VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EXTRAKCE INFORMACÍ Z WIKIPEDIE

INFORMATION EXTRACTION FROM WIKIPEDIA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **ONDŘEJ VALUŠEK**
AUTHOR

VEDOUCÍ PRÁCE — **doc. RNDr. PAVEL SMRŽ, Ph.D.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Valušek Ondřej**
Program: Informační technologie
Název: **Extrakce informací z Wikipedie**
Information Extraction from Wikipedia
Kategorie: Umělá inteligence

Zadání:

1. Seznamte se s metodami extrakce informací z textu na základě strojového učení.
2. Navrhněte a implementujte systém pro automatickou extrakci typů a základních atributů pojmenovaných entit z exportu dat anglické Wikipedie.
3. Vytvořte systém pro pravidelné aktualizace znalostní báze, který označí typy změn, k nimž došlo.
4. Vyhodnoťte výsledky systému na reprezentativním vzorku dat.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- Manning, C. D., Schütze, H., Foundations of Statistical Natural Language Processing, MIT Press, 1999, ISBN 0-262-13360-1.

Pro udělení zápočtu za první semestr je požadováno:

- funkční prototyp řešení

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Smrž Pavel, doc. RNDr., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 17. dubna 2019

Abstrakt

Tato práce se zabývá automatickou extrakcí typů entit ve článcích anglické Wikipedie a jejich vybraných atributů. Jsou v ní představeny postupy za využití prvků strojového učení, které lze ke splnění tohoto účelu využít. Z článků jsou také extrahovány některé důležité atributy, jako například data narození u osob, rozlohy u jezer a podobně. Pomocí systému představeného v této práci je možné ze souboru obsahující všechny články Wikipedie (tzv. dump souboru) vytvořit znalostní databázi, ve které budou klasifikovány miliony článků, dle typu entity o které pojednávají, na základě malé tréninkové sady. Při tomto procesu je také generován soubor, kde jsou kromě ostatních příznaků z článků extrahována tzv. definiční slova, což jsou klíčová slova nalezena pomocí analýzy přirozeného textu. Ta je možno použít také v jiných oblastech, než pouze při určování typů entit. Součástí celého systému je také modul, který označí změny mezi jednotlivými verzemi znalostní databáze, tedy například, které články byly přidány, které smazány a u kterých se udála změna.

Abstract

This thesis deals with automatic type extraction in English Wikipedia articles and their attributes. Several approaches with the use of machine learning will be presented. Furthermore, important features like date of birth in articles regarding people, or area in those about lakes, and many more, will be extracted. With the use of the system presented in this thesis, one can generate a well structured knowledge base, using a file with Wikipedia articles (called dump file) and a small training set containing a few well-classed articles. Such knowledge base can then be used for semantic enrichment of text. During this process a file with so called definition words is generated. Definition words are features extracted by natural text analysis, which could be used also in other ways than in this thesis. There is also a component that can determine, which articles were added, deleted or modified in between the creation of two different knowledge bases.

Klíčová slova

klasifikace článků, určování typů entit, přirozený text, zpracování přirozeného jazyka, určování slovních druhů, SpaCy, Stanford CoreNLP, Wikipedie, SVM, Metoda podpůrných vektorů, strojové učení, umělá inteligence, extrakce atributů

Keywords

article classification, entity type detection, natural text, natural language processing, part-of-speech tagging, SpaCy, Stanford CoreNLP, Wikipedia, SVM, Support Vector Machine, machine learning, artificial intelligence, attribute extraction

Citace

VALUŠEK, Ondřej. *Extrakce informací z Wikipedie*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Pavel Smrž, Ph.D.

Extrakce informací z Wikipedie

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Smrže a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Valušek
15. května 2019

Poděkování

Rád bych poděkoval panu doc. Pavlu Smržovi, za rady a postřehy, které mi v průběhu vypracovávání poskytl.

Obsah

Kapitola 1

Úvod

Již od dávných dob vznikala snaha koncentrovat lidstvu známé informace na jedno místo. Kdysi bylo toto doménou ručně psaných a později tištěných encyklopedií. S příchodem a hlavně rozšířením internetu se to však změnilo. V lednu 2001 vznikl portál Wikipedia[1], který je dnes největší internetovou encyklopedií na světě. Ta obsahuje jen v anglické verzi ke květnu 2019 přes 5 milionů článků.
Je pochopitelné, že internetové encyklopedie jsou psány „pro lidi“ a nikoliv „pro stroje“. Na jedné straně je k dispozici nepřeberné množství kvalitních informací, které by bylo vhodné různým způsobem strojově zpracovávat, a na druhé straně jsou k dispozici nestrukturovaná, nebo částečně strukturovaná data, která jsou pro počítačové zpracování v originální podobě nepoužitelná. Přirozeně je tedy snaha takové informace převádět do strukturované podoby, aby je bylo možné efektivně strojově zpracovávat.
Řešením takového problému se zabývá tato práce. Jsou v ní představeny postupy, jakými je možné z celé Wikipedie, respektive tzv. dumpu, což je soubor, kam je Wikipedie vždy k 1. a 20. dni v měsíci, exportována, vygenerovat znalostní bázi, ve které budou miliony záznamů. Tu bude možné použít například ke sémantickému obohacování textu.
K tomu jsou použity dva moduly. První z nich, dle postupu popsaného v kapitole 4.1, transformuje dump soubor na TSV soubor, který obsahuje pro každý článek jeden řádek, na kterém se nachází titulek, první věta, první odstavec a řetězec, ze kterého bude možné extrahovat základní atributy o entitě. Kromě těchto informací také příznaky, na základě kterých je pak možno u článků určovat typy entit, o kterých pojednávají. Těmito příznaky jsou jak text extrahovaný ze strukturovaných elementů na stránce, tak i klíčové fráze a slova nalezená pomocí analýzy přirozeného textu, také označována jako „definiční“.
Za použití metody podpůrných vektorů jsou pak pomocí postupů popsaných v kapitole 4.2 u článků na základě výše zmíněných příznaků zjišťovány typy entit, o kterých pojednávají. Protože vychází dump soubor relativně často a obsah Wikipedie se mění, bude v této práci v kapitole 4.3 také představen postup, jakým je možné detekovat změny mezi jednotlivými verzemi znalostní báze, označit články které přibyly, byly naopak smazány nebo ty, u kterých se změnil některý z atributů.

[1] https://www.wikipedia.org

Kapitola 2

Rozbor řešené problematiky

V této kapitole bude nejdříve představeno několik prací, které se již podobnou problematikou zabývaly. Poté bude popsán vývoj v posledních letech a do tohoto kontextu zasazen zde představovaný systém. Dále budou konkrétněji představeny existující nástroje, které se řešením podobného problému, jako tato práce, již zabývají. Protože budou výsledky systému prezentovaného v této práci v kapitole 5 s těmito nástroji srovnávány, bude také popsána srovnávací metodika. Poté budou představeny metody využívané v této práci a nakonec bude následovat popis toho, jak vypadají data, se kterými se pracuje.

2.1 Dosavadní práce v oblasti extrakcí informací z Wikipedie

Protože je Wikipedie největší internetovou encyklopedií, prací které se zabývají extrakcí informací z ní, již vzniklo mnoho. Obecně se dá říci, že ač se najdou výjimky [12], které už v roce 2007 pracovaly s analýzou přirozeného textu, starší z prací [21] v této oblasti se spíše omezovaly na lépe strukturované části Wikipedie. Typickým představitelem takových částí je entita infobox, která je na obrázku 2.7. V té je vidět, že text je zde napsán vyloženě heslovitě a lze jej extrahovat například použitím regulárních výrazů.

Problém v použití těchto a také jiných, infoboxům podobných, dobře strukturovaných částí stránek je ten, že obecně platí, že čím lépe strukturovaná část je, tím menší je počet stránek, na kterých se nachází.

Z tohoto důvodu se ukázalo jako výhodnější pracovat i s hůře strukturovanými částmi článku. Jednou z takových je pochopitelně samotný text článku, respektive jeho části, typicky první věta, nebo odstavec.

V roce 2010 tak například vznikla práce [11], ve které se z první věty článku extrahují slovní druhy a sémantické závislosti mezi slovy. Na základě těchto závislostí a také informací získaných pomocí modulu na rozpoznávání jmenných entit, se hledaly vzájemné souvislosti mezi nimi. Ty byly následně zobecňovány do „vzorů“ a poté z nich byla vytvořena tréninková sada pro automatickou klasifikaci. Správnost určování typů entit tehdy dosahovala kolem 90 %.

V dnešní době, kdy jsou běžně k dispozici nástroje pro zpracování přirozeného textu, jsou ke klasifikaci textu samozřejmě často používány. I k samotné extrakci příznaků a vytvoření tréninkové sady lze použít existující nástroje, jako jsou například DeepDive[1], nebo aktuálnější Snorkel[2]. První z nich byl například použit v práci [2], kde pomocí tohoto

[1] http://deepdive.stanford.edu

[2] https://hazyresearch.github.io/snorkel/

nástroje byly extrahovány příznaky pouze z přirozeného textu. Na vstup byly přivedeny celé články Wikipedie v podobě „čistého“ textu a pomocí určování slovních druhů a rozpoznávání jmenných entit byla počítána pravděpodobnost existence vztahu mezi entitami (například, zda jsou v manželství a podobně). Pouze pomocí práce s přirozeným textem zde bylo dosaženo přesnosti 91 %. Při extrakci informací se však samozřejmě není nutno omezovat pouze na entity viditelné při procházení Wikipedie pomocí webového prohlížeče. V roce 2014 byl představen nástroj [7], který kombinoval několik různých přístupů. Od analýzy textu pomocí nástrojů pro zpracování přirozeného jazyka, přes využití manuálních anotací přidaných uživateli, až po analýzu a extrakci informací z HTML značek.

Obecně totiž neexistuje důvod, proč analyzovat pouze strukturovaná data, nebo naopak pouze ta nestrukturovaná. Nejlepším přístupem je tak obvykle kombinace obou. Přestože strukturované části stránky jsou sice při využití u určování typů spolehlivější, zdaleka se nenacházejí na všech stránkách. Tam kde by nebyly, by tedy nebylo dle čeho klasifikovat.

Systém prezentovaný v této práci tak používá přístup, který je podobný například práci [19] z roku 2016, kde byly k určování typů entit použity vektory příznaků, které se skládaly jak z informací extrahovaných ze strukturovaných částí, jako jsou třeba kategorie uvedené u článků, tak i z těch, které přímo v článku nejsou a musí být odvozeny. Takovými jsou třeba poslední podstatné jméno z první věty, které bylo použito jako jedna ze součástí výše zmíněného vektoru příznaků. Další práce, která je podobná zde prezentované je například tato [20], kde byly podobně jako zde, extrahovány části textu z první věty článku. Ty byly poté využity k propojování entit napříč znalostními bázemi. Trend, kdy jsou příznaky extrahovány z obou typů dat potvrzuje i tato práce [13] z roku 2017, ve které jsou příznaky extrahovány i z titulku. Existují také práce, které nepoužívají extrahované příznaky jen ke klasifikaci, ale třeba v této [6] nebo této [8] práci byly použity k vytvoření systému, který je schopen odpovídat na otázky[3], na základě extrakce informací z Wikipedie. V této práci jsou tak jako příznaky používána kromě informací ze strukturovaných částí textu jakými jsou třeba infobox a seznam kategorií u každého článku, také klíčová slova a fráze. Všechny tyto jsou z textu předem extrahovány pomocí nástrojů pro zpracování přirozeného textu, čímž se také předejde problémům jako jsou například prokletí dimenzionality[4] nebo obecně příliš dlouhá doba zpracování, protože na vstup klasifikátoru nejsou přiváděny celé věty, nebo články, ale jen několik klíčových slov. Metod extrakce těchto klíčových slov z textu je samozřejmě mnoho, jak je popsáno třeba v práci [3] nebo zde [17].

V této práci byla použita extrakce pomocí dvou nástrojů. Těmi jsou knihovna SpaCy [1] a modul Stanford CoreNLP [9]. Oba z nich určují slovní druhy a analyzují sémantické závislosti ve větě, na základě kterých jsou pak klíčová slova hledána. Klasifikace pak probíhá pomocí metody podpůrných vektorů [10], za použití modulu Scikit-learn [14].

2.2 Existující řešení

Nyní budou představeny nástroje, které již existují a řeší stejný nebo podobný problém, jako systém prezentovaný v této práci. Celkové statistiky (například kolik entit které řešení obsahuje, jejich správnost a podobně) pak budou představeny v kapitole 5.

[3] Otázky typu kdo kde žil, kdo založil nějakou organizaci a podobně.

[4] https://towardsdatascience.com/curse-of-dimensionality-2092410f3d27

DBpedia

DBpedia[5] je projekt, jehož cílem je extrakce dat z Wikipedie a jiných projektů založených na softwaru Wikimiedia[6].

V podstatě se také jedná o znalostní bázi. Ta obsahuje, kromě klasifikovaných článků, také tzv. „Open knowlgedge graph“. To je struktura obsahující informace, které kromě článků obsahují i odkazy na související články, kategorie a podobně. V této struktuře je tedy možné procházet články obdobně, jako při použití standardního webového prohlížeče a stránek Wikipedie, informace ale mají pevnou strukturu. Je ji tedy možné stejně tak dobře procházet i strojově.

Největší „slabinou“ tohoto projektu je jeho neaktuálnost. Poslední ucelená verze vyšla v říjnu 2016.

