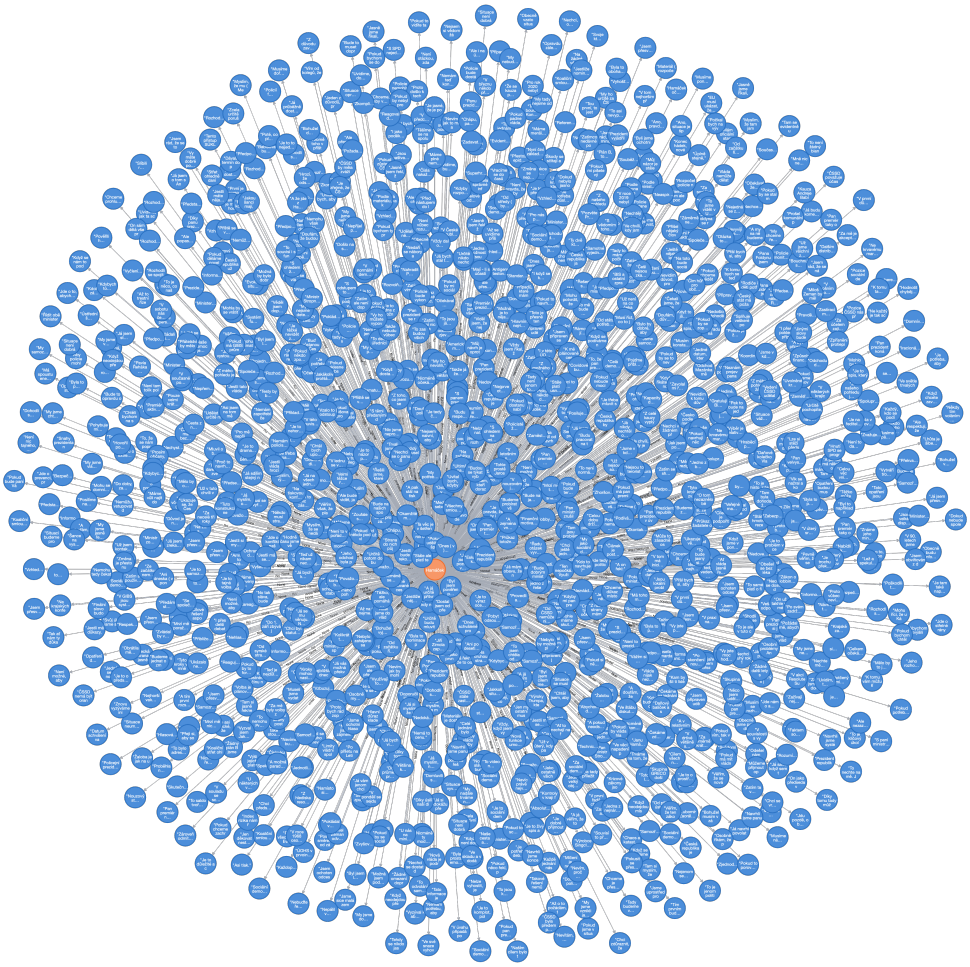
Vygenerovaný graf zobrazený na obrázku 5.5 ukazuje, že navržená metoda dokáže identifikovat jména ve všech jazycích.

- Zkusíme taky zobrazit relaci *Personal ID -> Person -> Organization -> Occupation* pro entity obsahující v jménu slovo "Hamáček".

Vygenerovaný graf zobrazený na obrázku 5.6 ukazuje, že všechny záznamy o osobě s příjmením "Hamáček"a prázdným jménem skutečně reprezentují jednu a tutež osobu.

- Taky pro ukázku funkcí Neo4j v aplikace Charts vytvoříme graf demonstrující počet citací v roce v závislosti na pohlaví autora.

Vygenerovaný graf zobrazený na obrázku 5.7 ukazuje, že muži produkují větší množství citací než ženy a že množství citací ve článcích je víceméně stejné a nárůsty počtu citací můžou být způsobeny volbami do Evropského parlamentu v roce 2019 a lockdownem v roce 2021.



