FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Extraktor informací o firmách z webových zdrojů
Student:	Bc. Tomáš Stanovčák
Vedoucí:	Ing. Jaroslav Kuchař, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

V současné době se na webu prezentuje většina firem a společností. Ne všichni však používají dostupné možnosti pro zpřístupnění informací i ve strojově čitelné podobě. Cílem práce je v co největší míře automatická extrakce informací o společnostech a firmách.

- Seznamte se s problematikou extrakce informací z webu.
- Seznamte se s doménou charakterizující popisy firem a společností.
- Připravte vhodný testovací dataset pro řešený problém.
- Navrhněte vhodné způsoby extrakce (pro minimální množinu informací jako je název, adresa, kontaktní informace, sociální sítě apod.):
  - základní řešení postavené na dostupných metadatech, pravidlech a regulárních výrazech,
  - pokročilejší řešení s využitím strojového učení.
- Proveďte experimenty a vyhodnoťte kvalitu zvolených přístupů.
- Výsledné řešení řádně zdokumentujte a uvolněte pod vhodnou otevřenou licencí

Elektronicky schválil/a Ing. Karel Klouda, Ph.D. dne 2. února 2022 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Extraktor informací o firmách z webových zdrojů

Bc. Tomáš Stanovčák

Katedra aplikované matematiky
Vedoucí práce: Ing. Jaroslav Kuchař, Ph.D.

2. května 2022

Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Jaroslavu Kuchařovi, Ph.D. za odborný přístup, důležité postřehy a čas, který mi věnoval během psaní této práce.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 2. května 2022 . . . . . .

Odkaz na tuto práci

Stanovčák, Tomáš. *Extraktor informací o firmách z webových zdrojů*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Abstrakt

Předmětem této práce je získání a zpracování dat o firmách z jejich webových stránek. Po obeznámení se s přístupy extrakce a množinou dostupných firemních informací bude připraven datový soubor ve vhodném formátu, na kterém budou prováděny experimenty. Tato datová množina bude podrobena rozličným způsobům extrakce na principu pravidel i strojového učení. Výsledky experimentů budou vyhodnoceny a implementace jednotlivých přístupů zveřejněna jako knihovna pod volnou licencí.

Klíčová slova firma, webová stránka, extrakce, vytěžování obsahu, web scraping, zpracování textu, Python

Abstract

The subject of this thesis is to obtain and process company data from their websites. After getting acquainted with extraction approaches and available set of company information, dataset will be prepared in a format suitable for experiments. This dataset will undergo the extraction procedures based on both rule and machine learning principles. The results of the experiments will be evaluated and the implementation of the individual approaches will be publicly accessible as a library under a free licence.

Keywords company, website, extraction, content mining, web scraping, text processing, Python

Obsah

Seznam obrázků

Seznam ukázek kódu

Seznam tabulek

Úvod

Množství volně dostupných informací během posledních let nezastavitelně stoupá. Kvanta údajů na webových stránkách spolu se zlepšujícím se výkonem hardwaru otevřely dveře příležitostem pokročilého zpracování a analýzy dat. To dalo za vznik novým odvětvím výzkumu, ale též i profesím a podnikatelským příležitostem. Jelikož je sběr a úprava dat manuálním přístupem časově náročná, lze využít několik možností jak tenhle proces automatizovat a zjednodušit. Přesně tomuto úkolu se v oblasti firemních stránek bude věnovat tato práce.

Obecně vzato, firmy často nevyužívají možnost označit svoje kontaktní údaje pro účely strojového čtení. Pokud je požadavkem tyto informace i přesto číst strojově, je potřeba využít přístupy založené na pravidlech prohledávajících strukturu webu nebo principech strojového učení. Těmito způsoby lze strojově vyčíst strukturované i nestrukturované údaje, které nejsou na úrovni kódu webové stránky řádně kategorizovány.

Hlavní motivací pro výběr tohoto tématu bylo moje nadšení pro volně dostupná data. Již studium oboru Znalostní inženýrství ve mě vzbudilo zájem o exploraci a zpracování dat, který aktuálně rozšiřuji i ve své profesní kariéře. Rozhodl jsem se proto propojit studium s praxí a věnovat se tématu, které má z mého pohledu budoucnost a praktické uplatnění. Možné budoucí užití přístupů představených v této práci vidím například ve sférách firemního obchodu, kde databáze získaných kontaktů může výrazné ulehčit obchodníkům komunikaci a oslovování nových klientů.

Cíle práce

Cílem práce je automaticky zpracovat informace o firmách z jejich webových stránek. V teoretické části se obeznámím s problematikou vytěžovaní dat z webových stránek, s jejich strukturou i zpracování těchto dat. Pro správnou identifikaci klíčových údajů využiji prostor i pro studium domény informací charakterizující firmy. Praktická část bude rozdělena do několika kroků. Nejprve se

budu věnovat připravě datasetů, na kterých bude následně prováděna extrakce informací, trénování modelů, a též vyhodnocování jednotlivých přístupů. Ve druhém kroku navrhnu konkrétní přístupy, jak data z webových stránek extrahovat. V neposlední řade provedu experimenty, na kterých vyhodnotím spolehlivost a přesnost zkoumaných způsobů extrakce. Konkrétní řešení zpřístupním jako knihovnu pod otevřenou licencí pro další možnosti explorace a rozvoje.

Kapitola 1

Charakteristika firem

Tato kapitola nabídne obeznámení s doménou firem a společností. Bude zaměřena na jejich webové stránky a informace, které dané subjekty identifikují a charakterizují. V detailu vysvětluje rozdíl mezi firmou a společností, popisuje jednotlivé charakterizující prvky a čtenáři přibližuje jejich vlastnosti. Navíc podrobně popisuje využití webové stránky reprezentující jednotlivé společnosti, a též rozlišuje tyto stránky dle jejich typů a změření.

1.1 Definice

V běžných rozhovorech bývají pojmy *firma*, *společnost* a dokonce i *podnikatel* často zaměňovány. Veřejnost nimi obvykle označuje různé formy podnikání. I přes to, že tyto výrazy obvykle vystupují jako synonyma, je mezi nimi patrný rozdíl.

Firma, nebo přesněji **obchodní firma**, označuje název, pod kterým je podnikatel nebo společnost vedena v obchodním rejstříků. Pokud podnikatel není zapsán v obchodním rejstříku, právně jedná pod svým vlastním jménem. V běžných rozhovorech se proto lidé často dopouštějí chyb, když například mluví o spolupráci s firmou Stavebnictví s.r.o., jelikož se jedná pouze o název. Správnou formulací by byla spolupráce se společností Stavebnictví s.r.o. [1].

Obchodní společnost lze definovat jako sdružení jednoho a více podnikatelů, které vykonává svou podnikatelskou činnost pod společným obchodním názvem – firmou. Založení společnosti upravuje písemně uzavřena společenská nebo zakladatelská smlouva. Existuje několik typů společností, kde mezi ty nejčastěji používané patří společnost s ručením omezeným, akciová společnost, veřejná obchodní společnost a komanditní společnost [2].

Podnikatel je dle definice fyzická nebo právnická osoba, která vykonává opakovanou činnost s cílem dosažení zisku. Činí tak samostatně na vlastní odpovědnost [3]. Jedná se tedy o obecnější pojem, vůči obchodní společnosti.

Dle těchto definic lze říct, že se práce bude věnovat webovým stránkám patřícím společnostem a podnikatelům, nikoliv firmám. Navzdory tomu, že jsou

pravidla použití jasně vymezena, v běžné praxi je použití zcela odlišné. Jelikož tato práce není smlouvou, ani nijak odborně nezkoumá odvětví obchodu a práva, lze přistoupit i na použití těchto pojmů obvyklým způsobem. Pro účely této práce je důležitější detailní rozbor všech údajů charakterizující danou společnost nebo podnikatele.