„Napravit“ tento nedostatek má projekt DBpedia Live[7]. Ten na změny na Wikipedii reaguje velmi rychle, někdy i v řádu několika minut. Je přístupný především přes webové rozhraní, kde lze psát dotazy v jazyce SPARQL[8]. To je jazyk podobný známějšímu SQL[9]. Například dotaz na všechny osoby v databázi by vypadal takto:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?resource
WHERE {
?resource rdf:type dbo:Person.
}
```

Protože je však toto rozhraní přístupné veřejně, aby nebyl server přetěžován, je zde nastaven pevný maximální limit na 10 tisíc záznamů ve výsledku dotazu. Protože je na Wikipedii mnohem více osob než 10 tisíc, výsledek je nekompletní a přesto, že lze vypsat alespoň počet entit ve výsledků, k jejich seznamu se pomocí online dotazování nelze dostat.

Lze se k němu však dostat přes dump soubor DBpedie Live, což je soubor obsahující celou znalostní bázi, stejně jako u „normální“ DBpedie. Po stažení lze běžnými textovými programy procházet a samozřejmě v něm i vyhledávat. Tyto dumpy jsou však, stejně jako v případě „normální“ DBpedie, zastaralé.

K aktuální verzi se lze po vynaložení určitého úsilí „dopracovat“ pomocí nástroje[10] vytvořeném autory DBpedie. Tento nástroj je schopen doplnit rozdíly mezi některým z dumpů a aktuální verzí DBpedie Live.

Wikidata

Wikidata[11] je dalším projektem, resp. nástrojem, pomocí kterého je možné procházet data z Wikipedie a dalších podobných projektů, ve strukturované podobě. Pro přístup k datům zde lze využít hned několik nástrojů. Prvním z nich je „Wikidata Query Service“[12]. Ten funguje obdobně jako dotazování popsané výše. Slouží k online dotazování a také používá

[5] https://wiki.dbpedia.org/
[6] https://www.wikimedia.org/
[7] https://wiki.dbpedia.org/online-access/DBpediaLive
[8] http://www.w3.org/TR/sparql11-query
[9] https://www.w3schools.com/sql/sql_intro.asp
[10] https://github.com/dbpedia/dbpedia-live-mirror
[11] https://www.wikidata.org/wiki/Wikidata:Main_Page
[12] https://query.wikidata.org/

stejný jazyk SPARQL jako v případě DBpedie. Dotazy vypadají také velmi podobně, například tento dotaz na hory vyšší než 8000m:

```
SELECT ?subj ?label ?coord ?elev
WHERE
{
?subj wdt:P2044 ?elev filter(?elev > 8000) .
?subj wdt:P625 ?coord .
SERVICE wikibase:label { bd:serviceParam wikibase:language "en".
?subj rdfs:label ?label }
}
```

Největší rozdíl oproti dotazování u DBpedie je složitější zápis názvu entit. Ten není slovní jako u DBpedie, kde byly řetězce jako „Person“ a podobně. Zde je nutné nejdříve najít identifikátor kategorie, který má podobu například „Q146“[13] a podobně. Online nástroj, pomocí kterého se lze takto dotazovat však obsahuje šikovnou nápovědu s mnoha vzorovými dotazy, které lze filtrovat, řadit a podobně, což začátky práce s ním značně usnadní.

Druhým nástrojem je „Reasonator“[14]. Ten ze strukturovaných dat na Wikidatech opět „složí“ souvislý text, ve kterém jsou pak obsaženy nejdůležitější informace. Z níže uvedeného příkladu je patrné, že se sice jedná o souvislý text, koncentrace extrahovaných atributů je však příliš vysoká, a text tak nepůsobí přirozeně. Následující ukázka prezentuje vygenerovaný text z informací extrahovaných z článku o Václavu Havlovi:

Václav Havel was a Czechoslovakia-Czech Republic writer, playwright, politician, director, and poet. He played a role in prisoner of conscience and Projev Václava Havla v Rudolfinu v roce 1997. He was born on October 5, 1936 in Prague to Václav Maria Havel and Božena Vavrečková. He studied at Czech Technical University in Prague from 1955 until 1957, Akademické gymnázium Štěpánská until 1955, and Faculty of Theatre. He was President of Czechoslovakia from December 29, 1989 until July 20, 1992, President of the Czech Republic from February 2, 1993 until February 2, 2003, and spokesperson of Charter 77. He was a member of Club of Rome, Royal Society of Literature, Civic Forum, Czech Helsinki Comitee, Committee for the Defense of the Unjustly Prosecuted, Deutsche Akademie für Sprache und Dichtung, American Academy of Arts and Sciences, Bavarian Academy of Fine Arts, and scouting. He worked for Brewery Trutnov, for ABC Theatre, for City Theatres of Prague, for Theatre on the Balustrade from 1960 until 1968, and for Institute of Chemical Technology in Prague from 1951 until 1955.

Řešení vyvíjená na FIT VUT

V rámci Výzkumné skupiny znalostních technologií [15] vzniká na Vysokém učení technickém v Brně několik podobných projektů. Některé z nich, stejně jako systém představený v této práci, generují z Wiki dumpu znalostní bázi ve strukturované podobě. Na rozdíl od řešení

[13] identifikátor typu kočka domácí
[14] https://tools.wmflabs.org/reasonator
[15] knot.fit.vutbr.cz

prezentovaného zde, se tyto systémy učí typy entit na základě předchozích verzí znalostních bází, nikoliv na základě souboru s tréninkovou sadou.

Výsledkem při použití těchto nástrojů je znalostní báze ve formátu TSV, nad kterou se nelze dotazovat přímo pomocí speciálního jazyka k tomu určenému, ale díky její pevné struktuře v ní lze jednoduše vyhledávat běžnými nástroji pro práci s textem, jak je například nástroj awk[16].

Jiným přístupem, který strojové učení používá, je práce [16] pana Bc. Rusiňáka, která na vstupu přijímá dvě množiny článků, z nichž jedna obsahuje pozitivní příklady a druhá negativní. Na základě těchto dvou množin je možné identifikovat všechny články o jednom typu entity, tedy například nalézt všechny články o psech.

2.3 Metodika srovnávání jednotlivých řešení

Jak bylo psáno v předchozí kapitole, několik řešení zabývající se podobnou problematikou již existuje a přirozeně je vhodné novou práci v podobné oblasti s těmito srovnat. Nyní bude popsáno, jak toto probíhá.

Při porovnávání budou srovnávány dvě hodnoty. První z nich je počet identifikovaných entit u každého typu, přičemž vyšší číslo je samozřejmě lepší. Druhou z nich je správnost určení, přičemž vyšší číslo je opět lepší.

Takovéto porovnávání je však problematické. Nelze totiž říci, jaký je u každého typu správný počet entit. K určení takové hodnoty by totiž byl potřeba systém, který by se 100% přesností byl schopen určit typy entit v článcích. Takový pochopitelně neexistuje.

Logickým přístupem k řešení tohoto problému by mohlo být například detekování rozdílů mezi verdikty jednotlivých systémů a manuální kontrola těch článků, u kterých se typy entit navzájem neshodují.

Tento postup byl při vývoji zde prezentovaného systému vyzkoušen, ale nakonec se ukázal jako nepoužitelný. Důvodem k tomu je fakt, že každý z nástrojů jehož výsledky byly srovnávány, používá jiné zdroje, ze kterých entity určuje. Tedy například v případě DBpedie jsou to různé jazykové mutace Wikipedie, projekt GeoNames[17] a WordNet[18]. V případě nástroje Wikidata je počet zdrojů o poznání vyšší a tedy u tohoto nástroje je přirozeně i počet určených entit mnohem vyšší. Stejně tak v případě nástrojů NER jsou zdroje odlišné. Tento problém by bylo možné řešit tak, že by se omezily zdroje u každého nástroje na stejnou množinu (tedy například všechny systémy by pracovaly pouze s anglickou Wikipedií). V takovém případě by šlo skutečně přesně uvést, který systém je přesnější. Takové „omezení se“ však v současné době není možné efektivně provést.

Z tohoto důvodu byl nakonec zvolen přímočarý postup, kdy je u každého nástroje vždy pro všechny typy dvakrát náhodně vybráno 100 článků a u těch je správnost určení typu manuálně zkontrolována, přičemž výslednou hodnotou je průměr z těchto dvou vyhodnocení. Za zmínku také stojí poznámka, že před vyhodnocováním byly odstraněny duplicity. Aby nebyly uváděny typy entit, které by byly pro účel obohacování textu nepoužitelné, typ byl do vyhodnocení zařazen pouze pokud obsahoval alespoň 100 entit s vyšší správností určení než 80 %.

[16] https://www.gnu.org/software/gawk/manual/gawk.html
[17] https://www.geonames.org/
[18] https://wordnet.princeton.edu/

Počet nalezených entit

První hodnota bude tedy určena vztahem:

$$P = P_t + P_p$$

Kde:

P_t: Je počet článků pojednávajících o entitě typu t

P_p: Je počet článků pojednávajících o entitě některého z podtypů typu t

Některé z nástrojů totiž používají více hierarchických úrovní typů. V takovém případě je nutné zahrnout i všechny podtypy.

Správnost

Tato metrika určuje, u kolika procent z počtu náhodně vybraných a manuálně zkontrolovaných byl správně určen typ entity.

Druhá hodnota je tedy určena vztahem:

$$S = 100 \cdot \frac{V_s}{V_c}[\%]$$

Kde:

V_s: Je počet článků se správně určeným typem entity

V_c: Je počet článků, které byly kontrolovány

2.4 Moderní přístupy k řešení klasifikačního problému

Je zřejmé, že u pěti miliónů článků není vhodné určovat typy entit manuálně. Nyní bude představeno, jak je tento problém řešen za použití počítačů a moderních metod s využitím prvků strojového učení.

Typy strojového učení

Aktuálně používané přístupy řeší tento problém typicky tak, že je ručně klasifikován malý počet článků a poté se přistoupí k využití strojového učení. Takto je možné natrénovat klasifikátor, který na základě malého počtu správně klasifikovaných článků, klasifikuje článků miliony. Zde samozřejmě vzniká riziko chyby. Toto riziko je možné minimalizovat, nikoliv však odstranit. Takový přístup je však „realističtější“ než výše popsáná manuální klasifikace, a proto je při řešení podobných problémů preferován.

Strojové učení je obecně možno rozdělit do dvou tříd:

1. Učení bez učitele
2. Učení s učitelem

Učení s učitelem

V této práci je použita jedna z metod spadající do třídy učení s učitelem. U metod učení s učitelem je na vstupu potřeba „informace navíc“ v podobě správných výsledků oproti učení bez učitele, kde klasifikátor pouze „hledá podobnosti“ ve vstupních datech. Díky tomu, že je tato informace k dispozici, je však klasifikátor schopen určit (pro zde uvedený příklad určování typů entit) nejen „podobné články“, ale i konkrétní typ entity o které pojednávají. Metod spadajících do třídy učení s učitelem je mnoho. Zde budou popsány dva možné přístupy, které lze použít ke klasifikaci textu a tedy určování typu entit v článcích.

Naivní Bayes

Tento klasifikátor je jedním z nejjednodušších možných. Vztah pro výpočet pravděpodobnosti, že článek pojednává o entitě typu y v závislosti na příznacích x_1 až x_n je uveden v rovnici 2.1.

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y) \tag{2.1}$$

Protože je žádoucí jako „výsledný typ entity“ uvést ten, pro kterou je pravděpodobnost nejvyšší, tento výsledný typ je určen maximální hodnotou y, jak je ukázáno v rovnici 2.2.

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y), \tag{2.2}$$

V dřívějších verzích této práce byl tento klasifikátor použit. Výsledky nicméně nebyly tak dobré jako za použití metody podpůrných vektorů, která bude představena dále. Oba vztahy 2.1 a 2.2 byly převzaty z dokumentace k modulu Scikit-learn [14] a implementace u jiných modulů se může mírně lišit.

Metoda podpůrných vektorů

Druhou zde představenou metodou, je metoda podpůrných vektorů (anglicky Support Vector Machine, zkráceně SVM). Výsledky dosažené za použití tohoto klasifikátoru byly při testování přesnější, než výsledky zíkané pomocí výše představeného klasifikátoru Naivní Bayes, a je proto v této práci použit při určování typů entit u článků.

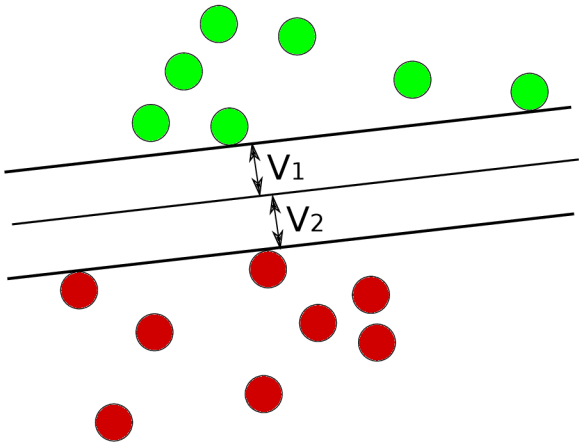
Nejjednodušší SVM fungují na principu rozdělení vstupních dat přímkou, přičemž je žádoucí, aby byla vzdálenost (na obrázku V_1 a V_2) mezi přímkou a podpůrnými vektory (tedy těmi nejblíže přímce), co největší [4]. Toto je ilustrováno na obrázku 2.1.

Jak je však patrné z obrázků 2.2 a 2.3, ne všechna data jsou lineárně separovatelná. Proto budou dále představeny i pokročilejší „modifikace“ SVM, které si s tímto problémem umí poradit mapováním dat do vyšších dimenzí.

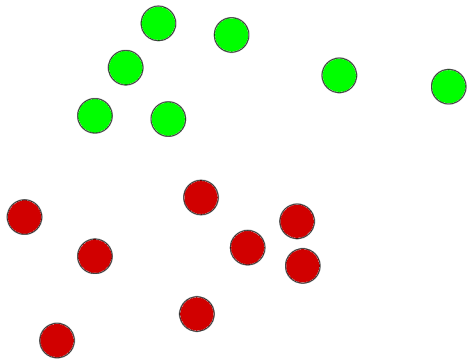
Počet tříd

Dle počtu tříd do kterých SVM data klasifikuje je možné tyto rozdělit na:

1. SVM klasifikující do dvou tříd (two class)
2. SVM klasifikující do více než dvou tříd (multiclass)



Obrázek 2.1: Rozdělení objektů reprezentující data přímkou s co největší vzdáleností od objektů



Obrázek 2.2: Příklad reprezentace lineárně separovatelných objektů

Je nutné poznamenat, že typicky provádějí všechny SVM klasifikaci binární. Klasifikace do více než dvou tříd je řešena vytvořením více klasifikátorů a postupným porovnáváním výsledků s ostatními [22].

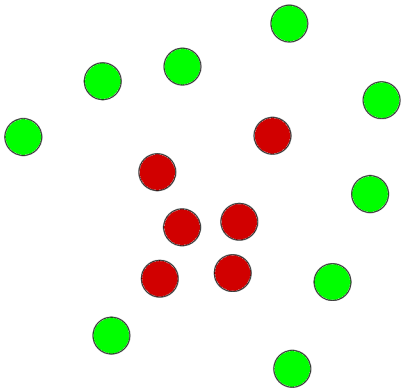
Jádrová funkce

Jádrová funkce umožňuje transformovat vstupní data na data jiná, typicky data ve více dimenzích. To je samozřejmě výhodné zejména tam, kde nejsou data lineárně separovatelná. Jednoduše řečeno, pokud by byly červené body z obrázku 2.3 „nadzvednuty“, oddělit je rovinou od zelených, by nebyl žádný problém. Jádrových funkcí existuje mnoho[19], nejznámější a nejpoužívanější z nich však jsou:

Lineární

Lineární jádrová funkce je ze všech funkcí nejjednodušší. Její výhodou při použití je především rychlost trénování klasifikátoru. Pokud jsou tedy vstupní data lineárně separovatelná, je vhodné ji využít. Protože je klasifikace pomocí této jádrové funkce typicky nejrychlejší,

[19] Ve skutečnosti nekonečně mnoho, protože uživatel si může nadefinovat jakoukoliv vlastní.

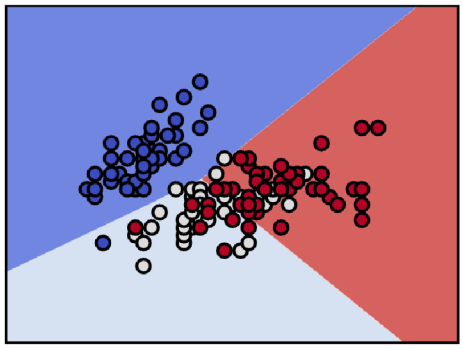


Obrázek 2.3: Příklad reprezentace lineárně neseparovatelných objektů

obecně se doporučuje nejdříve vyzkoušet, zda jsou výsledky za použití této funkce správné a teprve pokud nejsou, zvolit některou ze složitějších.

Pro lepší představu rozdílu mezi jednotlivými jádrovými funkcemi byly vygenerovány obrázky toto ilustrující[20].

Ilustrace rozdělení dat pomocí lineárního klasifikátoru je na obrázku 2.4.



Obrázek 2.4: Rozdělení dat při použití lineární jádrové funkce

Polynomiální

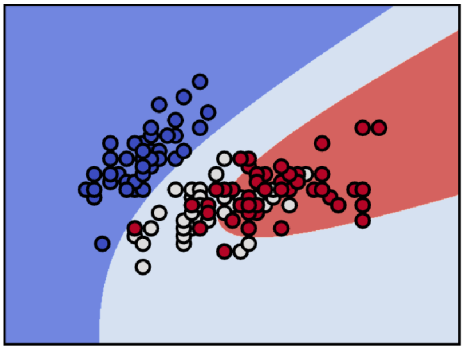
Polynomiální funkce umožňuje použít k oddělení vstupních dat nejen přímku, ale i křivku. V závislosti na nastavení parametru funkce to může být křivka kvadratická, kubická, nebo i vyšších stupňů[21]. Ilustrace rozdělení dat pomocí polynomiálního klasifikátoru je na obrázku 2.5.

Radiální bázová (Gaussovská)

Tato funkce je ze všech zde popisovaných typicky výpočetně nejnáročnější, ale její použití dosahuje obecně nejlepších výsledků, což se potvrdilo i pro data v této práci a radiální bázová funkce je v ní tedy použita. Ilustrace rozdělení dat pomocí klasifikátoru používajícího

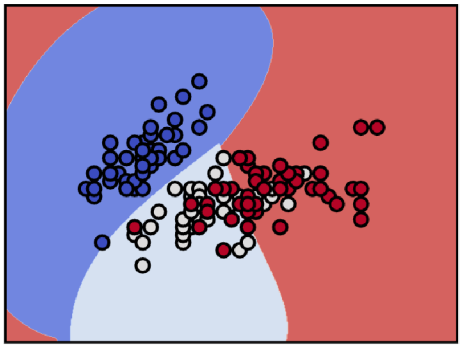
[20] Pomocí kódu dostupného ze stránek https://scikit-learn.org/stable/modules/svm.html

[21] Modul Scikit-learn využívaný v této práci umožňuje volbu až do stupně 6.



Obrázek 2.5: Rozdělení dat při použití polynomiálního jádra šestého stupně

radiální bázovou funkci je na obrázku 2.6.



Obrázek 2.6: Rozdělení dat při použití Radiální bázové funkce

Vytvoření tréninkové sady

Protože však nelze jako vstupní data klasifikátoru použít jakákoliv data, nyní budou popsány metody, které je možné použít při převodu přirozeného textu na reprezentaci využitelnou na vstupu při trénování klasifikátoru u strojového učení a pozdější klasifikaci dat.

Takový proces je při použití externího modulu, velmi přímočarý. V dříve zmíněném modulu Scikit-learn toto usnadňuje nástroj Count Vectorizer[22]. Při použití tohoto nástroje stačí na vstup „posílat celé věty“ a o zbytek už se uživatel nemusí starat. Obecně tento modul a jemu podobné fungují tak, že text převedou do tokenové reprezentace, jaká je popsána například v tabulce 2.4 a z četnosti textu tokenů udělají tzv. matici výskytů. Tento proces a následné úpravy budou nyní popsány. Mějme následující věty:

1. František Švantner was a Slovak prose writer.
2. Vittorio Curtoni was an Italian science fiction writer and translator.
3. Heartbreaker is the debut solo studio album by American singer/songwriter Ryan Adams, released September 5, 2000 on Bloodshot Records.

[22] https://scikit-learn.org/stable/modules/feature_extraction.html

Nyní by bylo možné každou z vět poslat na vstup nástroji Count Vectorizer. Jak je však patrné ze schématu 3.1, ve chvíli kdy se mají trénovat klasifikátory, již byly z vět extrahovány definiční slova a fráze, které jsou uloženy v TSV souboru. Posílat na vstup celé věty by tedy bylo nejen zbytečně výpočetně náročné, ale také by mimo užitečných slov a frází musely analyzátory pracovat se zbytečným informačním šumem v textu. Nyní bude popsán proces při trénování klasifikátoru příznaku víceslovné definiční fráze.

Pro každou z vět ve výše uvedeném seznamu je v TSV souboru k dispozici definiční fráze[23]. Ty pro věty ve výše uvedeném seznamu vypadají následovně:

1. Slovak prose writer
2. Italian science fiction writer
3. debut solo studio album

Tyto fráze tedy budou přivedeny na vstup Count Vectorizeru. Nyní je možné vypsat seznam slov, jejichž četnost bude analyzována. Ten vypadá následovně:
„album, debut, fiction, italian, prose, science, slovak, solo, studio, writer“.
Jak je vidět, jedná se o všechna slova z definičních frází. Při analýze celé věty by to takto obvykle nebylo. Byla by vypuštěna přinejmenším slova kratší než dva znaky[24] a interpunkce. Nyní je možné tento seznam převést na matici výskytů. Ta vypadá takto:

$$M_{vysk.} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Na první pohled je možná obtížné vyčíst jaká je souvislost mezi hodnotami v matici, větami uvedenými výše a seznamem slov. Princip je ale jednoduchý:
Každý řádek matice reprezentuje jednu definiční frázi ze seznamu. Hodnoty v matici nyní určují, které slovo ze seznamu těch, jejichž četnost je analyzována, se ve frázi nachází. Například pro první řádek je nenulová hodnota na indexu 4 (páté pozici), na indexu 6 a konečně na indexu 9. V první definiční frázi se tedy nacházejí ta slova ze seznamu, která se nacházejí na indexech 4, 6 a 9 v seznamu analyzovaných slov. Tedy slova „prose, slovak, writer“, což je očekávaný výsledek, protože přesně z těchto slov se první definiční fráze skládá. Analogicky pro ostatní řádky a definiční fráze.

Tuto matici výskytů je často před použitím na vstupu tréninkové funkce vhodné transformovat na tzv. reprezentaci tf-idf (term-frequency times inverse document-frequency). Tato reprezentace zvýší váhu slovům, která se nevyskytují v mnoha různých typech dokumentů zároveň. Problém, který je tímto vyřešen, není z analýzy vybraných částí věty tak zřejmý. Pokud by však na vstup byly přivedeny celé věty, velmi často by se v nich objevovaly tvary slovesa být, určité a neurčité členy a podobně. Ty však klasifikátoru jen sotva pomohou něco rozhodnout, a jejich váha při klasifikaci je tedy snížena. Výpočet této normalizované váhy jednotlivých hodnot z výše uvedené matice je dán vztahem 2.3. Tento vztah byl převzat z dokumentace modulu Scikit-learn [14] a pro jiné implementace se může lišit.

$$v_{norm} = \frac{v}{||v||_2} = \frac{v}{\sqrt{{v_1}^2 + {v_2}^2 + \cdots + {v_n}^2}} \tag{2.3}$$

[23] pokud ji věta obsahuje

[24] Toto je výchozí nastavení v implementaci příslušného modulu v knihovně Scikit-learn. U jiných implementací by se výsledek mohl mírně lišit.

Po normalizaci by tedy matice výskytů pro výše uvedený seznam vypadala po zaokrouhlení takto:

$$M_{norm.} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0.6 & 0 & 0.6 & 0 & 0 & 0.4 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0.4 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix}$$

Takto upravenou matici je tedy možné použít na vstupu tréninkové funkce klasifikátoru.

2.5 Extrahovaná data z Wikipedie

V minulých kapitolách bylo popsáno, jak funguje trénování klasifikátorů a jak je možné textová data obecně převést do reprezentace použitelné jako tréninkovou sadu při strojovém učení. Nyní bude popsáno, jak tato data vypadají v případě internetové encyklopedie Wikipedia, jaká data se v této práci extrahují za účelem klasifikace článků a případně uchovávají ve znalostní bázi.

Příznak Infobox

Infobox je speciální element na stránce, který obsahuje hlavní informace o entitě popsané v článku. Např. infobox pro článek „Brno University of Technology“ je uveden na obrázku 2.7.

Obrázek 2.7: Infobox u článku Brno University of Technology

Jeho (zkrácená) textová reprezentace je pak následující:

```
{{Infobox University
|name = Brno University of Technology
|native_name = Vysoké učení technické v Brně
|motto = "[[Sapere aude]]" ([[Latin]])
|mottoeng = Have the courage to be wise
|established = 1899
...
|logo=File:Brno BUT Logo.png}}.
```

Je možné vidět, že infobox obsahuje důležité informace v relativně dobře strukturované podobě. Je tedy vhodné jej k extrakci informací využít namísto složité extrakce z nestrukturovaných částí článků. Pro klasifikaci článků je pak obzvláště užitečné slovo, které je hned za řetězcem „{{Infobox“. To určuje typ (název) infoboxu, a tedy je možné jej použít pro určení typu entity, o které článek pojednává.

Kromě obecného infoboxu jsou na Wikipedii používány ještě další, speciální typy infoboxů. Ty u sebe namají uvedený název, jako ten obecný. Jsou jimi „Taxobox“, „Automatic Taxobox“ a „Speciesbox“. Tyto „speciální případy“ infoboxů jsou uvedeny u článků s rostlinami či živočichy, kde jsou v nich obsaženy taxonomické kategorie[25]. Ty se vyskytují u článků s živými organismy, jako jsou např. kytky, ptáci, ryby a podobně.

Výhoda použití entity infobox pro klasifikaci spočívá v tom, že typy infoboxů lze jednoduše extrahovat a také je klasifikace pomocí nich velmi přesná. Tato entita však bohužel není zdaleka na všech stránkách, a ne vždy je tedy ji použít.

Příznak Category

Příznak category, respektive řetězec, který jej reprezentuje je extrahován ze seznamu všech kategorií, do kterých je článek na Wikipedii zařazen. Jak vypadá tento seznam při procházení Wikipedie pomocí internetového prohlížeče pro článek „Brno University of Technology“, je uvedeno na obrázku 2.8.

Obrázek 2.8: Seznam kategorií u článku Brno University of Technology

Jeho textová podoba v dump souboru je pak následující:

```
[[Category:Brno University of Technology| ]]
[[Category:Universities in the Czech Republic]]
[[Category:Education in Brno|University]]
[[Category:Educational institutions established in 1899]]
[[Category:Buildings and structures in Brno]]
[[Category:Engineering universities and colleges in the Czech Republic]]
[[Category:1899 establishments in Austria-Hungary]].
```

[25] např. říše, kmen, řád, rod atd.

Stejně jako výše popsaný infobox, slouží tato entita k nalezení článků stejného typu při procesu rozšiřování tréninkové sady. To je popsáno v sekci 4.2

Definiční slova a definiční fráze

Definiční slovo je podstatné jméno, ze kterého je patrné o čem, nebo o kom článek je. Pro ilustraci jsou v tabulce 2.1 uvedeny příklady nadpisů článků anglické Wikipedie a jejich definičních slov. Definiční fráze je pak řetězec obsahující definiční slovo a slova, která jej

Nadpis	Definiční slova
Svratka	river
Brno	city
Leonardo DiCaprio	actor, producer, environmentalist
Google	company

Tabulka 2.1: Příklad definičních slov pro nadpisy různých článků

konkretizují. Typicky jsou to přídavná jména uvedená před ním, u delších definičních frází pak číslovky, předložky a podobně. Příklady definičních frází jsou uvedeny v tabulce 2.2.

Nadpis	Definiční fráze
Svratka	river in the South Moravian Region of the Czech Republic
Brno	second largest city in the Czech Republic
Leonardo DiCaprio	American actor, film producer
Google	American multinational technology company

Tabulka 2.2: Příklad definičních frází pro nadpisy různých článků

Nalezení definičních slov a frází

Vzhledem k tomu, že většina prvních vět článku na Wikipedii má ustálenou typickou strukturu, která je ukázána v tabulce 2.3, je možné z této věty extrahovat definiční slovo a to pak použít při určování typů entit.

Po shlédnutí tubulky 2.3 je možné nabýt dojmu, že by definiční slovo bylo možné extraho-

Nadpis	První věta článku
Svratka	The Svratka, formerly Švarcava is a river in the...
Brno	Brno is the second largest city in the Czech Republic.
Sněžka	Sněžka or Śnieżka is a mountain on the border...
Bleachers (band)	Bleachers is an American indie pop act based in...

Tabulka 2.3: Příklad typické struktury první věty

vat jednoduše, třeba pomocí regulárního výrazu. Tak jednoduše tento problém však řešit nelze. Nalezení definičních slov není triviální záležitost a pro získání korektních výsledků je potřeba využít „pokročilejších metod“. Nyní budou představeny postupy z oblasti zpracování přirozeného textu, které jsou pro tento účel využity. Jak konkrétně je toto „nalezení“ implementováno je pak popsáno v kapitole 4.1.

Tokenizace věty

Spolehlivé nalezení definičního slova ve větě vyžaduje využití hned několika nástrojů. Všechny tyto však nepracují s textem jako takovým, ale s její tokenovou reprezentací. Jak taková reprezentace vypadá, je naznačeno v tabulce 2.4, kde má již každý token také určený slovní druh. Je tedy třeba větu převést na tuto reprezentaci. Tento proces probíhá z pohledu uživatele automaticky, za použití předem natrénovaného statistického modelu, který je součástí externích modulů použitých k tomuto účelu.

Určení slovního druhu

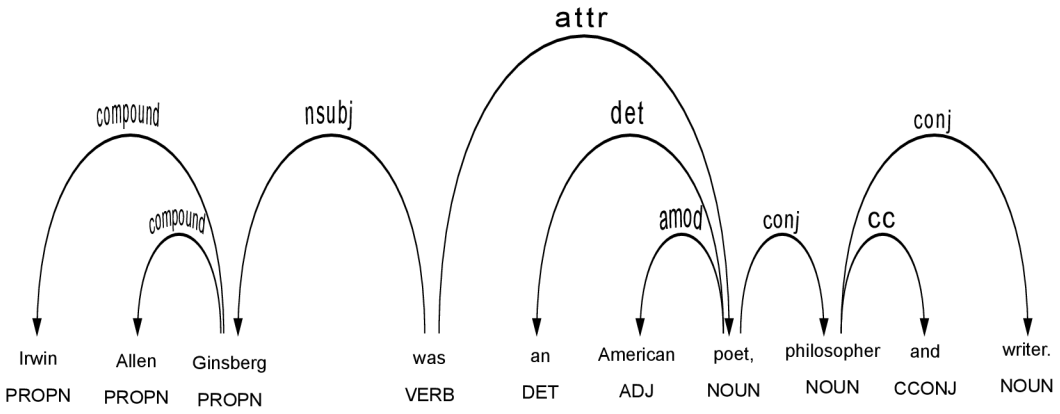
Určení slovního druhu (anglicky „Part-of-speech tagging“, zkráceně také POS tagging) probíhá rovněž automaticky, na základě statistického modelu, kterými použité knihovny disponují. Po tomto procesu je u každého tokenu uložen jeho slovní druh, jak je naznačeno v tabulce 2.4.

Poř. č.	0	1	2	3	4	5	6
Token	Apsines	was	a	sophist	from	Athens	.
Sl. druh	PROPN	VERB	ADP	NOUN	ADP	PROPN	.

Tabulka 2.4: Tokenová reprezentace věty s určenými slovními druhy

Analýza závislostí tokenů

Jak je patrné z obrázku 2.9, definiční slovo je buď přímým potomkem slovesa být, nebo potomek některého potomka slovesa být. Opět na základě statistického modelu je tedy vytvořena reprezentace těchto závislostí (je tedy provedena tzv. závislostní analýza). Tyto závislosti jsou použity při hledání definičního slova, jak je ukázáno v kapitole 4.1.



Obrázek 2.9: Graf sémantických závislostí tokenů ve větě po analýze pomocí knihovny SpaCy

Lemmatizace

Posledním nástrojem využitým při extrakci, je tzv. lemmatizer. Ten na základě nahrazování částí slov, analýzy a případně porovnáváním se slovníkem, transformuje slovo na jeho základní tvar (tzv. lemma) [18][15].

Toto se hodí nejen při hledání slovesa být, kde není nutné každý token porovnávat se všemi možnými tvary, ale hlavně při nalezení definičního slova, protože je vhodnější uložit jeho základní tvar, než ten nalezený ve větě, který může být například v množném čísle. Příklady definičních slov ve větě a jejich základních tvarů jsou uvedeny v tabulce 2.5.
Jak je rovněž vidět v tabulce 2.5, definiční slovo již může být (a typicky také je) v základním

Věta s vyznačeným definičním slovem	Lemma definičního slova
Yurua District is one of the four districts of...	district
Elephants are large mammals of the family...	mammal
New York is a state in the Northeastern United States.	state

Tabulka 2.5: Příklad základních tvarů definičních slov

tvaru. Ne vždy je tedy lemmatizace nutná. Protože však nelze poznat předem, zda nutná bude, provádí se vždy.

Extrakce základních atributů článků

Základními atributy se rozumí vybrané atributy uložené v elementu infobox. Takovými jsou například data narození a úmrtí u osob, rozlohy u jezer, délky u řek, zeměpisné souřadnice u hor a podobně. Všechny tyto jsou extrahovány pomocí regulárních výrazů z textové reprezentace entity infobox popsané v kapitole 2.5. Typicky se takové atributy ukládají ve struktuře známé jako tzv. znalostní graf. Ten v sobě uchovává nejen atributy, ale i vztahy mezi nimi a atributy jiných entit.

Protože však znalostní báze vygenerovaná pomocí systému prezentovaného v této práci slouží především jako index ke sémantickému obohacování textu, není třeba atributy ukládat do takto složité struktury, ale je možné je uchovávat jednoduše na řádku reprezentujícím příslušný článek ve znalostní bázi.

Kapitola 3

Návrh řešení

V této kapitole bude představeno schéma systému, popsány jeho jednotlivé moduly a naznačeno, jak spolu komunikují jeho dílčí části. Také jaké jsou jejich vstupy a výstupy a jak který modul přispívá do celkového výsledku.

3.1 Návrh celého systému až po vygenerování znalostní báze

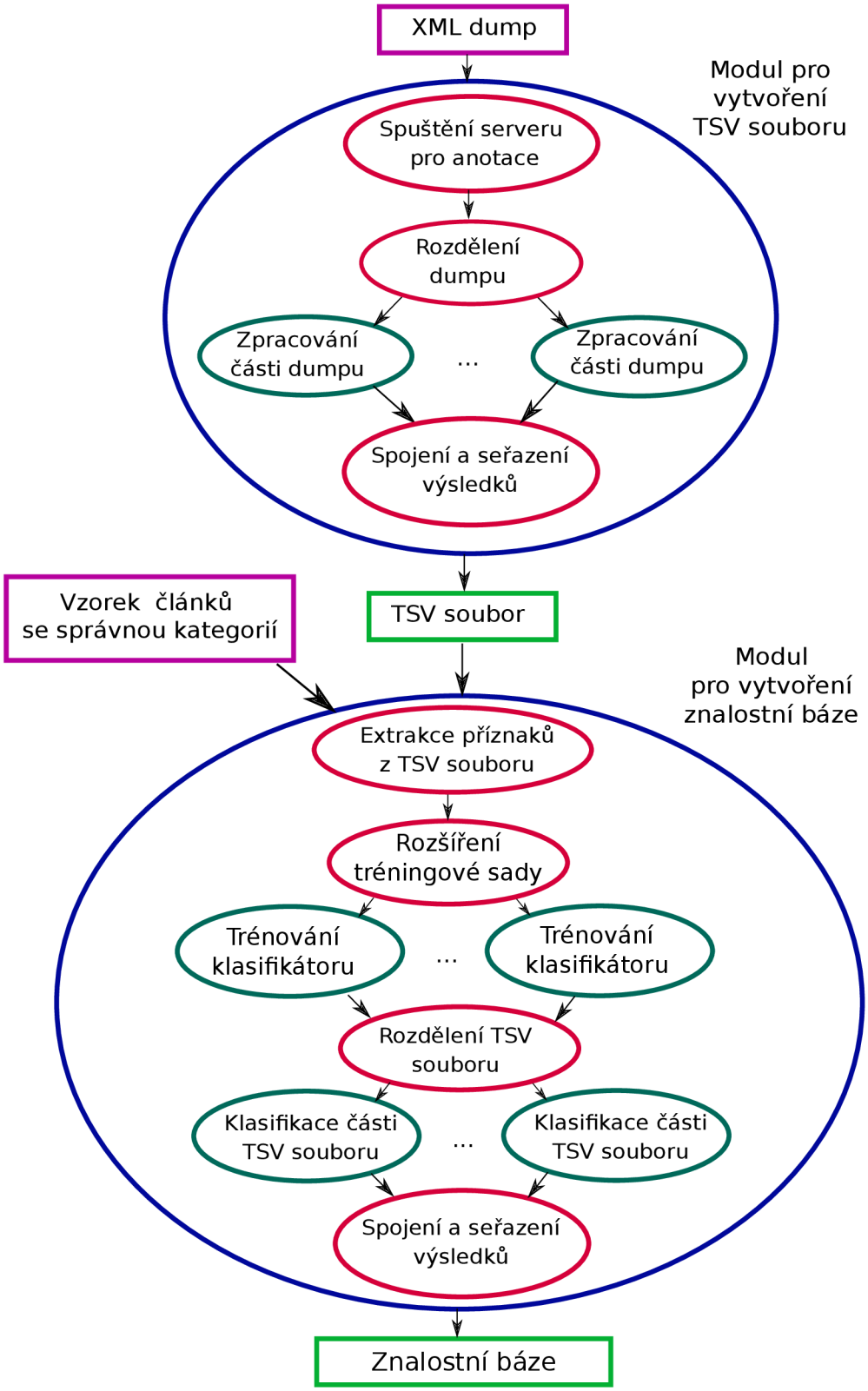
Schéma celého systému od přivedení XML dumpu na vstup, až po vygenerování znalostní báze je možné vidět na obrázku 3.1.

Tmavě modrou jsou označeny jednotlivé moduly, červeně jejich části, které jsou prováděny sekvenčně a světle modře pak ty, které se provádějí paralelně. Fialově jsou označeny vstupy a zeleně výstupy.

TSV soubor je sice označen zeleně, protože je již samostatným výstupem, ale slouží, jak je naznačeno šipkou, zároveň jako vstup pro další modul.

Ze schématu na obrázku 3.1 je patrné, že systém se skládá z dvou hlavních modulů, z nichž první vytvoří z dump souboru TSV soubor a druhý pak z TSV souboru vygeneruje znalostní bázi.

Posledním modulem, který bude popsán, je modul, který detekuje změny mezi dvěma znalostními bázemi. Nyní budou jednotlivé moduly popsány.



Obrázek 3.1: Schéma celého systému až po vygenerování znalostní báze

3.2 Modul pro vytvoření TSV souboru

- Vstup: XML dump se stránkami Wikipedie
- Výstup: TSV soubor se strukturovanými daty článků

Protože bude nyní potřeba analyzovat přirozený text, tedy určovat slovní druhy, sémantické závislosti mezi slovy a podobně, tento modul nejprve zapne Stanford CoreNLP server, aby na něj klienti mohli posílat požadavky na anotaci textu.

Poté je z důvodu urychlení extrakce, na základě zadaných vstupních parametrů rozdělen dump soubor na několik menších částí. Pro každou tuto část se spustí jeden podproces, který ji bude zpracovávat.

Vyčká se na dokončení všech podprocesů a v momentě, kdy toto nastane, jsou jednotlivé části abecedně, dle titulku článku, seřazeny a je vytvořen celý TSV soubor. Ten obsahuje pro každý článek titulek, první větu, první odstavec článku, řetězec reprezentující infobox a také sloupce s příznaky, které budou později použity při určování typů entit.

3.3 Modul pro vytvoření znalostní báze

- Vstup: TSV soubor s články, tréninková sada
- Výstup: TSV soubor se znalostní bází

Tento modul nejprve načte TSV soubor ze vstupu a extrahuje z něj potřebné části, tedy hlavně titulek a první větu článku, což jsou sloupce, které budou vypsány do znalostní báze v nezměněné podobě. Dále se načte řetězec s infoboxem, ze kterého se budou extrahovat atributy. Nakonec se načtou sloupce s příznaky, které budou použity při určování typů entit.

Poté se načte tréninková sada a je provedeno její rozšíření, aby mohly být natrénovány klasifikátory, jak je popsáno v kapitole 4.2.

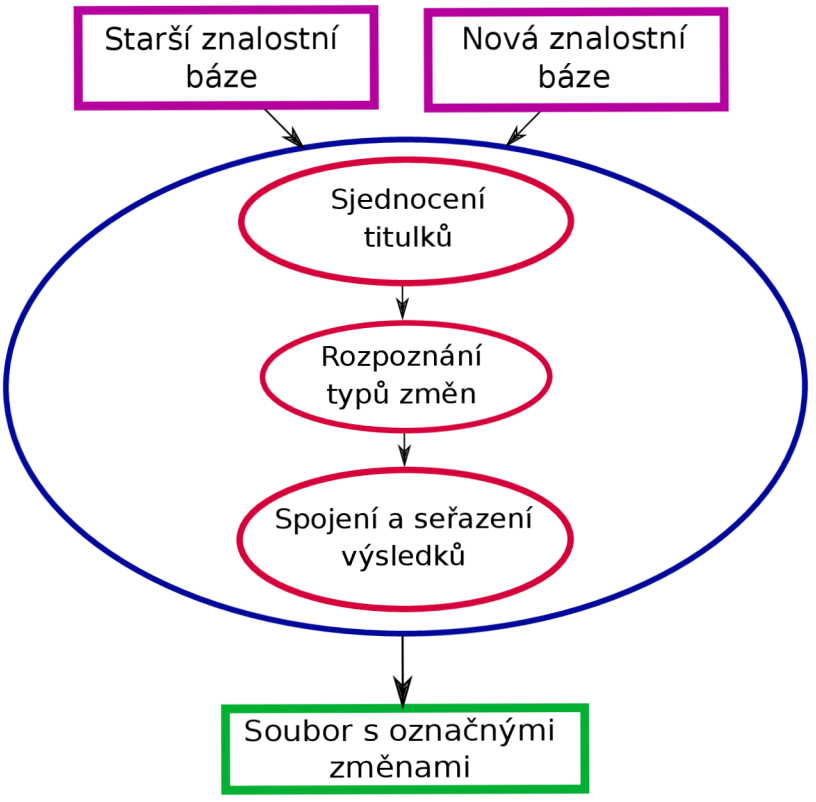
V momentě, kdy je celá tréninková sada připravena, se paralelně natrénují všechny klasifikátory. Protože klasifikace jednoho řádku (článku) pomocí všech klasifikátorů netrvá zanedbatelnou dobu, je i nyní soubor rozdělen na více částí. Ty jsou zpracovávány paralelně.

Pro každý řádek z TSV souboru je provedena klasifikace, jejíž výsledek je společně s titulkem, první větou a atributy zapsán do znalostní báze. Když je zpracování všech řádků hotovo, výsledky se podle titulku abecedně seřadí a vytvoření celé znalostní báze je hotovo.

3.4 Modul pro detekci změn mezi dvěma znalostními bázemi

- Vstup: dvě znalostní báze
- Výstup: soubor s vypsanými změnami

Schéma tohoto systému je na obrázku 3.2. Jak je přesněji popsáno v kapitole 4.3, tento modul nejprve provede sjednocení titulků článku z obou znalostních bází, což umožní jednoduchou detekci přidaných a odstraněných článků. Poté detekuje změny a tyto nakonec vypíše do souboru.



Obrázek 3.2: Schéma modulu pro označní změn

Kapitola 4

Implementace

V této kapitole bude konkrétněji, dle pořadí modulů uvedeného ve schématu 3.1, popsán způsob, jakým je celé řešení implementováno. Nejdříve bude popsáno, jak probíhá vytvoření TSV souboru z dump souboru, poté jak probíhá vygenerování znalostní báze na základě tohoto souboru a souboru s tréninkovou sadou. Nakonec bude popsána implementace modulu pro zjišťování změn mezi dvěma verzemi znalostní báze.

4.1 Vytvoření strukturovaného TSV souboru z dump souboru Wikipedie

Nyní bude popsána struktura TSV souboru, tedy co všechno vlastně obsahuje. Poté bude popsáno, jak je implementován první modul ze schématu 3.1, tedy jak je implementováno vytvoření strukturovaného TSV souboru z dump souboru.

Struktura TSV souboru s články

V TSV souboru je každý článek reprezentován jedním řádkem. Struktura jednoho řádku vypadá následovně:
`Titulek<TAB>Prvni veta<TAB>Nazev infoboxu<TAB>Wikikategorie<TAB>`
`DS Stanford<TAB>DS SpaCy<TAB>DF SpaCy<TAB>Narodnost<TAB>Infobox NF`
`<TAB>Prvni veta NF <TAB>Prvni odstavec<TAB>Prvni odstavec NF`
Kde:

- **Titulek** je neupravený titulek (nadpis) článku.
- **Prvni veta** je čistý text první věty článku bez formátovacích značek.
- **Nazev infoboxu** je název prvního infoboxu nalezeného na stránce.
- **Wikikategorie** je seznam všech kategorií uvedených u článku.
- **DS Stanford** a **DS SpaCy** jsou definiční slova z první věty.
- **DF SpaCy** jsou definiční fráze z první věty.
- **Narodnost** je národnost, typicky u osob, korporací apod.
- **Infobox NF** je řetězec obsahující neformátovaný první infobox v článku.

- **Prvni veta NF** je první věta článku s ponechanými formátovacími značkami.
- **Prvni odstavec** je první odstavec článku bez formátovacích značek.
- **Prvni odstavec NF** je první odstavec článku s ponechanými formátovacími značkami.
- **<TAB>** reprezentuje tabulátor použitý jako oddělovač.

Struktura tohoto souboru je pevná. Tedy i pokud jsou některé ze sloupců prázdné, jejich počet je neměnný.

Extrakce titulku, první věty a odstavce z článku

Jak je vidět v příloze A, nalezení jedné stránky není díky pevné struktuře XML souboru problém. Celý dump, přesněji část náležící jednomu klientovi, je tak postupně zpracovávána. Nejdříve je v dumpu identifikována část reprezentující jeden článek. Z této je nyní nutné extrahovat titulek a obsah článku. Extrakce titulku je triviální, v dumpu je uzavřen v XML značkách `<title></title>`. Nalezení prvního odstavce a první věty už tak jednoduché není. Jak tento proces vypadá, bude nyní popsáno.

V textu postupně probíhá nahrazování a odstraňování nadbytečných částí pomocí regulárních výrazů, až zbude pouze první odstavec. Ten je, na rozdíl od první věty, možné oddělit jednoduše pomocí formátovacích značek. Z toho je nyní nutné extrahovat pouze první větu.

Nejjednodušším přístupem k nalezení první věty by se mohlo zdát rozdělení řetězce na seznam podle znaku tečky. Tento přístup je však velmi naivní a rozhodně jej není možné použít. V odstavci je totiž mnoho zkratek, které tečku obsahují a věta by tak byla „useknutá“. Složitějším, ale lepším přístupem je tokenizace prvního odstavce. Zde je využito faktu, že pokud má tokenizer na vstupu text, který se skládá z více vět, automaticky odstavec rozdělí do seznamu vět, a to i s přihlédnutím ke zkratkám, které jsou přidány do jazykového modelu.

Protože však ne všechny stránky v dump souboru reprezentují články, v tomto okamžiku probíhá také filtrace nežádoucích stránek. Takovými stránkami mohou být například:

- Rozcestníky u mnohoznačných stránek,
- stránky s přesměrováním,
- seznamy (např. seznam sportovců na olympiádě),
- tabulky (např. výsledky na olympiádě).

Protože tyto nepojednávají o žádné konkrétní entitě, není žádoucí, aby se takové stránky objevily v TSV souboru. Z tohoto důvodu bude nyní představeno několik filtračních technik.

Kontrola obsahu titulku

Zde je kontrolováno, zda někde v titulku není přítomen řetězec, který byl v konfiguračním souboru uveden jako nežádoucí. Pokud je, stránka je odfiltrována.

Kontrola začátku první věty

Tento filtr, jak už název napovídá, kontroluje, zda první věta článku nezačíná některým z řetězců uvedených v konfiguračním souboru. Pokud ano, stránka je odfiltrována.

Kontrola obsahu první věty

Filtruje se výskyt nežádoucích řetězců v celé první větě. Pokud je tedy nežádoucí fráze uvedena zde, není ji již třeba uvádět ve filtru pro začátek první věty. Tento filtr je však z principu „agresivnější“ a je tedy často vhodné místo něj použít filtr předchozí.