Obrázek 5.3: Graf *Person -> Quotes* pro příjmení "Hamáček"

5.0.2 Nedostatky a problémy navržené metody

Při analýze získaných tabulek entit a citací a pozorování chování software bylo nalezeno několik problémů, které do značné míry ovlivnily získané výsledky. Seznam těchto problémů je uveden níže.

- Špatná detekce entit a relací kvůli chybám ve vstupních datech.

V předzpracovaných datech vyskytují se různorodé chyby, které způsobují jak nekorektní identifikaci entit, tak i tvorbu zbytečných a nesmyslných vztahů.

Tak například zkratka *ANO* jednou je identifikována jako *PROPN* (jméno) s parametrem *Animacy=Anim*, po druhé je identifikována jako *NOUN* (viz. příklad níže), což znamená, že v některých případech zkratka byla identifikována jako jméno.

```
w xml:id="doc-5348927.p1.s2.w9" lemma="Ano" pos="PROPN"
msd="UPosTag=PROPN|Animacy=Anim|Case=Dat|
```

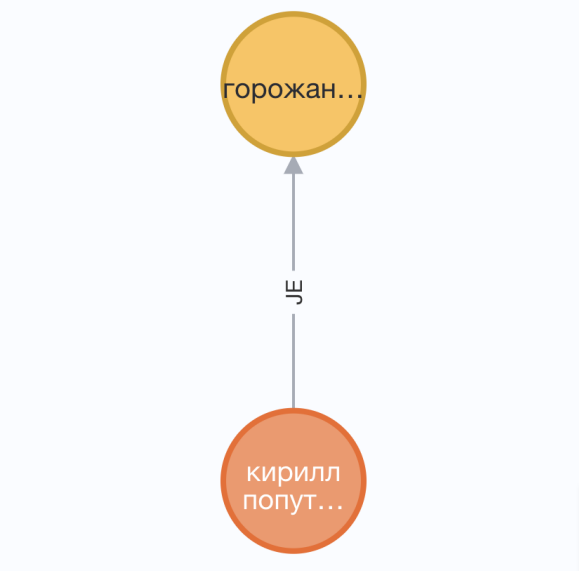
Obrázek 5.4: Graf *Person -> Organization -> Occupation* pro příjmení "Hamáček"

```
Gender=Masc|NameType=Giv|...
w xml:id="doc-8496981.p1.s1.w4" lemma="ano" pos="NOUN"
msd="UPosTag=NOUN"...
```

Stejně tak slovo *kurs* ve článku je identifikováno jako vlastní jméno, které na základě morfologických charakteristik je chápáno programem jako osoba:

```
w xml:id="doc-5347546.p12.s2.w1" lemma="kurs" pos="PROPN"
msd="UPosTag=PROPN|Animacy=Anim|Case=Nom|Gender=Masc|
NameType=Sur|Number=Sing|Polarity=Pos"...
```

Taky v textech se vyskytují tokeny typu *0.kohlschreiber* nebo *2.bennett*, které skutečně jsou vlastními jmény, ale to, že při tokenizaci došlo k chybě, následně neumožňuje správnou identifikaci osob.



Obrázek 5.5: Graf *Person -> Occupation* pro jméno "кирилл попутников"

Napsaní příjmení některých lidí se liší v závislosti na autorovi článku. Tak například ve článku *doc-7790358* se vyskytuje jméno Preben Aaman, ale ve článku *doc-6682194* toto příjmení se píše se dvěma n na konci - Preben Aamann.