1.2 Charakteristické informace

Společnosti o sobě na webových stránkách sdílí množství informací, díky kterým je může veřejnost identifikovat, kontaktovat nebo dokonce i navštívit. Jelikož se granularita a kvantita těchto informací liší případ od případu, postupným průzkumem lze dospět k závěru, že na webech bývají zpravidla tyto údaje:

Jméno společnosti (firma) – údaj, kterému byla věnována předcházející kapitola. Jeho typickým poznávacím znakem je zkratka typu společnosti, například s.r.o., a.s. nebo i v.o.s. Obyčejně se firma nachází na webu spolu s adresou sídla nebo jiným identifikátorem.

Identifikační číslo (zkráceně IČ) – identifikační číslo fyzické nebo právnické osoby. Tvoří ho osm cifer a používá se k jednoznačné identifikaci podnikatelského subjektu. Důvodem pro zavedení tohoto indikátoru byla častá duplicita občanských jmen a firem. Toto číslo je přiděleno na základě žádosti všem, kdo vyhoví podmínkám a chce v České Republice legálně podnikat. Zákon též klade povinnost toto číslo uvádět na svých webových stránkách, provozovně i účetních dokladech [4].

Adresa sídla – plní zpravidla funkci korespondenční adresy a musí být zapsána v obchodním rejstříku. Disponovat sídlem je jedna z podmínek pro založení společnosti. Pro živnostníky je tato adresa místem podnikání [5].

Adresa provozovny – jedná se o místo, kde společnosti nabízí své služby a zboží. Založení provozovny je podnikatel povinen hlásit živnostenskému úřadu. Navíc je její chod doprovázen povinnostmi, mezi které patří viditelné označení firmou a přiděleným identifikačním číslem. Na rozdíl od sídla je možné provozovat v jedné provozovně více společností, ale také jednu společnost (živnost) ve více provozovnách [5]. To je důvodem, proč lze na stránkách najít vícero adres tohoto typu.

Otevírací hodiny – konkrétní doby a dny, kdy je daná společnost dostupná se často vážou s provozovnami. V některých případech ale může jít i o zákaznické linky, provozní doby skladů nebo jiných objektů s vazbou na podnikání. Pro účely této práce nebude mezi těmito případy rozlišováno. Obecní snahou bude získat přibližné informace o týdenní dostupnosti společnosti jako celku.

E-mail – klíčový bod pro komunikaci společnosti. Používá se jak pro zákaznickou komunikaci, tak pro navazování nových obchodních vztahů. Větší společnosti zpravidla přistupují k založení více e-mailových adres pro různé cílové skupiny dané komunikace: obchodní nabídky, informace nebo například reklamace.

Telefonní číslo – druhý způsob kontaktu využívaný pro okamžitou komunikaci se společností. Jejich zveřejňování nebývá tak časté jako u e-mailů, právě kvůli vysoké časové náročnosti na správu telefonní linky. Stejně jako v případě e-mailové komunikace, někteří podnikatelé přistupují k zřízení separátních telefonních čísel například pro zákaznické linky nebo technickou podporu.

Odkazy na sociální sítě – profily jednotlivých společností na sociální sítích mohou poskytnout další cenné informace pro doplnění charakteristiky. Při pokročilejší analýze lze tyto informace navzájem ověřovat a zvyšovat tak přesnost extrahovaných údajů.

Odlišnosti bývají markantní i ve stylu prezentace a umístnění zmiňovaných prvků. Vliv na rozestavení mívá obvykle záměr, s kterým byl web vytvořen. Dle těchto kategorií lze odpozorovat jisté vzory rozmístěni údajů, které mohou napomáhat v jejich extrakci a identifikaci.

1.3 Webové stránky

Dobře navržená firemní webová stránka je mnohokrát důležitou součástí celkového podnikatelského úspěchů. V rychlém a konkurenčním online prostředí je nutností splňovat několik základních zásad pro tvorbu webu. Mezi první se řadí ulehčení přístupu možným zákazníkům na web. Dobrým zvykem bývá sjednocení jména domény s firmou a optimalizace použitých klíčových slov pro vyhledávání. Podstatná je též optimalizace pro různé prohlížeče a obrazovky, protože s nástupem chytrých telefonů a tabletů se mobilní verze webů dostávají do popředí. Společným znakem dobře navržených webů je i snadná navigace, jasná komunikační strategie a neposledně i přesnost sdělovaných informací [6]. Povinnost uvádět některé informace (firma, IČ, sídlo...) dokonce upravuje i občanský zákoník. Pravidla jsou nastavena stejně jako v případě obchodních listin, faktur nebo smluv [7].

Navzdory několika jmenovaných společných znaků se konkrétní zásady odvíjí od cílové skupiny a druhu webové stránky. Zákazníky, pro které jsou stránky vytvářeny lze rozdělit na dvě velké skupiny podle zaměření samotného podnikání: *B2B* (business to business) a *B2C* (business to customer). Tyto zkratky pocházející z anglických výrazů a vyjadřují, na jakou skupinu dané podnikání cílí. Jsou to buď další podnikatelé, nebo koncový zákazníci. Této skutečnosti se následně přizpůsobuje celá komunikace, strategie prezentace, vzhled a dokonce

i granularita informací. Dalším kritériem, které určuje konečnou podobu webu je cíl, s kterým byl web vytvořen. Mezi časté záměry patří generování nových obchodů, prodej zboží, prezentace společnosti, sdílení vlastního obsahu a mnohé další. Právě tyto faktory nejvíc ovlivňují umístnění hledaných informací pro extrakci.

Ačkoliv umístnění požadovaných informací nemá nikde jasně definována pravidla, bývá dobrým zvykem je umístit na jedno z těchto míst:

- hlavička (ukázka 1.1),
- patička (ukázka 1.2),
- dedikovaná stránka (ukázka 1.3).

Obrázek 1.1: Informace v hlavičce [8]

Obrázek 1.2: Informace v patičce [9]

Obrázek 1.3: Informace na dedikované stránce [10]

Informace v hlavičce a patičce jsou ze své podstaty přístupné na jakékoliv podstránce. U dedikované stránky je možné narazit na dvě případy – informace v plném rozsahu jsou uvedeny již na hlavní stránce (častá praxe u jednostránkových webů), nebo je pro ně vytvořena speciální podstránka

(většinou s titulem *Kontakty* nebo *O nás*). Celkový soubor informací je tak tvořen obsáhlou množinou podstránek daného webu. Je proto důležité, do postupu vyhledávání informací zařadit i krok prohledávání vybraných podstránek aby se zamezilo redukci získaných informací.

Kapitola 2

Zdroje dat

Tato kapitola bude věnována popisu zdrojů dat a formátům jejich uložení. Primární zdroj webových stránek *Firmy.cz* bude čtenáři představen spolu s informacemi, které budou ukládány pro další zpracování. Budou rozděleny do tří datových souborů dle jejich funkce. S důrazem na detail bude také vysvětleno, jakým způsobem budou jednotlivé data formalizovány pro následné experimenty, vyhodnocování a trénování.

2.1 Primární zdroj

Pro zajištění diverzity stránek bylo potřeba obrátit se na jistou databázi nebo katalog, která tyto podnikatele shromažďuje. Jedním z volně přístupných a obsáhlých katalogů v České Republice je web *Firmy.cz*. Jak uvádí provozovatel Seznam a.s., jedná se o katalog ověřených firem a institucí, který je pravidelně aktualizován i o kontakty a uživatelské recenze. Velkou výhodou této databáze je její aktuálnost, přehlednost a obsáhlost. Pro lepší orientaci jsou zalistované subjekty řazeny do kategorií a subkategorií dle jejich oblasti podnikání. Běžný uživatele má tak v nabídce několik způsobů vyhledávání, které lze libovolně kombinovat: omezení se na geografickou oblast nebo oblast z mapy, vyhledávání pomocí textu a filtrování dle kategorií a subkategorií. Navigace je proto velice praktická a přehledná. Navíc lze také využít unifikovaný způsob zasílání poptávek, co může zájemcům o služby zkrátit čas pro oslovení dodavatelů. Portál také podporuje větší zviditelní pro firmy nad rámec standardního seřazení v katalogu. Tato funkce je zpoplatněna a umožňuje se posunout inzerované firmě výše ve zobrazeném pořadí [11].