Kontrola posledního znaku věty

Zde je využito poznatku, že pokud první věta začíná dvojtečkou, je téměř jistě[1] možno ji z dalšího zpracování vyloučit. Toto jsou typicky stránky, které neobsahovaly žádnou z filtrovaných frází v konfiguračním souboru a nebyly tedy zachyceny předchozími filtry, ale o články se nejedná. Tyto stránky tedy do TSV souboru nejsou přidány. Místo toho jsou, aby nebyly „úplně ztraceny“, vypsány do samostatného souboru.

Nalezení infoboxu

Nalezení infoboxu je, pokud jej stránka obsahuje, relativně jednoduchý proces postupného hledání a nahrazování v textu. V tom jsou hledány řezězce „**Infobox**“, „**Taxobox**“, „**Automatic taxobox**“ a „**Speciesbox**“. Pokud je některý z těchto nalezen, uchovají se dvě jeho varianty.

Jedna jako „surový“ řetězec, bez jakéhokoliv formátování (kromě odstranění znaků nových řádů a tabulátorů) a druhá, ve které se uchová pouze název infoboxu v případě obecné varianty, nebo slovo „taxobox“ či „speciesbox“ v případě ostatních variant.

Nalezení všech kategorií článku

Nalezení řetězce obsahující všechny kategorie probíhá opět na základě prostého hledání v textu. Tentokrát je hledán řetězec „[[Category“[2] a za ním nejbližší znak „]“. Mezi těmito se, jak je popsáno v kapitole 2.5 nachází kategorie odděleny znakem „|“. Řetězec je tedy rozdělen dle tohoto znaku do seznamu, odstraní se formátovací značky a proces nalezení kategorií je hotov.

Předzpracování první věty článku před extrakcí definičních slov

Protože se při hledání definičních slov používají nástroje, které pracují s přirozeným textem na základě statistického modelu, je žádoucí, aby byl analyzovaný text, respektive jeho struktura, co nejjednodušší. Před samotnou analýzou textu je tedy první věta ještě předzpracována.

První věty článků obsahují často závorky s dodatečnou informací, jako jsou například data narození a úmrtí u osob, zkratky u měst a také, pro textovou analýzu obzvláště nepříjemné, znaky jiných abeced, než obsahuje použitý jazykový model. Takovými jsou například originální názvy čínských vesnic a podobně. Tyto znaky jsou při použití anglického modelu netokenizovatelné, a působí problémy při dalším zpracování.

Protože se však definiční slovo v závorce u první věty z principu nikdy nevyskytuje[3], závorky i jejich obsah jsou při hledání definičních slov z řetězce dočasně vypuštěny.

[1] 100 % manuálně zkontrolovaných článků z cca 200 vzorků.
[2] Případně jeho jinojazyčná varianta, protože slovo Category je jazykově závislé.
[3] V závorce ani není souvislý text.

Nalezení definičních slov a frází ve větě

Prvním ze dvou nástrojů použitých k extrakci definičních slov je Stanford CoreNLP [9]. Jak je patrné ze schématu 3.1, řídící modul již ve chvíli, kdy se zpracovává Wiki dump, spustil Stanford CoreNLP Server, který nyní naslouchá požadavkům na anotaci textu.

Nejdříve se tedy na server odešle požadavek na anotaci věty. Ten vrátí odpověď ve formátu JSON[4]. Tato odpověď obsahuje všechny informace, které jsou potřeba k extrakci definičních slov. Ta bude nyní popsána.

Prvním krokem k nalezení definičního slova, je nalezení některého z tvarů slovesa být. Pokud jich je ve větě více, tím vhodným pro využití při extrakci, je vždy to první. Poté se postupně kontroluje seznam sémantických závislostí ve větě a hledá se rodičovský token tokenu reprezentujícího sloveso být. Ten je, jak je patrné například z obrázku 2.9, „adeptem“ na definiční slovo. Proběhne kontrola, zda se jedná o podstatné jméno a pokud ano, token je označen za první definiční slovo a jeho základní tvar se přidá do seznamu.

Protože ve větě může být definičních slov samozřejmě více, nyní je třeba zjistit, zda věta nějaké další obsahuje. Jak je opět patrné z obrázku 2.9, tato slova již nemají jako rodiče sloveso být, ale první definiční slovo, tedy slovo „poet“. Opět tedy probíhá iterace přes závislosti a pokud je nalezen token, jenž má jako rodiče uveden token reprezentující poslední definiční slovo v seznamu[5], zkontroluje se kategorie a jeho závislosti. Pokud jde o závislost spojovací, ověří se ještě, zda token označuje podstatné jméno, a pokud ano, je jeho základní tvar přidán do seznamu definičních slov. Nakonec se tato slova seřadí dle abecedy a oddělovačem „|“ spojí do jednoho řetězce.

Nalezení definičního slova pomocí knihovny SpaCy probíhá velmi obdobně. Rozdíl je v podstatě jen v jiné interní reprezentaci závislostí a slovních druhů a také způsobu, jakým je prováděna anotace. Tu Stanford CoreNLP provádí pomocí posílání požadavků na server, kdežto SpaCy pouhým zavoláním metody na analyzovaný text. Při extrakci pomocí knihovny SpaCy je ale hlavním rozdílem to, že pomocí tohoto nástroje jsou hledány také celé definiční fráze. K tomuto účelu slouží objekt „noun chunks“. Do toho jsou při analýze textu automaticky uloženy všechny jmenné fráze[6]. Stačí tedy iterovat přes jejich seznam, nejprve zkontrolovat, zda je tzv. rodičovským tokenem sloveso být, případně pro další chunky pak některý z předchozích chunků zcela analogicky, jako u hledání definičních slov.

Tím je celý proces hotov. Definiční slova pro větu reprezentovanou grafem 2.9 jsou tedy: „poet, philosopher, writer“. Věta také obsahuje jednu víceslovnou definiční frázi, a to frázi: „American poet“.

Po tomto procesu jsou tedy z článku extrahovány všechny žádoucí infromace. V momentě, kdy je toto provedeno se všemi články v dumpu, jsou abecedně seřazeny dle titulku, a poté vypsány do TSV souboru.

4.2 Transformace TSV souboru na znalostní bázi

Nyní bude popsáno, jak je implementován modul, který z TSV souboru vygeneruje znalostní bázi.

Na výstupu minulého modulu byla data pro každý článek tabulátorem rozdělena do dvanácti sloupců. Oproti datům, která jsou uložena na konci celého procesu ve znalostní

[4] https://tools.ietf.org/html/rfc7159

[5] Třetí definiční slovo by tedy mělo jako rodiče uvedené to druhé atd.

[6] https://spacy.io/usage/linguistic-features#noun-chunks

bázi však TSV soubor stále obsahuje velký nadbytek formátovacích značek a informací. Ty je nyní nutno odfiltrovat.

V TSV souboru také zatím pochopitelně nebyl pro články určen typ entity o které pojednávají, ale jsou v něm pouze jednotlivé příznaky (více či méně formátované). V této kapitole také bude popsáno jak se data z TSV souboru extrahují do podoby ve znalostní bázi a poté také, jak probíhá klasifikace článků na základě příznaků.

Extrakce základních atributů z infoboxů

Protože je v TSV souboru uložen řetězec reprezentující celý infobox, je tento nutno rozdělit, do „čistých“ a pro člověka dobře čitelných dat. Toto rozdělení a formátování probíhá hlavně pomocí regulárních výrazů. Zde je největším problémem počet různých možností zápisu stejné informace v infoboxu, který tento jinak jednoduše řešitelný problém, značně komplikuje. V tabulce 4.1 je uvedeno několik reprezentací stejné informace (data narození) v infoboxu různých článků.

Titulek článku	Neformátované datum narození v infoboxu
Mike Hosking	{{Birth date and age|1965|1|24|df=y}}
Lee Albon	{{birth-date and age|7 November 1959}}
Dave Davidson (footballer)	{{Birth date|1905|6|4|df=y}}
Celestine Raalte	1948
Ralph Andrews Barry	November 30, 1883

Tabulka 4.1: Hodnota atributu data narození v řetězci infoboxu

Z tabulky 4.1 je zřejmé, že formátování pouhého jednoho atributu není úplně triviální, protože způsob, jakým se má hodnota správně zapisovat, není unifikován. Z tohoto důvodu je formátování časově velmi náročné. Kromě formátování hodnot atributu je při výpisu do znalostní báze vhodné uvážit ještě další úpravy.

Nevypisovat prázdná pole

Přestože při pohledu na obrázek 2.7 je vidět „pár políček“, textová reprezentace infoboxu v dumpu je typicky mnohem delší. Infoboxy totiž obsahují o mnoho více atributů, než je poté vidět při běžném prohlížení Wikipedie pomocí webového prohlížeče, kde jsou prázdné atributy skryty. Obvykle je tak řetězec atributů dlouhý, ale jen malá část jich obsahuje nějakou neprázdnou hodnotu. Tyto samozřejmě nemá cenu vypisovat. Pokud tedy atribut nemá hodnotu, je zahozen.

Rozdílné soustavy jednotek

Při extrakci atributů jako jsou rozlohy, délky a podobně je nutno dbát na použité jednotky. Ne vždy jsou totiž stejné. Může se tak klidně stát, že v jednom článku bude uvedena např. délka řeky v kilometrech, ale v jiném budou použity míle. Informace o použitých jednotkách je naštěstí v řetězci reprezentujícím infobox zachycena. Protože se do znalostní báze jednotky nevypisují, jsou před vypsáním do znalostní báze unifikovány[7].

[7] V této práci jsou všechny jednotky uvedeny v soustavě SI.

Nevypisovat všechny atributy

Protože se v infoboxech nachází atributů mnoho a jejich formáty jsou různé, nelze vypisovat všechny atributy. Před uložením extrahované hodnoty z atributu je tedy kontrolováno, zda je tento konkrétní atribut v seznamu těch, které by měly být extrahovány.

Klasifikace článků

Výše bylo popsáno, jak probíhá extrakce atributů. Nyní už tedy zbývá „jen“ u článků určit typy entit, o kterých pojednávají. Tento proces bude nyní detailněji popsán.

Jak bylo uvedeno výše, tento modul má na vstupu soubor několika vzorových článků, u kterých byl manuálně určen typ entity o které pojednávají[8]. Je však zřejmé, že pro natrénování SVM klasifikátoru do „použitelné podoby“ je jich potřeba relativně hodně[9]. Nyní tedy bude popsáno, jak probíhá rozšíření této tréninkové sady z několika desítek až stovek článků na několik stovek tisíc.

Protože jsou pro zjednodušení manuální práce v souboru se vzorovou klasifikací pouze dvojice (titulek článku, správný typ), je nutné z této dvojice získat příznaky. Tento proces bude nyní popsán. Mějme na začátku v souboru pro ilustraci například pouze tyto tři dvojice:

- (Władysław Bartoszewski, person),
- (Maumee River, watercourse),
- (Whiting Bay, setttlement).

Tyto jsou v TSV souboru uloženy s následujícími informacemi[10]:

- (Władysław Bartoszewski, politician, 1922 births|2015 deaths|...|Herder Prize recipients, activist|...|writer, activist|...|writer, Polish politician|social activist),
- (Maumee River, river, Rivers of Indiana|Rivers of Ohio|...|Rivers of Henry County,Ohio, river, river,),
- (Whiting Bay, UK place, Villages in the Isle of Arra, village, village,).

Samotný algoritmus pak pracuje následovně:

1. Dvojice ze souboru jsou přidány do seznamu dvojic v paměti.
2. V TSV souboru se postupně hledají řádky reprezentující článek, který má záznam v načtených dvojicích.
3. Když je takový řádek nalezen, přečtou se sloupce s příznaky a uloží do dočasného seznamu.
4. Je zkontrolováno, zda se příznaky infobox, nebo některá z wikicategories neobjevují u více než jednoho typu entity zároveň[11].

[8] Při vývoji bylo použito cca 200 článků.

[9] Při vývoji bylo typicky používáno kolem 20 tisíc článků.

[10] U některých příznaků jsou výpustky z důvodu jejich velkého množství, v TSV souboru jsou samozřejmě uloženy všechny.

[11] Toto je typické např. pro wikicategories jako jsou (rok) establishments, která je obvykle u mnoha typů článků. Takové kategorie pochopitelně není vhodné používat pro další odvozování.

5. Pokud se objevuje pouze u jednoho typu, jsou příznaky přidány do seznamu klasifikovaných příznaků.

Nyní je tedy namísto dvojic (Maumee River, watercourse), (Władysław Bartoszewski, person) a (Whiting Bay, settlement) k dispozici seznam extrahovaných příznaků pro každý zmíněný typ, tedy:

- watercourse: (infobox: (river), wikicategories: (Rivers of Indiana, Rivers of Ohio, ..., Rivers of Henry County,Ohio), defwordStanford: (river), defwordSpaCy: (river), defphrSpaCy: ()),
- person: (infobox: (politician), wikicategories: (1922 births, 2015 deaths, ..., Herder Prize recipients), defwordStanford: (activist, ..., writer), defwordSpaCy: (activist, ..., writer), defphrSpaCy: (Polish politician, social activist)),
- setttlement: (infobox: (UK place), wikicategories: (Villages in the Isle of Arra), defwordStanford: (village), defwordSpaCy: (village), defphrSpaCy: ()).

V tuto chvíli je tedy k dispozici seznam s extrahovanými příznaky pro všechny záznamy se vzorovými klasifikacemi. Na základě těchto bude nyní tréninková sada rozšiřována následujícím postupem:

1. TSV soubor se prochází a pro každý článek se kontroluje sloupec s příznaky infoboxu a wikicategories.
2. Pokud je hodnota některého z těchto shodná s hodnotou v již klasifikovaných (přičemž typ u infoboxu a wikicategories se nesmí rozcházet), je titulek tohoto článku s příslušným typem přidán do dříve zmíněného seznamu dvojic.
3. Tímto je dokončena první iterace rozšiřování tréninkové sady.

Další iterace probíhají obdobně, s tím rozdílem, že na začátku mají k dipozici rozsáhlejší seznam dvojic. Algoritmus tedy obecně pracuje takto:

1. Je k dispozici seznam dvojic (počáteční ze souboru nebo rozšířený z předchozí iterace)
2. Prohledává se TSV soubor, a pokud je nalezen článek s titulkem, který je v seznamu dvojic, příznaky které jsou u něj uvedeny se uloží do dočasného seznamu
3. U příznaků infobox a wikicategories je zkontrolováno, zda by po přidání do typu zmíněného ve dvojici, nebyly u několika různých typů najednou.
4. Pokud ne, jsou přidány do seznamu klasifikovaných příznaků
5. Nyní se opět prochází TSV soubor a pro každý článek se kontroluje hodnota příznaku infobox a wikicategories
6. Pokud je některá z hodnot mezi klasifikovanými příznaky, je dvojice (titulek, typ entity) přidána do seznamu dvojic (který tedy bude pro další iteraci obsahovat více hodnot)
7. Pokud nebyl dosažen limit iterací, pokračuje se opět od bodu 1 s rozšířených seznamem dvojic

Na konci tohoto procesu je tedy k dispozici namísto dvojic (titulek, typ entity) seznam mnoha nalezených příznaků pro každý typ. Aby bylo pokrytí lepší, je však samozřejmě vhodné udělat iterací více. Z důvodu snížení počtu nutných iterací a zlepšení pokrytí, jsou také k dispozici tři filtry, které je možné použít pro předzpracování příznaku wikicategories. Těmi jsou:

1. Odstranění čísel (toto se hodí typicky pro odstranění letopočtů z řetězců),
2. odstranění národností,
3. odstranění zemí.