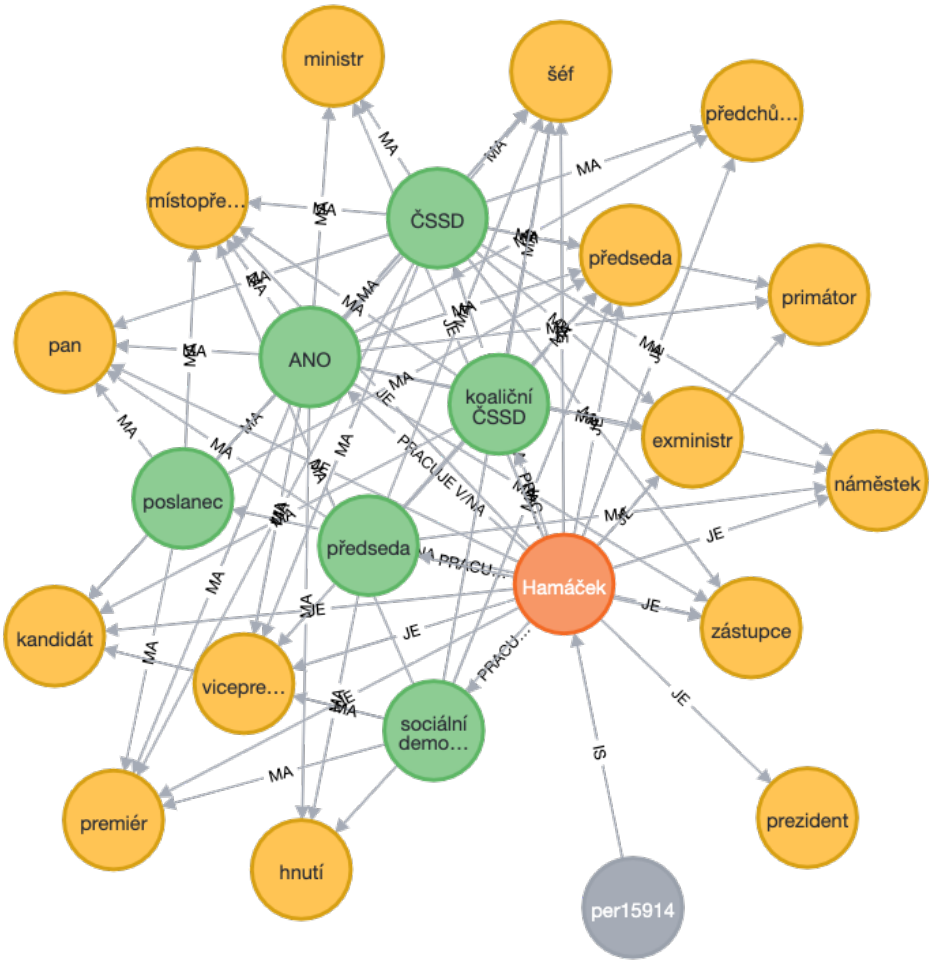
Díky identifikaci povolaní na základě relací a morfologických charakteristik, do entity *Occupation* spadá hodně slov, které by tam neměly být. Tak v níže uvedené větě autor nazývá Gawlasa talentem. Kvůli tomu, že slovo *talent* bylo ohodnoceno jako životní (viz morfologickou charakteristiku slova níže), relace *Gawlas -> talent* byla uložena do souboru s entitami.

```
Teď se český talent Gawlas probojoval mezi šipkařskou smetánku
w xml:id="doc-8429882.p1.s2.w4" lemma="talent" pos="NOUN"
↪  msd="UPosTag=NOUN|Animacy=Anim|Case=Nom|Gender=Masc|
Number=Sing|Polarity=Pos"
```

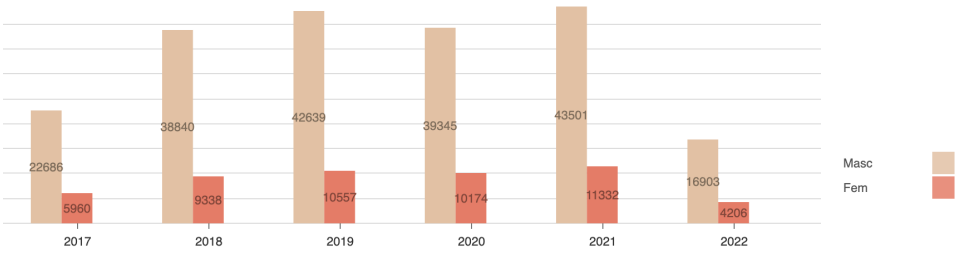
- Specifická syntaktická struktura vět.