Z tohoto katalogu budou získávány webové stránky automatizovaným způsobem. Jak konkrétně tento proces probíhá a na jakých principech stojí bude vysvětleno v sekci 3.3.4. Z těchto stránek budou vytvořeny tři datové soubory (tzv. datasety) použité k extrakci, vyhodnocování experimentů a trénování klasifikátorů. Pro zajištění větší diverzity dat budou vybrány všechny hlavní kategorie, z kterých bude následně získáno několik jednotek odkazů na webové

stránky (obecně se bude jednat o první dvě stránky katalogu). Výsledný počet vzorků je odhadován na přibližně 200 unikátních webů, v závislosti od vhodnosti jednotlivých případů (atypické a nefunkční weby budou následně manuálně odstraněny). Už v této fázi práce je nutno zmínit, že počet zkoumaných vzorků je poměrně malý a to zejména z důvodu nutnosti manuální anotace jednotlivých záznamů. Tento proces je vysoce časově náročný, proto se v praxi k daným úkolům využívají týmy lidí a specializované platformy. Jelikož tyto prostředky nebylo možné v rámci vypracování diplomové práce využít kvůli jejich nedostupnosti, je nutno se omezit na menši počet vzorků.

2.1.1 Datový soubor pro extrakci

První datový soubor bude tvořen daty, které budou použity jako zdroj pro extrakci. Součástí budou primárně domovské adresy jednotlivých vybraných firem. Pokud budou k dispozici i dodatečné stránky, na kterých se mohou vyskytovat hledané informace, budou zde též uloženy. Mělo by se jednat o sekce kontaktů, obchodních podmínek, podmínek použití osobních údajů a v neposlední řadě i stránky zaměřené na představení společností. Podstránky zmíněných kategorií budou nalezeny automatizovaně pomocí regulární pravidel. Získané stránky budou organizovány do jednotlivých sloupců v následující podobě:

- **id** – jako identifikační a indexovací údaj webové stránky bude sloužit její adresa a doména ve tvaru `address_domain`;
- **url** – absolutní cesta ve tvaru podléhající konvenci zápisu URL, kterou je možno použít k okamžitému přístupu ke stránce pomocí prohlížeče;
- **encoding** – konkrétní kódování použito pro uložení HTML stránky;
- **content** – stránka v HTML připravena pro extrakci informací.

Konkrétní počty jsou k nalezení v závěru při rozboru získaného datového souboru v tabulce 5.1.

2.1.2 Datový soubor pro experimenty

Druhý datový soubor bude obsahovat všechny hledané informace, které bylo možné získat průzkumem uložených webových stránek z předchozího datasetu. Tyto informace jsou získávány postupným manuální procházením všech uložených stránek a podstránek. Údaje budou uspořádány do této podoby:

- **id** – jako identifikační a indexovací údaj webové stránky bude sloužit její adresa a doména ve tvaru `address_domain`;
- **company_name** – jméno podnikatele nebo firma spolu s označením druhu společnosti;

- **company_identifier** – osm ciferné identifikační číslo podnikatele nebo společnost uložené jako textový řetězec;
- **company_address** – adresa sídla společnosti, často také na webech označována jako fakturační adresa v původním formátu (jakékoliv původní oddělovače nahrazeny čárkou), z kterého budou odstraněny doplňující informace v závorkách a označení, že se jedná o adresu v České Republice, jelikož z podstaty úkolu jsou hledané pouze české adresy;
- **other_addresses** – pole ostatních adres patřící provozovnám, prodejnám a případně i skladům, pro kterých formát zápisu platí stejné podmínky, jako pro adresu společnosti;
- **opening_hours** – otevírací doba pro provozovnu, prodejnu, technickou podporu nebo zákaznickou linku uloženou ve JSON formátu, konkrétně v podobě následující standard pro ukládání otevíracích hodin `{"day": "hh:mm-hh:mm"}` [12];
- **phone_numbers** – pole českých telefonních čísel v mezinárodním formátu, pro kontakt společnosti jako celku, nikoliv jednotlivců;
- **emails** – pole emailových adres, které jsou určeny k prvnímu kontaktu s firmou, nikoliv adresy zaměstnanců;
- **facebook_url** – odkaz na profil stránky společnosti na síti Facebook uložen ve formátu, který byl na webu nalezen;
- **twitter_url** – odkaz na profil účtu společnosti na síti Twitter uložen ve formátu, který byl na webu nalezen;
- **instagram_url** – odkaz na profil účtu společnosti na síti Instagram uložen ve formátu, který byl na webu nalezen;
- **youtube_url** – odkaz na kanál společnosti na YouTube uložen ve formátu, který byl na webu nalezen;
- **linkedin_url** – odkaz na firemní profil společnosti na profesní síti Linkedin uložen ve formátu, který byl na webu nalezen.

Jelikož jsou tyto informace sbírané z vícero stránek, nastávají případy duplicit záznamů, které jsou ze své charakteristiky unikátní. Takové případy jsou řešeny způsobem zachování prvního nalezeného údaje, především na hlavní stránce. Toto pravidlo se vztahuje na všechny údaje vyjímaje adres, telefonních čísel a e-mailů. Právě tyto tři kategorie údajů umožňují ukládat vícero unikátních záznamů. Konkrétní počty vzorků jsou k nalezení v kapitole experimentů při charakteristice datového souboru v tabulce 5.1.

2.1.3 Datový soubor pro trénování klasifikátorů

Poslední připravený datový soubor obsahuje ve svých záznamech texty vybrané z HTML elementů, které jsou rozřazené do tří kategorií dle svého obsahu. Budou sloužit k trénování a evaluaci vybraných klasifikátorů a vektorizérů. První kategorie firemních informací je tvořena bloky obsahující adresu, jméno společnosti nebo IČ. Další kategorií jsou bloky textů obsahující otevírací dobu provozoven nebo dostupnosti telefonních linek. Poslední kategorii tvoří zbývající elementy s podobnou strukturou neobsahující ani jeden ze zmiňovaných údajů. Jelikož je poměr těchto textů v experimentálním datasetu nevyvážen, tento datový soubor je tvořen výběrem z těchto kategorií, tak aby každá z nich byla podpořena několika stovkami vzorků (konkrétní čísla v sekci 5.2). Tyto texty jsou získány během procházení datového souboru pro experimenty, z kterého se na základě regulárních výrazů vybíraly části HTML, které byly následně podrobeny manuální kontrole. Jsou uspořádány do této podoby (ukázka k nalezení v tabulce 2.1):

- **text** – jednotlivé textové záznamy předzpracované pomocí postupů detailně popsaných v sekci 4.1.3;
- **y** – číselné označení příslušné kategorie, kde je `0` přiřazena otevíracím hodinám, `1` informacím o společnostech a `2` textům zbývajících elementů.

text	**y**
Otevřeno Po až Pá: 8:00 - 16:00	0
Provozní doba prodejny Pondělí - Pátek: 7.30 - 18.00 hod.	0
Copyright © 2022 OG Soft s.r.o.. Všechna práva vyhrazena	1
MULTITIP Moravia s.r.o. Palackého 1136/27 741 01 Nový Jičín	1
Již 25 let se snažíme řešit balení produktů příznivě pro životní prostředí	2
Přihlašte se k odběru dárkového newsletteru Buďte první kdo se dozví o každé akci.	2
...	...

Tabulka 2.1: Ukázka záznamů z datasetu pro trénink

Kapitola 3

Získávaní dat

V této kapitole bude představen proces automatizovaného získávání dat jako celek. V úvodu bude do detailu přiblížena struktura webových stránek. Rozebrány budou nejčastěji používané HTML elementy a metadata spolu s jejich použitím v praxi. Kapitola bude dále obsahovat alternativy zpřístupnění dat pro strojové čtení a nutné teoretické základy pro ozřejmění principů, na kterých je postaveno vytěžování dat z webových stránek. Po teoretickém uvedení bude čtenář seznámen i s konkrétními nástroji, které se pro získání datových souborů používají.