Tímto způsobem lze tedy hodnoty příznaku wikicategories zobecnit a pro odvození stejného počtu příznaků je třeba méně iterací.

V závislosti na počtu iterací a článcích klasifikovaných na začátku, je nyní k dispozici rozsáhlá tréninková sada, čítající mnoho stovek tisíc klasifikovaných článků[12]. Protože však není reálné trénovat pět klasifikátorů na takto rozsáhlé tréninkové sadě, je ji nyní nutno naopak zmenšit. Toto „zmenšení po zvětšení“ může na první pohled působit podivně. Je třeba si ale uvědomit, že pokud by se při limitu n článků přímočaře použilo prvních n řádků z TSV souboru, rozhodně by se nejednalo o reprezentativní vzorek dat a jak učení, tak i následná klasifikace by mohly být značně zkresleny.

Vytvoření tréninkové sady ze seznamu příznaků

Nyní bude představeno vytvoření tréninkové sady na konkrétních datech. Jak bylo popsáno výše, v tuto chvíli je k dispozici seznam mnoha hodnot příznaků pro každý typ entity. Nyní je ještě vhodné datovou sadu před samotným trénováním upravit a vyřadit z ní anomálie, které se do ní dostaly například vlivem chyb statistického modelu. Pokud by třeba u typu „Person“ bylo v příznaku defwordStanford (tedy příznaku definiční slovo nalezené pomocí Stanford CoreNLP) chybně nalezeno slovo „musical“, které zjevně osobu necharakterizuje, je možné jej na základně nízké četnosti, odfiltrovat. Tato filtrace probíhá na základě dvou hodnot:

Ponechání slov s největším pokrytím

Tento filtr funguje na principu počítání, kolikrát se který příznak v seznamu vyskytuje a z toho odvodí, kolik procent typu je schopen sám pokrýt. Mějme například tento seznam hodnot příznaku defwordStanford:
„writer,writer,painter,writer,scientist,scientist,writer,painter,musical“.
Pokrytí by přibližně odpovídalo hodnotám uvedeným v tabulce 4.2.
Pokud by tedy tento filtr byl nastaven na prahovou hodnotu 88 %, slovo „musical“by se

Slovo	Pokrytí [%]
writer	44
scientist	22
painter	22
musical	12

Tabulka 4.2: Přibližné pokrytí jednoho typu slovy

[12] pro anglickou Wikipedii

s jeho nejnižším pokrytím 12 % již do slov, které budou v seznamu ponechány, „nevešlo“.

Aby však v tomto příkladu bylo odstraněno chybně nalezené slovo „musical“ ze seznamu, bylo by nutné „zahodit“ slova, pokrývající 12 % celého typu. Takový filtr je zjevně příliš agresivní.

Je však nutné si uvědomit, že zde uvedený seznam byl „uměle vytvořen“ pouze pro ilustraci[13]. U „opravdových hodnot“ je pochopitelně seznam mnohem delší a při vytváření znalostní báze je pokrytí nesprávně přidaných slov řádově ve zlomcích procent. Už nastavení filtru třeba na 99 % značnou část z nich spolehlivě odfiltruje.

Odfiltrování slov s nízkou četností

Druhý filtr pracuje na jednoduchém principu. Vzhledem k tomu, že jsou u hodnot příznaků k dispozici nejen hodnoty pokrytí, ale samozřejmě také informace o jejich absolutní četnosti, je možné také každému příznaku nastavit prahovou hodnotu pro odfiltrování. Pokud pak bude četnost této hodnoty příznaku pod nastavenou prahovou hodnotou, bude zahozena.

Formátování tréninkových dat

Nyní jsou tedy data jako taková připravena a zbývá je pouze převést do nějaké podoby, která bude moci být na vstupu klasifikátoru při učení. Je tedy třeba z tréninkových dat vytvořit dva vektory. Ty se často označují jako „data“ a „target“, nebo „input“ a „target“ [5], z nichž první bude obsahovat hodnoty příznaků a druhý jejich správně určené typy. V tabulce 4.3 je vidět tvar, do kterého jsou jednotlivé příznaky převedeny. V tabulce 4.4 je pak uveden konkrétní příklad.

data	target
hodnota_priznaku1	spravna_kategorie
hodnota_priznaku_2	spravna_kategorie
⋮	⋮
hodnota_priznaku_m	spravna_kategorie

Tabulka 4.3: Obecný formát tréninkových dat

data	target
settlement	settlement
painting	artwork
poet	person
judge	person
⋮	⋮
cartoonist	person

Tabulka 4.4: Příklad tréninkových dat pro příznak defwordStanford

[13]Seznam vygenerovaný při klasifikaci je pro uvedení samozřejmě zde příliš dlouhý.

Trénování klasifikátorů

Když je připravena tréninková sada, je vzhledem k použití externího modulu trénování klasifikátoru z pohledu uživatele, triviální.

Pro každý z klasifikátorů je po provedení kroků popsaných v kapitole 2.4, paralelně spuštěna funkce pro natrénování. Protože je pro klasifikaci jednoho článku využito pěti různých klasifikátorů, ani predikce typu entity pro jeden článek netrvá zanedbatelnou dobu.

V momentě, kdy jsou všechny klasifikátory připraveny, je tedy celý TSV soubor rozdělen na několik menších částí, aby mohly tyto být zpracovávány paralelně.

Klasifikace článků

Zpracování jednoho článku pak probíhá následovně:

1. Ze sloupce obsahujícího řetězec s infoboxem jsou extrahovány atributy pomocí regulárních výrazů. Tyto budou později zapsány do znalostní báze.
2. Pro každý příznak s nenulovou hodnotou je provedena predikce typu entity příslušným klasifikátorem.
3. Výsledky predikcí jednotlivých klasifikátorů jsou poté shromážděny.
4. Dle vah nastavených jednotlivým klasifikátorům je rozhodnuto, který typ bude nakonec článku přiřazen.

Při postupu klasifikace byly zmíněny váhy příznaků, respektive výsledků jednotlivých klasifikátorů. Ty jsou uvedeny v tabulce 4.5. Hodnoty u jednotlivých příznaků byly vybrány

Příznak	váha
infobox	11
wikicategories	1
defwordStanford	2
defwordSpaCy	2
defphrSpaCy	5

Tabulka 4.5: Váhy jednotlivých příznaků použité při klasifikaci

tak, aby byly splněny vtahy 4.1 a 4.2.

$$V_{infobox} > V_{wikicategories} + V_{defwordStanford} + V_{defwordSpaCy} + V_{defphrSpaCy} \tag{4.1}$$

Vztah 4.1 zaručuje , že příznak infoboxu „přebije“ všechny ostatní. Toto je žádoucí, protože u klasifikace pomocí infoboxů je typicky generována (až na dále popsanou výjimku) téměř nulová chyba. Pokud tedy článek má infobox který lze klasifikovat, je výsledek této klasifikace upřednostněn před všemi ostatními.

$$V_{defphrSpaCy} > V_{defwordStanford} + V_{defwordSpaCy} \tag{4.2}$$

Pomocí vztahu 4.2 je pak zaručeno, že i když budou klasifikátory definičních slov predikovat cokoliv, pokud bude možné určit typ pomocí víceslovné definiční fráze, bude upřednostněna. Důvodem pro tento vztah je fakt, že pokud je k dispozici delší definiční fráze, mohou být odlišena mnohoznačná slova, která by jinak byla považována za totožná.

Například u vět „The watt is a unit of power.“ a „A platoon is a military unit typically composed of two or more squads. . . “, je definičním slovem v obou případech slovo „unit“.

Přestože v nich má toto slovo naprosto rozdílný význam, bez zohlednění kontextu to nelze poznat.

Když by však byly články klasifikovány pomocí celé definiční fráze, která by v případě první věty byla „unit of power“ a v případě věty druhé „military unit“, rozdělit by je najednou nebyl problém. V tomto je tedy definiční fráze spolehlivější a výsledky získané pomocí něj, jsou upřednostněny.

Příznak wikicategories je možné dobře použít při odvozování dalších článků do tréninkové sady, jak bylo popsáno výše. Při klasifikaci jej však, především díky množství různých kategorií a jejich možným kombinacím, je vhodné používat spíše jako pomocný, pokud nelze typ entity v článku určit pomocí jiných klasifikátorů. Jeho váha je tedy stanovena na 1.

Pokud tedy některý z klasifikátorů detekuje některý typ, jemu přidělená váha je přičtena do „skóre“ tohoto typu. Jako výsledný typ entity v článku je pak zvolen ten, který má toto skóre nejvyšší. Při pohledu na tabulku 4.5 a v ní uvedené váhy je také zřejmé, že může výjimečně nastat i situace, kdy budou mít dva typy stejné skóre a tedy nebude možné jednoznačně rozhodnout o výsledném typu. V takovém případě zůstane článek bez určeného typu.

Jak bylo uvedeno ve vztahu 4.1, typicky je klasifikace pomocí infoboxu nejpřesnější, a klasifikátor tohoto příznaku má proto největší váhu. Najdou se ale i případy[14], kde je toto spíše na škodu a určování typů pomocí příznaku infobox je naopak horší. Je tedy možné zapnout mód, ve kterém je typ entity určen pouze tehdy, pokud se žádné z klasifikátorů nerozcházejí v predikci typu. Při použití tohoto módu je sice ve znalostní bázi klasifikováno entit o něco méně, ale správnost je mírně vyšší. Rozdíly mezi výsledky dosaženými pomocí těchto dvou různých módů jsou také patrné z kapitoly 5.

Shromáždění výsledků a vytvoření celé znalostní báze

Když jsou takto připraveny všechny řádky z TSV souboru, analogicky jako v 4.1 se abecedně, tentokrát podle zjištěného typu entity, články seřadí a poté jsou vypsány do TSV souboru se znalostní bází.

4.3 Zjištění změn mezi dvěma verzemi znalostní báze

V této sekci bude popsáno, jak je implementován modul, který označuje změny mezi znalostními bázemi vytvořenými z jiné verze dump souboru. Typické využití je při vydání nového dumpu souboru, kdy je z něj vygenerovaná znalostní báze.

V této znalostní bázi mohly některé články přibýt, jiné mohly být naopak smazány. Pokud byl článek ve starší znalostní bázi a je i v té nové, zjišťuje se, zda se u něj neudála nějaká změna.

Touto změnou může být buď změna typu, což by znamenalo, že pro řádek v TSV souboru, který reprezentuje tento článek, vyšla jinak extrakce příznaků, nebo změna první věty článku, či některého z atributů, tedy například přidání data úmrtí u osob, nebo oprava některého z atributu, pokud byla jeho hodnota v minulých verzích chybná.

Postup zjišťování takovýchto změn je jednoduchý:

[14] Typicky u typů s menšími počty entit

1. Obě znalostní báze jsou nečteny do paměti
2. Z obou znalostních bází se vyberou titulky článků a provede se jejich sjednocení

V momentě kdy je toto provedeno, je postup nalezení nově přidaných nebo naopak smazaných článků, triviální.

- Titulek je ve sjednocené množině, ale není v první ⇒ článek je nový.
- Titulek je ve sjednocené množině, ale není v druhé ⇒ článek byl smazán.
- Titulek se nachází ve všech množinách ⇒ článek byl v obou verzích znalostní báze.

Pokud nastal třetí případ, jsou provedeny dvě kontroly. Pomocí první je zjištěno, zda se nezměnil typ článku, tj. příznaky v TSV souboru byly změněny, tedy bylo například přidáno nové definiční slovo nebo upřesněna definiční fráze. Pomocí druhé kontroly se pak zjišťuje, zda se nezměnila některá z informací uvedených v infoboxu, ze kterého jsou atributy extrahovány. Všechny změny jsou i s titulky článků kterých se týkají, vypsány do souboru.

Kapitola 5

Dosažené výsledky a srovnání s jinými řešeními

Nyní budou představeny srovnání pomocí metodiky popsané v kapitole 2.3. Nejdříve budou představeny výsledky ve formě statistik dosažené pomocí nástrojů představených v kapitole 2.2. Poté budou prezentovány výsledky dosažené pomocí systému prezentovaného v této práci, a nakonec budou všechny srovnány.

5.1 DBpedia

Typ entity	Nalezeno entit	Správně určeno [%]
Person	1 847 608	98
Organisation	495 298	95
Event	133 991	100
Organism	344 633	100
Fictional character	23 994	99
Body of water	54 065	100

Tabulka 5.1: Výsledky dosažené pomocí nástroje DBpedia

Z tabulky 5.1 je patrné, že nejhorší výsledky měl tento nástroj u entity typu „Organisation“, kde byly nesprávně zařazovány články o entitách z oblastí informačních technologií a fyziky. U typu „Person“ byl pak zařazen článek o hrdinovi řecké z mytologie a články o fyzikálních pojmech. U typu „Fictional character“, pak byl dvakrát chybně uveden seznam, který sice pojednával o postavách ze seriálu, nicméně o článek popisující konkrétní entitu, tedy například jednu určitou postavu, se nejednalo.

5.2 DBpedia verze Live

I přes to, že DBpedia Live pracuje s aktuálnějšími daty, při pohledu na tabulku 5.2 nelze říci, že by výsledky byly nutně lepší. Důvodem, proč k tomuto jevu dochází, je pravděpodobně odlišný algoritmus použitý pro určování typů entit. Tomu napovídají i chyby, kterých se tento nástroj dopustil a které se v „normální“ verzi DBpedie, nevyskytovaly. Ty jsou popsány dále. Pravděpodobně Nejhorší jsou u entity typu „Event“, kde byl jako entita tohoto typu často chybně označen vesmírný satelit. Tato chyba vznikla pravděpodobně proto, že

Typ entity	Nalezeno entit	Správně určeno [%]
Person	1 018 645	98.5
Organisation	269 847	99
Event	82 519	95,5
Organism	250 070	100
Fictional character	13 045	98.5
Body of water	35 683	100

Tabulka 5.2: Výsledky dosažené pomocí Live verze nástroje DBpedia

články o vesmírných satelitech často obsahují stejný typ infoboxu, jako ty o vesmírných misích[1]. K entitám typu „Person“ a „Organisation“ byly naopak chybně přiřazeny články o událostech. Do typu „Fictional character“, byly nesprávně zařazeny články o rodině mnichů, básni a eposech.

5.3 Wikidata

Typ entity	Nalezeno entit	Správně určeno [%]
Person	5 107 330	99
Organisation	1 682 504	90.5
Body of water	1 405 860	100
Event	606 990	93,5
Fictional character	98 455	100
Organism	27 547	100

Tabulka 5.3: Výsledky dosažené pomocí nástroje Wikidata

Počty nalezených entit uvedené v tabulce 5.3 vypadají u většiny typů o poznání lépe, než v případě nástroje DBpedia. Především u typů „Person“ a „Body of water“ je počet určených entit o poznání vyšší, než v případě předchozích nástrojů. U typů „Organism“ a „Fictional character“, je sice entit méně, ale přesnost je zde stoprocentní. Delší komentář si však zaslouží typy „Organisation“ a „Event“. U těchto je správnost určení horší. Nejhorší výsledky byly u typu „Organisation“. Zde byla zařazována například muzea, která zde jistě patřit mohou. Dále však také památníky, a to jak velké památníky, které jsou svou povahou spíše muzeem a jejich zařazení by tak opět bylo v pořádku, tak i ty malé, třeba v podobě „kapličky u cesty“. Ty již za organizaci jistě považovat nelze. Také zde byla řazena například švýcarská města, několik švédských vesnic a podobně. Nejinak na tom byl typ „Event“, do kterého byly zařazovány videohry, typy softwarových licencí a jednou také stroj v podobě pohyblivého modelu pavouka.

Oproti výsledkům dosažených pomocí DBpedie však díky vyšším číslům u nalezených entit, statistiky působí, že je tento nástroj „lepší“. Zde je však nutno podotknout, že zatímco u DBpedie je u entity uvedeno typicky několik faktů a minimálně jeden zdroj, v případě nástroje Wikidata toto často neplatí. Nezřídka kdy je jako entita prezentována stránka, která nemá ani jméno a obsahuje pouze identifikátor ve tvaru například „Q57347688“. Ke jménu se lze v lepším případě dostat přes odkaz uvedený na stránce, ale i ten někdy chybí