Při hledaní citací v některých případech nebylo možné určit jejích autora. Toto bylo způsobeno specifickou syntaktickou strukturou vět (viz příklad níže), kde neexistoval podmět a jelikož identifikace autora probíhala za předpokladu, že ve větě s citací vyskytují se podmět a přísudek, 40 813 citací nebylo možné asociovat s žádnou osobou z entity *Person*.

```
"Já se znepokojením sleduji, že v mezinárodních žebříčcích
↪  není většina našich univerzit příliš úspěšná, a někdy má
↪  dokonce klesající úroveň," řekl.
```



Obrázek 5.6: Graf *Personal ID -> Person -> Organization -> Occupation* pro příjmení "Hamáček"

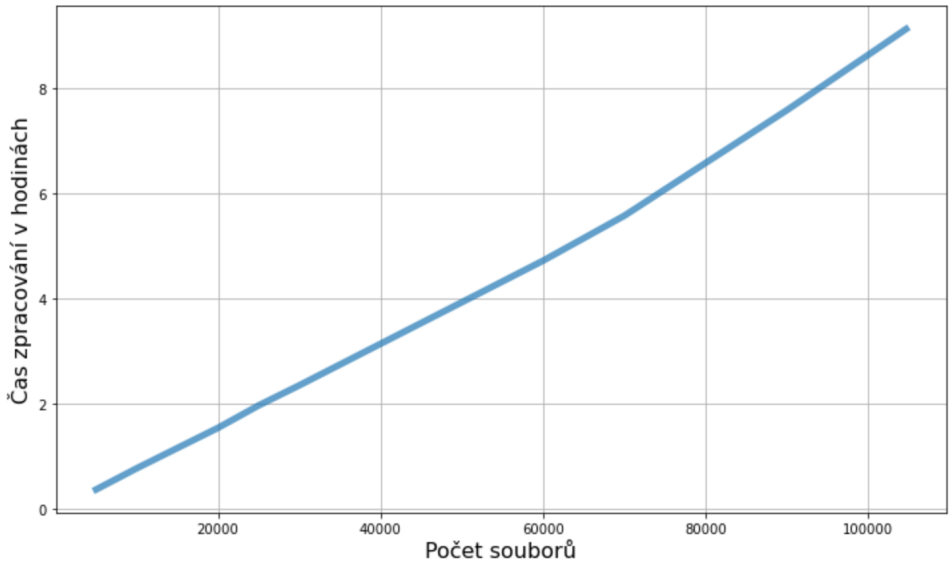


Obrázek 5.7: Graf ilustrující počet citací dle pohlaví autorů

```
"Milá paní rektorko, v případě Univerzity Karlovy je slabý
↪  argument, že jste o něco lepší než univerzita v Oregonu, a
↪  bylo by dobré, kdyby prestiž slavné Karlovy univerzity pod
↪  vaším vedením vzrostla," obrátil se na Králíčkovou.
```

- Vysoká časová náročnost programů.