3.1 Webové stránky a jejich struktura

Webová stránka je hypertextový dokument, který se uživatelům zobrazuje pomocí webového prohlížeče. Obvykle je složena z hypertextových odkazů směrujících na další weby a obsahu v podobě textu, audia, grafiky nebo jiných multimédií [13].

Obsah stránek je tvořen pomocí Hypertext Markup Language (HTML). Jedná se o značkovací jazyk, který skrytým označováním definuje strukturu a obsahovou kostru dokumentu. V minulosti tento jazyk kódoval i vzhled, ale tato odpovědnost byla převedena na tzv. kaskádové styly, aneb zkráceně CSS. Takhle zakódovaný vzhled je uživatelům bez dalšího rozšíření prezentován staticky. Existuje ale možnost vzhled měnit dynamicky v reálném čase a to pomocí jazyka JavaScript [14].

Zmiňované značky (*tagy*) slouží jako klíčová slova, které popisují jak jednotlivé položky formátovat a zobrazovat. Každý HTML tag má 3 části – otevírací část, obsah a uzavírací část (`<tag>Obsah<\tag>`) (výjimkami jsou tagy, které nemusí mít uzavírací část). Aktuálně existují desítky druhů těchto tagů a další postupně přibývají se zvedající se verzí standardizace HTML. Pro získání přehledu, lze rozřadit existující tagy do těchto skupin:

- **textové**: `<p>, <h1>...`;

- **odkazující**: `<a>, <base>...`;
- **objektové a obrazové**: `<img>, <area>...`;
- **seznamové**: `<ul>, <li>...`;
- **tabulkové**: `<table>, <tr>...`;
- **formulářové**: `<form>, <input>...`;
- **skriptovací**: `<script>, <noscript>` [15].

Vybrané tagy mají svoje pojmenovaní voleno tak, že mimo svého implicitního účelu nesou i sémantický rozměr. Díky němu lze během vytěžování získat cenné informace o jednotlivých textech, například tag `<description>` na stránkách označuje popisek dané entity. Nápomocné pro vytěžovaní mohou být i poziční tagy, například `<h1>`. Ten v případě stránky s detailem produktu v e-shopu obsahuje název daného produktu. Další možností doplňkových informací pro vytěžovací nástroje jsou tzv. meta tagy, které jsou nositelem metadat.

3.1.1 Metadata

Metadata jsou v obecné rovinně speciálním případem dat, které poskytují informace o jiných datech. Obsahují informace o vlastních datech, celkovém kontextu ale i o dalších atributech. Mohou vznikat manuálním i automatizovaným přístupem a mají za cíl lepší orientaci v dokumentu. Přispívají také k schopnosti nalézt určité informace, jelikož vystupují i jako „záložky“ k jednotlivým sekcím stránky [16].

Metadata lze do webů ukládat pomocí tagů `<meta>`. Tyto informace jsou zpravidla uložené v nevykreslující se sekci `<head>`, která ale obsahuje potřebné informace pro korektní vykreslení ostatního obsahu stránky. Typicky se zde uvádí informace o autorovi webu, popisek, název nebo kódování stránky. Tyto základní možnosti dokážou nástrojům pro automatické vytěžovaní poskytnout jen zlomek informací (v praxi slouží pro indexování ve vyhledávačích) [17].

Nositelem větší části sémantiky jsou meta-datové standardy od různých poskytovatelů, které dovolují do meta tagů ukládat mnohem obsáhlejší množinu údajů. Tyto způsoby jsou poměrně spolehlivým prostředkem pro distribuci informací k automatickému vytěžování. Jsou známé různé alternativy, které se liší svou strukturou i cílem použití. V následujících sekcích bude představena pětice z nejpoužívanějších standardů.

Mikrodata jsou používány k označování již existujícího obsahu na stránce. Vznikly díky komunitě Web Hypertext Application Technology Working Group (WHATWG) s cílem poskytnout lepší uživatelskou zkušenost při prohlížení webu. Jsou sestaveny ze skupiny párů klíč-hodnota, kde

každou takovou skupinu lze označit výrazem *item* a každý pár výrazem *property*. Pro označování jednotlivých položek je nutno používat některý z unifikovaných slovníků. Mikrodata používají slovník Schema.org , který nabízí širokou kolekci atributů pro různé druhy informací, jak možno vidět v kódu 3.1 [18].

```
<div itemscope itemtype="http://schema.org/SoftwareApplication">
  <span itemprop="name">Angry Birds</span> -

  REQUIRES <span itemprop="operatingSystem">ANDROID</span><br>
  <link itemprop="applicationCategory"
  ↪  href="http://schema.org/GameApplication"/>

  <div itemprop="offers" itemscope itemtype="http://schema.org/Offer">
    Price: <span itemprop="price">1.00</span>
    <meta itemprop="priceCurrency" content="USD" />
  </div>
</div>
```

Kód 3.1: Mikrodata [18]

Mikroformáty jsou jedním ze standardů pro popis sémantických a strukturálních dat na HTML stránkách. Pokrývají velkou škálu entit od základů až po doménově závislé informace. Používají minimalistickou syntaxi založenou na prefixech. Při vytváření objektů v mikroformátech se používá předpona `h-*` a jméno třídy. Pro přiřazení atributů danému objektu se používají prefixy jako `p-*`, `u-*`, `dt-*` a `e-*` spolu s názvem třídy. Pro označení položek používají svůj vlastní slovník microformats2. Jak konkrétně vypadá syntaxe pro jednotlivé elementy lze pozorovat v kódu 3.2 [19].

```
<p class="h-card">
  <img class="u-photo" src="https://example.org/photo.png" alt="" />
  <a class="p-name u-url" href="https://example.org">Joe Bloggs</a>
  <a class="u-email"
  ↪  href="mailto:joebloggs@example.com">joebloggs@example.com</a>,
  <span class="p-street-address">17 Austurstræti</span>
  <span class="p-locality">Reykjavík</span>
  <span class="p-country-name">Iceland</span>
</p>
```

Kód 3.2: Mikroformáty [19]

JSON-LD propojuje populární strukturovaný zápis JSON a Linked Data do jednoho formátu zápisu. Strukturou a slovníkem vychází tento protokol z mikrodat, jelikož též používá Schema.org pro označování jednotlivých elementů. Jeho výhodou je, že nezasahuje přímo do HTML kódu, jelikož jeho implementace se vyskytuje obvykle v hlavičce dokumentu a je označena pomocí tagu `<script type="application/ld+json>`. Pro představu, jak praktické jsou zápisy pomocí tohoto způsobu, je připravena ukázka v kódu 3.3 [20].

```
<script type="application/ld+json">
    {
      "@context": "https://schema.org/",
      "@type": "Recipe",
      "name": "Party Coffee Cake",
      "author": {
        "@type": "Person",
        "name": "Mary Stone"
      },
      "datePublished": "2018-03-10",
      "description": "This coffee cake is awesome and perfect for parties.",
    }
```

Kód 3.3: JSON-LD [21]

Open Graph je protokol propůjčující jakémukoliv webu vlastnosti tzv. sociálního objektu. Objekty tohoto typu používají sociální sítě k rozšíření náhledových informací pří sdílení odkazu daného webu. Jednotlivé meta tagy se označují prefixem `og:` a používají se zejména k označení názvu, typu, odkazu na náhledový obrázek a URL. Existují samozřejmě i další označení specifické pro jednotlivá multimédia (obrázek, video, audio). Jak konkrétně zapsat jednotlivé atributy pomocí Open Draph protokolu lze nalézt v kódu 3.4 [22].

```
<meta property="og:title" content="The Rock"/>
<meta property="og:type" content="video.movie"/>
<meta property="og:url" content="https://www.imdb.com/title/tt0117500/"/>
<meta property="og:image" content="https://www.imdb.com/images/rock.jpg"/>
```