[1] Vzhledem k tomu, že se tato chyba vyskytuje jen ve verzi Live, je možné, že je zde větší snaha klasifikovat články na základě typu infoboxu.

a není tak jasné o jakou entitu se jedná, ani „odkud se v projektu Wikidata vzala“. Také je nutné brát v úvahu počet zdrojů ze kterých tento nástroj entity určuje. Ten je, jak bylo uvedeno v kapitole 2.3 u tohoto systému, vyšší.

5.4 Systém NER vyvíjený na FIT VUT (cz)

Typ entity	Nalezeno entit	Správně určeno [%]
Person	107 343	100
Settlement	36 834	100
Relief	6 377	94.5
Watercourse	4 844	100
Waterarea	2 611	100
Fictional character	1 435	94
Island	1 038	99.5
Country	212	100
Waterfall	110	100
Peninsula	100	100

Tabulka 5.4: Výsledky dosažené pomocí nástroje NER (cz)

Při pohledu na tabulku 5.4 je předně nutno zdůvodnit nízký počet nalezených entit. Ten je způsoben tím, že tento nástroj pracuje pouze s českou jazykovou mutací Wikipedie. Na té je článků samozřejmě o poznání méně, než na anglické[2]. Přesnost tohoto systému je však až na dvě výjimky, velmi dobrá. U typu „Fictional character“ byly výsledky horší, neboť byly k tomuto typu zařazeny historické postavy, jeden seznam postav a dva eposy. U typu „Relief“ se pak objevovaly články o geomorfologickém uzemním celku, které však nebyly horou, ani vrcholem. U kategorie „Island“ byl pak jako ostrov klasifikován jeden poloostrov, který však má vlastní kategorii. Tyto chyby lze však jistě tolerovat spíše než zařazení stroje mezi události.

5.5 Systém NER vyvíjený na FIT VUT (en)

Typ entity	Nalezeno entit	Správně určeno [%]
Person	3 299 035	100
Location	950 305	100
Artwork	299 733	89.5
Event	83 974	94

Tabulka 5.5: Výsledky dosažené pomocí nástroje NER (en)

V tabulce 5.5 jsou uvedeny jen čtyři typy entit. Tento systém totiž rozpoznává jiné typy entit než verze, která pracuje s českou Wikipedií a systém prezentovaný v této práci. Například v typu „location“ jsou tak zahrnuty všechny přírodní lokace a také vesnice. Předností tohoto systému je velmi dobrá identifikace entit typu „Person“. Správnost u typu

[2]Protože však typy entit používané systémem prezentovaným v této práci vycházely hlavně z tohoto systému, je zde pro zajímavost uveden.

„location“ je rovněž velmi dobrá. Je však nutno poznamenat, že tento typ je velmi obecný a byla do něj započítávána například muzea, která v ostatních systémech spadají do typu „Organisation“. U entit typu event byly někdy určovány sportovní kluby a umělecká díla. U typu Artwork byly pak často uváděni lidé.

5.6 Výsledky systému prezentovaného v této práci

TSV soubor

I přes to, že jde v kontextu celého zde prezentovaného systému spíše o „mezikrok“, za samotný výsledek lze jistě považovat i TSV soubor s extrahovanými příznaky, prvními větami a odstavci. Ten obsahuje pro dump Wikipedie z května 2019 přes 5,2 miliónů zpracovaných článků. Z toho u 99,9 % byl extrahován alespoň jeden příznak. Po odečtení příznaku wikicategories který se používá spíše jako pomocný, je pro 94,5 % článků k dispozici alespoň jeden příznak, pomocí kterého lze určit typ entity. Z těchto článků pak obsahuje 88,8 % alespoň jeden příznak získaný pomocí analýzy přirozeného textu. Pro ilustraci je v příloze B uveden seznam nejčastěji nacházených definičních slov.

Znalostní báze s určenými typy entit

Jak bylo popsáno v kapitole 4.2, při vytváření znalostní báze jsou k dipozici filtry, jejichž nastavení ovlivňuje poměr mezi počtem nalezených entit a správností určení jejich typů. Tímto nastavováním lze sice mírně ovlivnit výsledek, při vyhodnocování výsledků se však jako hlavní problém ukázalo to, že i když je postup, kdy infobox „přebije“ všechny ostatní klasifikátory obecně užitečný, u některých typů entit je toto na škodu. Lze tedy zapnout i mód, ve kterém je entita klasifikována, pouze pokud nejsou žádné klasifikátory navzájem v rozporu. Z tohoto důvodu budou představeny dva výsledky. Jeden, kdy je tento mód vypnutý a druhý, kdy je naopak aktivní.

Typ entity	Nalezeno entit	Správně určeno [%]
Person	1 471 354	98
Settlement	502 572	100
Organism	356 927	98
Organisation	426 989	94.5
Relief	48 751	92
Watercourse	49 655	93
Fictional character	5 774	98
Island	6 085	99.5
Event	210 389	97.5
Artwork	575 970	98.5

Tabulka 5.6: Výsledky běžného módu

Jak je vidět při srovnání tabulek 5.6 a 5.7, typy entit, které byl systém schopen určit, se liší. Zde je pravděpodobně největší slabina tohoto systému. Některé z určovaných typů entit totiž nedosahují „použitelné“ přesnosti a v tabulkách tedy nejsou uvedeny. Protože extrahované příznaky jsou v obou případech stejné, chyba musela vzniknout při klasifikaci pomocí SVM. Hlavní důvod tohoto nedostatku je pravděpodobně ve volbě parametrů klasifikátoru. Ty byly sice nastaveny na základě porovnání výsledků různých nastavení tak,

aby byly výsledky co nejlepší, při použití sofistikovanějšího přístupu jako je například grid search[3], by však pravděpodobně byly lepší. Zde tedy systém ve své aktuální verzi „naráží na své limity“.

Při pominutí tohoto nedostatku lze však při pohledu na tabulku 5.7 vidět, že u některých typů podávají lepší výsledky ostatní nástroje, u některých typů je však zde prezentovaný systém, předčí. Je také nutné mít na paměti, že jak DBpedia tak Wikidata pracuji s více zdroji, než jen s anglickou Wikipedii, jako tento systém.

Typ entity	Nalezeno entit	Správně určeno [%]
Person	1 406 259	99
Settlement	478 618	100
Organism	357 829	100
Organisation	405 008	94.5
Relief	28 604	100
Watercourse	42 725	98
Waterarea	10 735	100
Waterfall	1 122	99.5
Event	219 704	99
Artwork	515 871	100

Tabulka 5.7: Výsledky módu s vynucenou shodou všech klasifikátorů

Obecně lze říci, že tento systém se, až na výše zmíněný problém s některými typy entit, dopouští spíše pochopitelných chyb, jako je například zařazení fiktivní postavy mezi umělecká díla, zařazení vodopádu k vodnímu toku. Někdy však nastala také situace, kdy bylo do typu „Organisation“ zařazeno místo. Vzácně také vznikají chyby z důvodu nejednoznačnosti definičních slov, kdy není k dispozici víceslovná definiční fráze. Systém tedy omylem například zařadil článek s klíčovým slovem „subdivision“ mezi organizace, i když se mělo jednat o místo.

Detekce změn mezi dvěma znalostními bázemi

Jak bylo předesláno v kapitole 3, součástí systému je i modul, který detekuje změny mezi dvěma znalostními bázemi. Nyní budou představeny změny, které byly pomocí tohoto modulu zjištěny. K získání těchto výsledků byly použité znalostní báze vygenerované s použitím totožné konfigurace. První z nich byla vygenerována z dumpu, který vyšel k 1. 4. 2019 a druhá z dumpu o měsíc novějšího, tedy z 1. 5. 2019. Ve starší znalostní bázi bylo celkem 3 604 284 záznamů a v novější 3 594 526 záznamů. Celkem bylo detekováno 476 256 změn. Tyto změny jsou ukázány v tabulce 5.8.

Při pohledu na tabulku 5.8 je možné si všimnout počtu přidaných, tedy nových článků

Přidaných článků	135 064
Smazaných článků	144 820
Články s jinak určeným typem entity	136 189
Články se změnou atributu, nebo první věty	60 183

Tabulka 5.8: Změny ve znalostní bázi vygenerované z dumpů s odstupem jednoho měsíce

[3] https://scikit-learn.org/stable/modules/grid_search.html

ve znalostní bázi a také článků, které byly smazány. U článků s jinak určeným typem je tato změna způsobena změnou některého z příznaků v TSV souboru, tedy například bylo přidáno nové definiční slovo, nebo třeba přidán infobox. Články se změnou atributu, nebo první věty pak jsou ty, u kterých se změnila některá z hodnot uchovávaných ve znalostní bázi. Tedy mohl být některý z atributů opraven, nebo přidán. Tímto přidáním mohlo být například úmrtí u některé osoby. Také mohla být upravena první věta článku, která je rovněž ve znalostní bázi uložena.

Kapitola 6

Závěr

V této práci byl představen systém, který se skládá z několika modulů. První modul zpracuje dump Wikipedie do podoby TSV souboru, kde je pro každý článek extrahován titulek, první věta a odstavec článku a také řetězec, ze kterého je možné extrahovat základní atributy entity, o kterých článek pojednává. Zároveň jsou extrahovány příznaky, které lze použít pro určení typu entity, o které článek pojednává. Ty jsou extrahovány jak ze strukturovaných částí článku, tak i z přirozeného textu, přičemž alespoň jeden příznak, který se používá k určení typu entit, se povedlo extrahovat u 99,9 % článků. Po odečtení článků které obsahují pouze příznak wikicategories, který je v této práci používán spíše jako pomocný, je alespoň jeden příznak extrahován u 94,5 % článků. Výhodou tohoto modulu je také to, že byl navržen takovým způsobem, aby jej bylo možné snadno adaptovat pro použití v různých jazykových mutacích Wikipedie.

Druhý modul pak s použitím malé tréninkové sady, z výše zmíněného TSV souboru vygeneruje znalostní bázi, kde jsou u článku určeny typy entit, o kterých pojednávají, přičemž správnost určení je pro většinu detekovaných typů kolem 99 %. U článků jsou také extrahovány základní atributy, jako například data narození u lidí, délky u řek a podobně. Takto získanou znalostní bázi je možné použít například k sémantickému obohacování textu.

Součástí systému je také modul, který umožní zjistit, jaké změny se udály mezi dvěma verzemi znalostních bází. Ten detekoval u dvou znalostních bází vygenerovaných z dumpu s odstupem jednoho měsíce přes 130 tisíc nových článků. Přes 140 tisíc článků bylo naopak smazáno, u cca 60 tisíc se v průběhu jednoho měsíce změnil některý z atributů a u téměř 140 tisíc vyšla jinak extrakce příznaků a byl jim tedy přiřazen jiný typ.

Systém jako celek je tedy schopen transformovat dump sobor s články Wikipedie na znalostní bázi, ve které je každý článek z dump souboru s určeným typem reprezentován jedním řádkem. Na tom je uveden kromě titulku a první věty také typ entity, o které článek pojednává a její základní atributy. Takovou znalostní bázi je možno použít například k sémantickému obohacování textu. Různé verze znalostní báze lze také pomocí tohoto systému srovnávat a detekovat v nich změny.

Literatura

[1] AI, E.: spaCy: Industrial-Strength Natural Language Processing. [Online; navštíveno 13.01.2019].
URL http://spacy.io

[2] Ameta, D.; Jat, P. M.: Information extraction from wikipedia articles using DeepDive. In *2018 International Conference on Communication information and Computing Technology (ICCICT)*, Feb 2018, s. 1–6, doi:10.1109/ICCICT.2018.8325869.

[3] Beliga, S.; Meštrović, A.; Martinčić-Ipšić, S.: An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences*, ročník 39, č. 1, 2015: s. 1–20.

[4] Bennett, K. P.; Bredensteiner, E. J.: Duality and Geometry in SVM Classifiers. [Online; navštíveno 11.04.2019].
URL https://www.researchgate.net/profile/Kristin_Bennett2/publication/2801700_Duality_and_Geometry_in_SVM_Classifiers/links/00b4951d95825e8c8e000000/Duality-and-Geometry-in-SVM-Classifiers.pdf

[5] Bhattacharjee, J.: Some Key Machine Learning Definitions. [Online; navštíveno 7.05.2019].
URL https://medium.com/technology-nineleaps/some-key-machine-learning-definitions-b524eb6cb48

[6] Chen, D.; Fisch, A.; Weston, J.; aj.: Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[7] Dong, X.; Gabrilovich, E.; Heitz, G.; aj.: Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, New York, NY, USA: ACM, 2014, ISBN 978-1-4503-2956-9, s. 601–610, doi:10.1145/2623330.2623623.
URL http://doi.acm.org/10.1145/2623330.2623623

[8] Du, X.; Cardie, C.: Harvesting Paragraph-level Question-Answer Pairs from Wikipedia. In *ACL*, 2018.

[9] Manning, C. D.; Surdeanu, M.; Bauer, J.; aj.: The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, s. 55–60.
URL http://www.aclweb.org/anthology/P/P14/P14-5010

[10] Meyer, D.; Wien, F. T.: Support vector machines. *The Interface to libsvm in package e1071*, 2015: str. 28.

[11] Miháltz, M.: Information Extraction from Wikipedia Using Pattern Learning. *Acta Cybern.*, ročník 19, 01 2010: s. 677–694.

[12] Nguyen, D. P.; Matsuo, Y.; Ishizuka, M.: Relation extraction from wikipedia using subtree mining. In *Proceedings of the National Conference on Artificial Intelligence*, ročník 22, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, str. 1414.

[13] Ni, J.; Florian, R.: Improving multilingual named entity recognition with wikipedia entity type mapping. *arXiv preprint arXiv:1707.02459*, 2017.

[14] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; aj.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.

[15] Plisson, J.; Lavrac, N.; Mladenic, D.: A Rule based Approach to Word Lemmatization. [Online; navštíveno 7.05.2019].
URL https://www.researchgate.net/profile/Nada_Lavrac/publication/228525639_A_rule_based_approach_to_word_lemmatization/links/546e16dd0cf2bc99c2151cb0/A-rule-based-approach-to-word-lemmatization.pdf

[16] Rusiňák, P.: *Určování typů entit na základě extrakce informací z Wikipedie*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2018.
URL http://www.fit.vutbr.cz/study/DP/BP.php?id=20995

[17] Siddiqi, S.; Sharan, A.: Keyword and keyphrase extraction techniques: a literature review. *International Journal of Computer Applications*, ročník 109, č. 2, 2015.

[18] Straková, J.; Straka, M.; Hajic, J.: Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. [Online; navštíveno 7.05.2019].
URL https://www.aclweb.org/anthology/P14-5003

[19] Suzuki, M.; Matsuda, K.; Sekine, S.; aj.: Fine-Grained Named Entity Classification with Wikipedia Article Vectors. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Oct 2016, s. 483–486, doi:10.1109/WI.2016.0080.

[20] Tan, C.; Wei, F.; Ren, P.; aj.: Entity Linking for Queries by Searching Wikipedia Sentences. In *EMNLP*, 2017.

[21] Weld, D. S.; Hoffmann, R.; Wu, F.: Using Wikipedia to Bootstrap Open Information Extraction. [Online; navštíveno 10.05.2019].
URL http://sigmodrecord.org/publications/sigmodRecord/0812/p062.special.weld.pdf

[22] Weston, J.; Watkins, C.: Support Vector Machines for Multi-Class Pattern Recognition. [Online; navštíveno 7.05.2019].
URL https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es1999-461.pdf

Příloha A

Ukázka dumpu wikipedie

V této příloze je uvedena ukázka XML souboru se stránkami Wikipedie. První stránka je článek a druhá rozcestník.