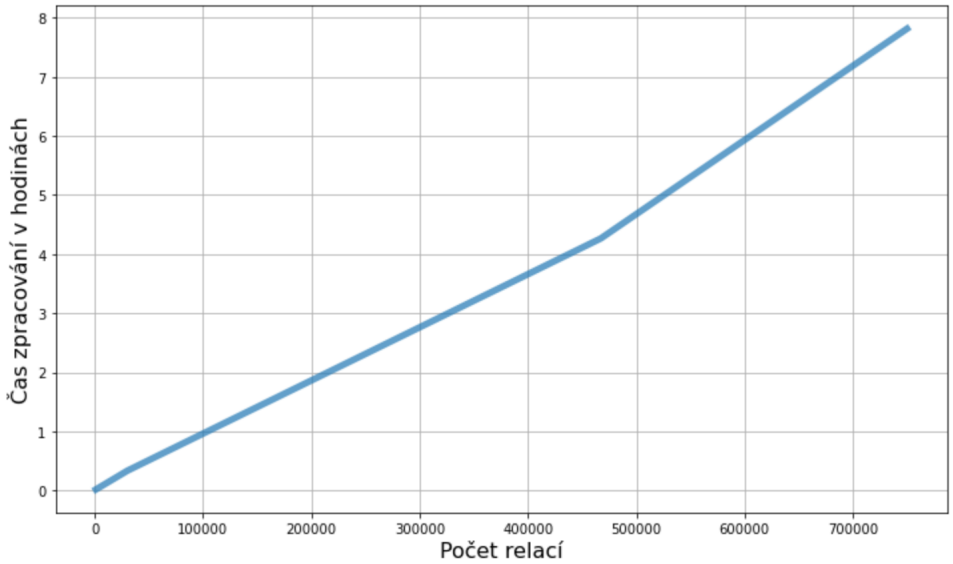
Doba přípravy entit a relací v závislostí na množství dat je popsána funkcí na obrázku 5.8. Jak je vidět z obrázku, zpracování všech souboru trvá cca 9



Obrázek 5.8: Doba přípravy entit a relací

hodin. Z toho cca 8,5 hodin trvá zpracování metody *read_files*, cca 20 minut trvá zpracování metody *get_unique_entities*. Zbytek trvá několik vteřin až jednu minutu.

Doba nahrávání relací do grafové databáze v závislostí na počtu relací je zobrazená na obrázku 5.9. Nahrávaní všech entit trvá 3-5 min, ale nahrávání relací trvá cca 6 hodin. Způsobeno to tím, že relace se tvoří na základe ID, a proto před tvorbou hrany software prochází všechny uzly, které spadají do stanovené kategorie a hledá uzel s určitým ID. Navíc díky tomu, že Neo4j je interaktivní databází, práce s velkým množstvím uzlů (nad 10 000) v závislostí na dotazu může trvat několik minut, během kterých software se zasekává a přestává odpovídat.



Obrázek 5.9: Doba nahrávaní dat do Neo4j

Kapitola 6

Závěr

Cílem teto práce bylo navržení metody identifikující relace mezi vybranými objekty a aspekty lingvisticky předzpracovaných novinářských článků.

Na základě analýzy vstupních dat bylo rozhodnuto provést rozpoznávaní pojmenovaných entit a extrakcí relací na bázi ručně definovaných pravidel. Hlavním důvodem volby teto metody zpracování datasetu bylo to, že data byla již předběžně zpracována, a proto implementace metod NER a RE založených na pravidlech se zdála vhodným způsobem získávání dat potřebných pro tvorbu znalostního grafu.

Tak identifikace vlastních jmen byla provedena na základě morfologických charakteristik tokenů a existence syntaktické charakteristiky *flat* reprezentující relaci mezi vlastními jmény. Identifikace povolání byla provedená na základě přítomnosti relace mezi vlastními a podstatnými jmény. Nalezení organizací bylo provedeno na základě existence vztahů mezí podstatnými jmény, kde jedním z tokenu bylo slovo spadající do entity povolání. Identifikace citací byla realizována na základě počtu vyskytnutí uvozovek.

Navržená metoda pro identifikací relací a entit byla implementována v programovacím jazyku Python, vizualizace znalostního grafu byla provedena v grafové NoSQL databázi Neo4j.

Analýza časové náročností metody ukázala, že doba zpracovaní skriptu je lineárně závislá na množství vstupních dat a pro 100 000 tisíc novinářských článků zpracování včetně nahrávání dat do databáze trvá 18 hodin.

Experimentální dotazy nad databází ukázaly, že navržená metoda umí identifikovat všechny požadované entity a relace a dokonce i umožňuje detekování jmen v cizích jazycích (například arabštině nebo ruštině), nicméně v datových souborech entit povolání a organizace často se vyskytovala slova, která tam nemusela být. Bylo to způsobeno tím, že vstupní data obsahovala chyby v morfologických vlastnostech, což následovně neumožňovalo správnou detekcí entit, protože nebylo možné navrhnout pravidlo tak, aby ono fungovalo i v případě chybného morfologického rozboru tokenů. Toto ale znamená, že pro přesnější identifikaci entit a relací metoda detekce NER a RE musí být upravena takovým způsobem, aby všechny potenciální slova spadající do entit povolání a organizace byly po druhé zkontrolovány jinou metodou.