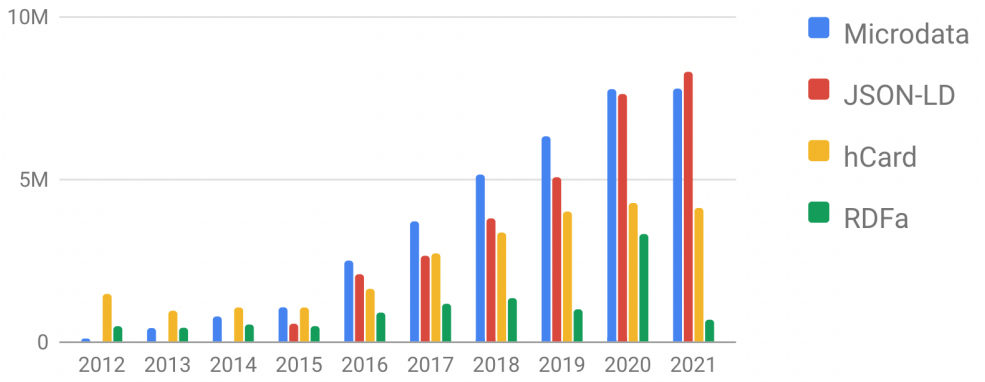
Kód 3.4: Open Graph [22]

RDFa je dalším alternativním způsobem, jak označovat metadata v HTML. Jedná se o preferovaný způsob na základě doporučení World Wide Web Consortium (W3C). K označování používá několik rezervovaných klíčových slov: `property` pro přiřazovaní specifikace domény jednotlivým elementům, `vocab` pro označení používaného slovník (umí pracovat s Schema.org), `typeof` pro přiřazení typů pro jednotlivé elementy dle slovníku a `resource` pro přidání identifikátorů. Přehled syntaxe tohoto zápisu je zachycen a v kódu 3.5 [23].

```
<p vocab="http://Schema.org/" typeof="PostalAddress"><br>
  <span property="name">Google Inc.</span><br>
  P.O. Box <span property="postOfficeBoxNumber">1234</span><br>
  <span property="addressLocality">Mountain View</span>,<br>
  <span property="addressRegion">CA</span><br>
  <span property="postalCode">94043</span><br>
  <span property="addressCountry">United States</span><br>
</p>
```

Kód 3.5: RDFa [23]

I když tyto zápisy představují pro automatizované vytěžování dat značnou míru ulehčení, jejich výskyt na webech není pravidlem. Postupně se mění jak jejich rozšíření, tak i preference ohledně výběru konkrétního zápisu na straně vývojářů stránek. Pomáhá tomu i fakt, že jednotlivé zápisy prosazují velké technologické společnosti a organizace věnující se problematice webů a distribuci dat. Jak konkrétně se mění distribuce a popularita jednotlivých protokolů lze pozorovat v grafu 3.1.



Obrázek 3.1: Distribuce a popularita zápisů metadat [24]

3.1.2 DOM

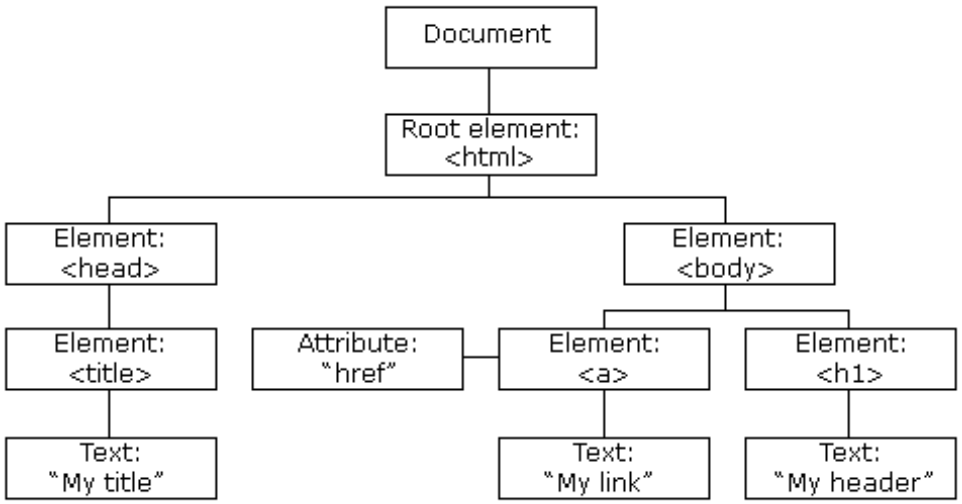
Zobrazované webové stránky zapsané v HTML musí prohlížeč nějak interně reprezentovat. Pro tento účel slouží tzv. Document Object Model (DOM) – rozhraní definující logickou strukturu dokumentů a způsob přístupu k nim. Díky této objektové reprezentaci může program modifikovat strukturu, styl nebo obsah webové stránky [25].

Jak lze z obrázku 3.2 vidět, DOM transformuje stránku do hierarchické struktury, která umožňuje uživatelům lepší orientaci v dokumentu. Tato struktura má podobu stromu (alternativně několika stromů – lesa). Programovací jazyky jako JavaScript, který se používá pro přidání funkcionalit do stránek rozumí jen této objektové reprezentaci, jejím elementům a modifikuje je pomocí vlastních funkcí [25].

DOM pracuje s různými datovými typy pro komunikaci s vnitřní reprezentaci objektů. Následující seznam představuje několik z nich ve stručném přehledu:

- **Document** – objekt tohoto typu je předkem pro všechny prvky na dané stránce;

- **Node** – základní datový typ pro každý objekt v modelu, obecně se jedná o jednotlivé uzly ve stromové struktuře;
- **Element** – více specifický typ odvozen od *Node* určen pro jednotlivé HTML prvky na stránce;
- **NodeList** – seznam jednotlivých uzlů, ke kterým je možno přistupovat pomocí indexů [26].



Obrázek 3.2: DOM struktura [27]

3.2 Automatizace získávání dat

Získávání dat lze zadefinovat jako proces sběru a měření pozorovaných informací z vybraného zdroje [28]. Jelikož zdroje dat nejsou vždy unifikované, existuje několik způsobů, jak k automatizaci sběru přistoupit. Jedním ze způsobů ulehčení přístupu k datům pro strojové zpracování je uveřejnit je jako otevřená data. To prakticky znamená neklást zbytečné technické či jiné překážky a jasně definovat podmínky jejich užití [29]. V technické rovině jde zpravidla o použití jednoho z těchto způsobů:

Webové rozhraní API – přesně definované rozhraní k online databázi nebo webové aplikaci, která data poskytuje v určeném strukturovaném formátů (například JSON, XML...). Jedná se o preferovanou cestu, jelikož tento přístup lze snadno monitorovat, kontrolovat a omezovat jen pro vybrané skupiny, a díky tomu jej i zpeněžovat. Tato cesta zpřístupnění

je většinou opatřena i podrobnou technickou a uživatelskou dokumentací spolu s podmínkami užití, které uživatelům stanovují rozsah práv a povinností při nakládání s daty [30].

Datový soubor ke stažení – stažením souborů ve formátu vhodném pro strojové čtení (typicky CSV, Excel, Parquet) uživatel získá jednu nebo více databázových tabulek, kde každý sloupec reprezentuje jistou skupinu informací a každý řádek reprezentuje záznam. Takto přístupný datový soubor ke stažení je obvykle opatřen dokumentací, která pro sloupce typicky obsahuje popis uložených informací a jejich datový typ [31].

Tyto alternativy výrazně usnadňují následnou práci s daty, jelikož jejich poskytovatelé zajišťují aktuálnost, korektnost formátu i správnost popisu. Tento druh údržby spolu s pořizovací cenou za vytvoření jednoho z těchto přístupů k datům bývají velice časově i finančně náročné. V závislosti od velikosti a rozsahu projektu se pořizovací cena může měřit až v stovkách tisíc českých korun [32].