```
<mediawiki xmlns="http:... version="0.10"xml:lang="en»
<siteinfo>
<sitename>Wikipedia</sitename>
<dbname>enwiki</dbname>
<base>https://en.wikipedia.org/wiki/Main_Page</base>
<generator>MediaWiki 1.33.0-wmf.6</generator>
<case>first-letter</case>
<namespaces>
<namespace key="-2"case="first-letter»Media</namespace>
...
</namespaces>
</siteinfo>
<page>
<title>Allen Ginsberg</title>
...
<text xml:space="preserve»
...
"'Irwin Allen Ginsberg"' ... was an American poet, philosopher and writer.
He is considered to...
</page>
<page>
...
This is a list of characters in [[Ayn Rand]]'s novel "[[Atlas Shrugged]].
...
</text>
<sha1>r2s7mmader52fgob4g7bndelthj793h</sha1>
</revision>
</page>
</mediawiki>
```

Příloha B

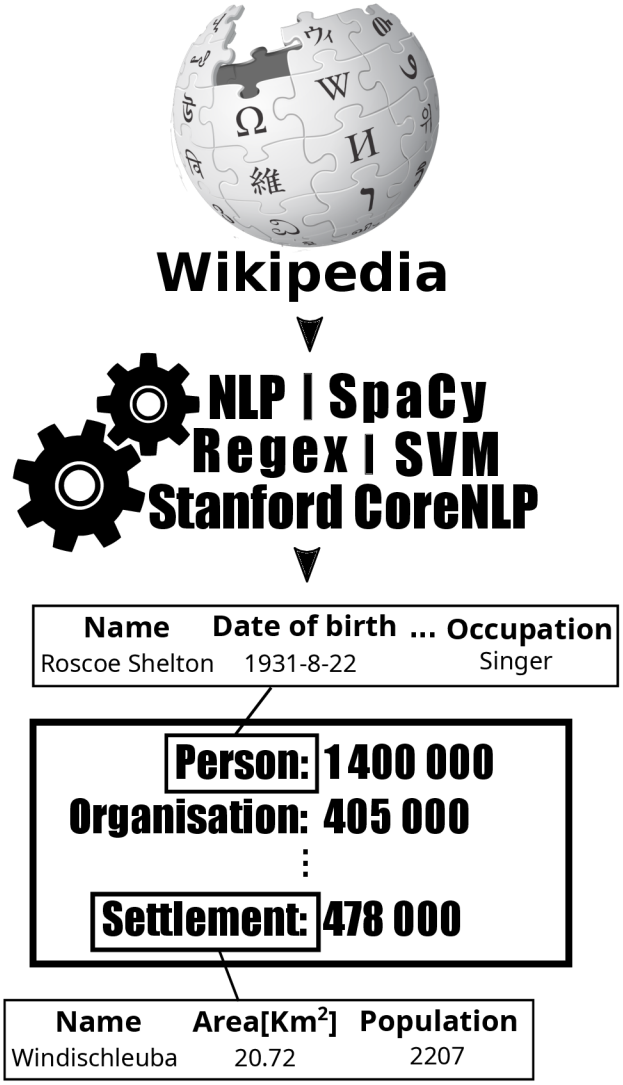
Seznam nejčastějších nalezených definičních slov pro oba nástroje

SpaCy		Stanford CoreNLP	
Četnost	Slovo	Četnost	Slovo
225421	village	226007	village
190076	species	192852	species
131505	politician	131237	politician
127331	player	127217	player
124772	footballer	125888	footballer
121489	film	123744	film
106120	album	103088	album
71805	genus	72272	genus
61790	station	62002	town
59909	town	60843	station
56626	moth	56681	moth
51637	writer	51575	writer
49858	municipality	50056	actor
48067	actor	48937	municipality
46298	commune	47456	commune
44402	community	44022	community
43854	member	39252	district
39882	district	38754	season
38100	season	38599	actress
37837	actress	37395	school
37549	school	34682	member
36321	author	33904	author
35646	series	33807	series
33858	company	33777	artist
33561	artist	33246	company

Tabulka B.1: Seznam nejčastějších nalezených definičních slov pomocí knihovny SpaCy a modulu Stanford CoreNLP

Příloha C

Plakát prezentující tuto práci



Obrázek C.1: Plakát prezentující tuto práci