Nejvhodnějším způsobem zdokonalení navržené metody je její sloučení se slovníkovou metodou. Takový hybridní systém by následovně umožnil zbavit

se nerelevantních relací a redukovat počet entit, což by taky snížilo časovou náročnost metody provádějící import dat do grafové databáze. Navíc toto by snížilo reakční dobu databáze na dotazy.

Literatura

[1] Baru, Chaitanya K. *An Introduction to Knowledge Graphs: Knowledge Graph Definition* [online]. In: . April 1, 2021 [cit. 2023-02-21]. Dostupné z: `https://web.stanford.edu/~vinayc/kg/notes/What_is_a_Knowledge_Graph.html`

[2] Hana, Jiří a Daniel Zeman. *MANUAL FOR MORPHOLOGICAL ANNOTATION: 2.2.1. POSITIONAL TAGS* [online]. [cit. 2022-07-23]. Dostupné z: `https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/m-layer/html/ch02s02s01.html`

[3] Hogan, Aidan a Eva Blomqvist, a Michael Cochez, a Claudia d'Amato, a Gerard de Melo, a Claudio Gutiérrez, a Sabrina Kirrane, a José Emilio Labra Gayo, a Roberto Navigli, a Sebastian Neumaier, a Axel-Cyrille Ngonga Ngomo, a Axel Polleres, a Sabbir M. Rashid, a Anisa Rula, a Lukas Schmelzeisen, a Juan F. Sequeda, a Steffen Staab, a Antoine Zimmermann. *Knowledge Graphs*. Morgan & Claypool Publishers, 2021. ISBN 1636392369.

[4] Kravalová, Jana a Zdeněk Žabokrtský, 2009. Czech Named Entity Corpus and SVM-based Recognizer [online]. Association for Computational Linguistics. Suntec, Singapore [cit. 2023-05-23]. Dostupné z: `https://aclanthology.org/W09-3538/`

[5] Konoshenko, Evgeny. *Možnosti využití databáze Neo4j*, 2016. Dostupné také z: `https://insis.vse.cz/zp/56523`. Bakalářská práce. Vysoká škola ekonomická v Praze.

[6] Lessner, Dan a Martin Lána, a Michala Podrázká Tomková, a Jiří Haut. *Základy informatiky pro střední školy* [online], 2020. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta [cit. 2023-02-17]. Dostupné z: `https://popelka.ms.mff.cuni.cz/~lessner/mw/index.php/Meta:Z%C3%A1klady_informatiky_pro_st%C5%99edn%C3%AD_%C5%A1koly`

[7] Maynard, Diana a Kalina Bontcheva, a Isabelle Augenstein. *Natural Language Processing for the Semantic Web*. Morgan & Claypool Publishers, 2016. ISBN 1627059091.

[8] Rusínová Zdenka a Vladimír Petkevič. *MORFOLOGICKÁ ANALÝZA | Nový encyklopedický slovník češtiny*. [online]. CzechEncy - Nový encyklopedický slovník češtiny. [cit. 2023-03-19]. Dostupné z: `https://www.czechency.org/slovnik/MORFOLOGICK%C3%81%20ANAL%C3%9DZA`

[9] *Czech UD* [online]. Universal Dependencies contributors [cit. 2023-01-07]. Dostupné z: `https://universaldependencies.org/cs/`

[10] *Universal Dependencies* [online]. Universal Dependencies contributors [cit. 2023-01-07]. Dostupné z: `https://universaldependencies.org/u/dep/all.html#al-u-dep/aux`

[11] *Tools | ÚFAL* [online]. Institute of Formal and Applied Linguistics [cit. 2023-03-21]. Dostupné z: `https://ufal.mff.cuni.cz/tools`