Na první pohled se může zdát, že zpřístupnění vlastních dat nabízí konkurenci neférovou výhodu. Opak je ale pravdou, jelikož liberalizování přístupu k datům dává organizacím možnost růstu i v dosud neprobádaných odvětvích. Mezi konkrétní případy může být zařazen například přímý prodej jednotlivých datových bodů již stávajícím zákazníkům, ale také i prodej licence pro přístup ke všem datům zcela novým odběratelům. I když jsou benefity značné, je důležité pro organizace vyvažovat komplexitu správy, celkové ceny a získaných benefitů. V praxi zpřístupňují svá data vetší společnosti, které umí čerpat zmiňované benefity a mají dostatečné zdroje pro investice [33].

Pokud organizace nezpřístupní data jedním ze zmiňovaných způsobů, pořád existuje možnost, jak tyto informace vytěžovat z existujících webových stránek. Tuto činnost lze zadefinovat jako použití algoritmů a technik vytěžování dat za účelem extrakce informací přímo z webových stránek. Ze stránek lze vytěžovat obsah, dokumenty, služby, odkazy, záznamy o provozu a mnohé další druhy poznatků. Zmiňované informace spadají do tří kategorií dle cíle vytěžování:

Struktura – rozbor uzlů a propojení ve struktuře webu za pomocí grafové analýzy. Tímto způsobem lze získat přehled o propojení jednotlivých podstránek webu.

Způsoby používání – extrakce informaci z provozních záznamů serverů obvykle za účelem získání vhledů do uživatelského chování. Jsou zde obsaženy informace jako počet a místo kliknutí uživatele, čas interakce s jednotlivými elementy a další aktivity uživatele.

Obsah – extrakce obsahových informací z jednotlivých stránek, především z textu, obrázků, audio i video záznamů [34].

Další sekce teorie vytěžování bude zaměřena na extrakci obsahu, jelikož je tato kategorie provázána s celkovým cílem práce.

3.3 Vytěžování obsahu webových stránek

Obsah na webových stránkách je nabízen uživatelům ve formě textů, obrázků, videí, tabulek nebo formulářů. Proces jeho vytěžování je nejčastěji provázán s vytěžováním textu, jelikož většina informací je na webech právě v této podobě. Tento postup lze uplatnit na jednu z možných reprezentací HTML stránky:

- zdrojový HTML text,
- DOM struktura,
- vizualizace.

Samotné vytěžování je složeno z několika úkolů, které jsou uspořádané do pěti kroků:

1. **nalezení zdrojů** – uplatňují se zde techniky pro získaní relevantní informací z webu,
2. **předzpracování a selekce informací** – získané informace se reprezentují pomocí zvolené metriky, nad kterou probíhá následná selekce,
3. **generalizace** – evaluace získaných vzorů za účelem identifikace obecně platných pravidel,
4. **analýza** – vyhodnocení přesnosti získaných vzorů pomocí zvolených metrik,
5. **prezentace a vizualizace** – vhodná prezentace extrahovaného [35].

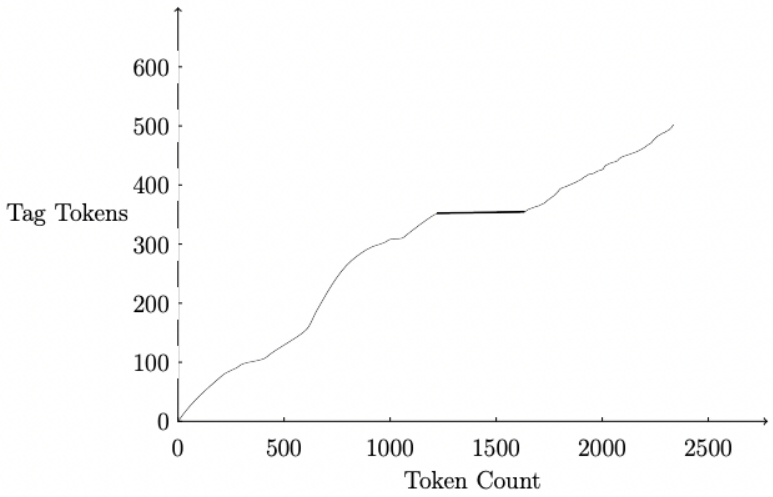
Pro účely práce budou po stránce teoretických detailů prezentovány první dva kroky spolu možnými alternativami formalizace schématu. Některé z uváděných technik a metrik budou též použity v praktické části práce během extrakce a experimentů.

3.3.1 Reprezentace HTML stránek

Díky trojici alternativních reprezentací lze ze stránek získat předvybrané oblasti zájmu, na které jsou v zapěti použity jednotlivé kroky z procesu extrakce. I když jsou v práci použity zejména přístupy, kde se web formalizuje do DOM struktury, je v tomto kroku vhodné zmínit i některé z existujících alternativ.

Zdrojový HTML text

Snahou extrakce založené na zdrojovém HTML kódu je identifikovat část s hlavním textem stránky. Tato identifikace je postavena na předpokladu, že část stránky tohoto typu sestává primárně z textu s minimálním obsahem tagů. Jedna z dostupných variant algoritmů funguje na principu kategorizace jednotlivých tokenů kódu do dvou skupin, konkrétně tagy a slova. Vznikne tak sekvence binárních označení, kterou lze reprezentovat jako distribuční křivku (3.3). Cílem algoritmu je v tomto grafu najít nejdelší ustálenou úroveň vývoje (vodorovný úsek), která reprezentuje úsek s nejmenší koncentrací tagů, neboli hlavní text. Nevýhodou tohoto postupu je neschopnost identifikovat hlavní text, pokud se v něm nachází menší textové bloky představující šum pro tento algoritmus [36].



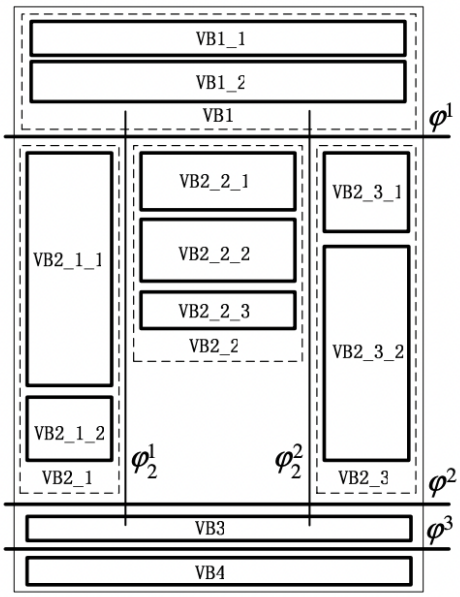
Obrázek 3.3: Křivka vyjadřující závislost mezi počtem všech tokenů a počtem tokenů označených jako tag [37]

DOM struktura

Jak již bylo vysvětleno v 3.1.2, DOM představuje vyšší úroveň abstrakce za pomocí stromové struktury nad HTML stránkou. Při přechodu sestaveným stromem algoritmus dokáže filtrovat elementy dle jejich významu. Aplikuje se tak dvojí úroveň filtrování, kde první odstraňuje elementy jako obrázky, skripty, styly nebo odkazy. Druhá úroveň se vypořádává se složitějším úkolem, kterým je odstranění reklam nebo tabulek bez informační hodnoty. Po této úpravě je možno převést z DOM reprezentace stránku opět do HTML kódu nebo textu bez tagů, na který lze dál uplatňovat techniky extrakce [37].

Vizualizace

Vizualizační přístup zvaný VISP odvozený od anglického *Vision-based Page Segmentation Algorithm* se snaží stránku reprezentovat jako by byla vykreslena v prohlížeči. Tím simuluje reálné zobrazení webu, které vnímá běžný uživatel. Algoritmus, který stránku segmentuje pomocí vizualizace operuje rekurzivně nad DOM strukturou. Postupně prohledává uzly stromu a vybírá z nich jednotlivé bloky vhodné k vizualizaci. Ukládá je do tzv. *block pool*, nad kterým probíhá další rozpad získaných bloků na menší bloky. Heuristika, která o rozpadu rozhoduje je postavena na posouzení relativní velikosti vůči ostatním přítomným prvkům v poolu, nebo například rozdílností barev pozadí jednotlivých podbloků. Blokům v poolu se dále přiřadí tzv. stupeň koherence, který vyjadřuje konzistenci obsahu. Pomocí této metriky jsou následně spojovány nebo oddělovány bloky pro dosažení požadované granularity. Jak tento algoritmus vnitřně reprezentuje stránku lze vidět na obrázku 3.4. Na základě takto rozdělené stránky se mezi bloky hledá region s hlavním obsahem, který může být podroben dalším technikám extrakce [38].



Obrázek 3.4: Reprezentace webu algoritmem vizuální segmentace [38]

3.3.2 Techniky vytěžování

Získané podoblasti zájmu jsou dále podrobeny technikám vytěžování navržených dle odpovídajících vlastností struktury stránek (dokumentů). Lze je proto rozdělit do dvou skupin dle přítomnosti struktury.

Dokumenty postrádající strukturu

Skupina dat, která nemá jasně předdefinovanou strukturu nebo model. V praxi se jedná vetšinou o text (případně data a další číselné údaje), a proto jsou tyto techniky postaveny na principu zpracování přirozeného jazyka (NLP). Jelikož postrádají jakoukoliv implicitně danou informaci o obsahu nebo formě, je náročné je automatizovaně zpracovávat. Přes to existuje několik technik, které dokážou z textu vytěžovat informace cenné pro uživatele [39].

Extrakce informací v nestrukturovaných datech funguje na základě hledání shody vzorů v textech. Po nalezení klíčových slov a frází jsou hledány jejich propojení s dalšími částmi textu. Tato technika funguje dostatečně pokud je prováděna na velkém objemu dat. Její princip tvoří základ pro další pokročilejší techniky extrakce, jelikož její předností je schopnost převádět nestrukturované informace na více strukturované data. Tento proces probíhá ve dvou krocích, kde se nejprve uskuteční vytěžení hledané informace, na které se následně uplatní soubor pravidel pro doplnění chybějící informace [40].

Sumarizace slouží ke zkrácení celkové délky psaného textu za předpokladu zachování hlavních obsahových bodů. Zbaví uživatele nutnosti procházet při čtení celý text, protože to v kratším čase za něj provede sama sumarizace. Může sloužit jako jistý „vhled“ do tématu textu bez potřeby jeho přečtení. Nejtěžší úkolem je naučit skript sémantické analýze textu, která tvoří základ pro získání interpretace elementární myšlenky. Algoritmus funguje za pomoci převažování jednotlivých vět na základě jejich důležitosti pro celkový význam. Vážení částí textu je navíc doplněno i cíleným vyhledávání nadpisů a podnadpisů pro zahrnutí klíčových bodů textu. Tato technika se zpravidla používá spolu s kategorizací a sledováním tématu, pro získání kompletního přehledu o textu [41].

Vizualizace je proces, který vykresluje extrakci vlastností a indexování do grafické podoby. Pomocí zobrazení ve formě grafiky lze v textech pozorovat a odhalit podobnosti, které by jinak nebylo možné nalézt. Sestavené hierarchie, mapy nebo grafy umožňují uživatelům procházet a analyzovat obsah zkoumaného textu [42].

Sledování tématu je technika, která sleduje uživatelskou aktivitu při prohlížení dokumentů a používá ji k profilaci daného uživatele. Pro získanou interpretace používatele je dále schopna nabídnout podobný obsah k již prohlíženým dokumentům v minulosti. Toto doporučení je často navázáno na klíčové slova, která mohou být součástí nějaké ustálené fráze. Nevýhodou tohoto přístupu je ale nemožnost detekovat homonyma, tedy slova stejně znějící, které významově odkazují na jiné jevy nebo věci. Skvělým příkladem je několikrát zmíněno slovo vytěžování, které se

v praxi často váže s nerostnými surovinami, místo odkazu na algoritmy a data [42].

Shlukování lze zadefinovat jako proces zařazování textů do předem neznámých skupin na základě jejich podobností. Lze rozlišovat různé parametry shlukování – dle hierarchie, s přítomností/absencí překryvu množin... Tento algoritmus lze považovat za jeden z nejúčinějších, v případě, že nejsou k dispozici žádné předefinované skupiny, označená data nebo jiné formy explicitně daných informací o nestrukturovaném dokumentu [43].

Dokumenty obsahující strukturu

Tento typ dat je uložen ve struktuře, která může být definována striktně i flexibilně. Nejčastěji se tento formát používá v případech, kde se na jednom místě kombinuje více datových zdrojů. Typickým příkladem nerigidního schématu je webová stránka, která díky označení jednotlivých prvků přímo v kódu dokáže dostatečně popsat svou strukturu. V obecné rovině se jedná o jakýkoliv dokument, který představuje propojení fragmentů kratších textů zapsaných a označených jedním ze značkovacích jazyků (HTML, XML, RDF...). V případě jasně dané struktury je typickým příkladem tabulka. [44]

Extrakce odshora-dolů je metodou, která postupně rozkládá nejsložitější prvky na méně složité. Rozpad je učiněn za pomocí nálezu prvku s podobnou strukturou. Tento proces je iterativní a jeho zastavení nastane při dosáhnutí úrovně atomických objektů [45].

Object Exchange Model je způsob jakým lze ukládat relevantní extrahované informace z dokumentu. Je samo-popisný, a proto nepotřebuje dodatečnou dokumentaci struktury uložení. Tento model napomáhá uživateli v porozumění extrahované informační struktury [46].

Generování obalu je technika, která shromažďuje dodatečné informace jako statistiky, odkazy a domény. Tyto prvky se souhrnně nazývají obalem. Pro vygenerování obalu lze použit jeden ze dvou známých přístupů. Prvním přístupem je indukce, která používá supervizovaný přístup učení pro extrakci pravidel. To má nevýhodu v nutnosti manuálního označování datového souboru, co má za následek vysokou časovou náročnost a těžkosti při spravě vytvořeného obalu. Druhým přístupem je automatické generování obalu, které je prováděno na základě zaužívaných šablon, které bývají na webech použity. Úspěšnost obou těchto postupů závisí od celkové informační kapacity zdroje [47].

3.3.3 Předzpracování a selekce informací

Druhým krokem procesu vytěžování informací je předzpracování a následná selekce informací. Předzpracování unifikuje formu reprezentace získaných dat.

Bez něj by nebylo možné zodpovědně určit, která data mají větší informační hodnotu. Po formalizaci dat pomocí zvolené měřitelné hodnoty – metriky následuje krok selekce informace. [35]

V následujícím seznamu budou představeny alternativy převodu získaných dat do numerických hodnot, které jsou vhodnější formou pro srovnání a selekci.

Binární předzpracování používá množinu slov ke identifikaci reprezentací, které se v daném dokumentu nacházejí. Dle přítomnosti jim pak přiřadí jednu z binárních hodnot {`0, 1`} [35] .

Term Frequency (TF) je metodika, která vyjadřuje počet daného fráze, slova nebo jiné textové jednotky v dokumentu. Frekvence dané části textu by měla odrážet jeho důležitost pro celkový význam. Jelikož dokumenty mohou mít různou délku, je potřeba tento fakt zohlednit ve výpočtu, aby se předcházelo tendenci sklouzávat k nesprávným závěrům. Tato situace může nastat v případě delších dokumentů, kde je pravděpodobnost výskytu určitého slova nebo fráze větší, než v textu kratším. Pro získaní relativizované frekvence je používán výpočet

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

kde t je hledaný výraz, d je dokument a t' označuje všechny výrazy v dokumentu [48].

Inverse Document Frequency (IDF) je metodika založena taktéž na principu frekvence výskytu výrazů, používá se zde ale jiný princip výpočtu. IDF je inverzně proporcionální k počtu dokumentů, které obsahovaly daný výraz. Výpočet má následující podobu

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

kde D jsou všechny dokumenty a N je počet dokumentů [49].

Term Frequency – Inverse Document Frequency (TF-IDF) je spojením dvou předcházejících metodik. Je odolnější vůči tendenci posuzovat jednotlivé měřené výrazy jako frekventovanější v případě delších dokumentů než-li v kratších. IDF zde vystupuje jako jistá forma normalizace, jelikož výpočet je následovný $tfidf(t,d,D) = tf(t,d) * idf(t,D)$ [50].

Textové modely poskytující embedding jsou pokročilejším přístupem převodu textu do podoby číselných vektorů. Na vstupu tohoto procesu je korpus jazyka spolu s textem k převodu a výstupem je číselná reprezentace jednotlivých textových jednotek. Existuje několik populárních alternativ jako například Word2Vec nebo Doc2Vec. Konkrétně Word2Vec

je algoritmus, který funguje ve dvou vzájemně invertibilních fázích: *Continuous Bag Of Words (CBOW)* a *Skip-Gram*. Jejich fungování je založeno na postupném procházení vět, kde se na základě slov a jejich okolí predikuje kontext, nebo na základě kontextu následující slovo. Velikost okolí je možno u tohoto postupu regulovat parametrem [51].

Po transformaci textu do numerických reprezentací přichází použití rozhodovacích postupů, kterých výsledkem je selekce vhodných informací. Ta probíhá na základě statistických modelů, nebo pomocí modelů strojového učení. Následující seznam obsahuje několik základních přístupů výběru informací.

Naivní Bayes patří do skupiny pravděpodobnostních klasifikátorů, jelikož je založen na Bayesově větě za předpokladu použití silně nezávislých prediktorů – proto má přívlastek naivní. K predikci používá pravděpodobnosti model

$$p(C_k|x) = \frac{p(C_k) * p(x|C_k)}{p(x)}$$

který je možno řetězit pro výpočet pravděpodobnosti návazných jevů. Ve svém výpočtů může používat různé druhy pravděpodobnostních modelů, v závislosti od pozorovaných jevů (gaussovský, Bernoulliho, multinomiální...). Navzdory jednoduchosti výpočtu založeném na kombinacích jednotlivých známých hodnot, přesnost predikcí překonává mnohem složitější klasifikátory [52, 53].

Rozhodovací stromy vytváří skupiny rozhodovacích pravidel ve formě stromové struktury. Každá interní větev ukládá test pro jednotlivé atributy, každá větev uchovává zase výsledek testu a každý list označení koncového atributu. Mezi hlavní výhody tohoto postupu patří intuitivnost, jelikož princip fungování je přímočarý a jednoduchý pro pochopení. Další předností je jeho vysoká informační hodnota, protože při procházení stromu od listů ke kořenu je možno zjistit, jak se strom k danému řešení dostal. Posledním výrazným plusem je jeho škálovatelnost, protože i při použití velkých datových souborů netrpí výrazným zpomalením predikční rychlosti nebo ztrátou přesnosti [54].

K-nejbližších sousedů (k-NN) je metoda stojící na předpokladu, že podobné věci stojí blízko sebe. Pro výpočet vzdálenosti lze použít různé postupy v závislosti na zkoumaných datech. Výhodou je absence nutnosti stavět model s parametry k ladění. Dalším bonusem je jeho všestrannost, jelikož se nepoužívá jen k selekci informací ale i pro regresi a klasifikaci. Na druhou stranu, není odolný vůči vetším datovým souborům, jelikož jeho výpočetní náročnost stoupá s přibývajícími daty i prediktory [55].

Support Vector Machine (SVM) je algoritmus, který má za cíl nalézt nadrovinu rozdělující datové body v N-dimenzionálním prostoru. Pokud

existuje víc možností pro umístnění nadroviny do prostoru, metoda hledá tu, která je od dat dat nejvíce vzdálená. Tímto přístupem lze do budoucna zajistit selekci dat s větší jistotou, jelikož se tak předchází mezním případům [56].

Neuronová síť je výpočetní model inspirován biologickými strukturami. Typická neuronová síť se skládá ze vstupní vrstvy, skupiny vnitřních (skrytých) vrstev a výstupní vrstvy. Každá z těchto vrstev je složena z neuronů, kterých matematický model lze zapsat jako

$$y = F(\sum_{i}^{m} x_i * w_i)$$

kde w je vektor vah, F je aktivační funkce a x je vektor vstupů z předcházející vrstvy. Konkrétní architektura a výběr aktivační funkce záleží od zkoumaného problému. Pro přenos informací napříč vrstvami je použit algoritmus zpětné a dopředné propagace [40, 57].

3.3.4 Technologie a nástroje

Pod nástrojem pro vytěžování obsahu webů lze rozumět program, který používá zmiňované techniky za účelem nalezení a sestavení datových souborů. Tyto soubory jsou sestavovány z hledaných informací, které spoluvytváří soubor nových i již extrahovaných znalostí z webů [58].

Jelikož data jsou na dnešním technologickém trhu cennou „komoditou“, jejich získávání je vysoko konkurenční prostředí. Dodavatelé jednotlivých řešení se snaží software co nejvíc uzpůsobit potřebám uživatelů. Navíc umožňují i méně technicky zdatným jednotlivcům pracovat s automatizačními nástroji díky pokročilému uživatelskému prostředí. Pro shrnutí, z hlediska uživatelské náročnosti lze nástroje pro vytěžování zařadit do těchto skupin:

Aplikace – tímto obecným pojmem jsou zastoupeny nástroje, které používají uživatelské prostředí pro nastavení vytěžování jednotlivých webů. Mají zpravidla dostupné intuitivní grafické prostředí, které umožňuje definovat oblasti a okolnosti extrakce jednotlivých položek webu. Absentují nutnost použití jakéhokoliv programovacího jazyka, jelikož jsou plně uzpůsobeny pro uživatele bez znalosti programování.

Automatizační nástroje – představují jakýsi mezistupeň mezi aplikacemi a knihovnami funkcí, jelikož propojují uživatelské prostředí a možnosti použití skriptovacího jazyka. Jejich produktem jsou roboti, kteří fungují na konceptu deterministických stavových automatů pracujících na vykonávání souboru podmínek. Jejich předností je jednoduchost, jelikož je zvládne obsloužit pokročilejší uživatel bez hlubší znalosti problematiky [59].

Knihovny – kolekce funkcí speciálně připravených pro vytěžování webů, které lze používat při sestavování vlastního programu a odkazovat se na ně přímo z kódu. Jsou psány v daném preferovaném programovacím jazyku a jsou optimalizovány pro sběr informací ze stránek. Mimo hlavních funkcí obsahují i mnoho konfiguračních schémat, typů a tříd, které ulehčuji a urychlují tvorbu vlastních skriptů. Pro jejich využití musí být uživatel obeznámen s dokumentací, a též i programovacím jazykem, v kterém je napsáno API dané knihovny [60].

Vývoj těchto nástrojů zpravidla probíhá v režii nějaké společnosti nebo open-source komunity. Jelikož jde o nákladnou činnost, využívá se několik modelů zpoplatnění nástrojů. Dostupné bez omezení bývají zpravidla (open-source) nástroje a knihovny, na kterých vývoji se podílí komunita. Podnikové a firemní aplikace bývají limitovány počtem extrakcí nebo plně zpoplatněny.

Nástroje pro vytěžování v sobě obecně kombinují dva principy: *crawling* a *scraping*. Navzdory častému zaměňování těchto pojmů, označují rozdílné činnosti, které se spolu doplňují pro kompletnost extrakce informací.

Crawling je proces věnující se objevování a hledání odkazů na webové stránky (URL) na právě procházených webech. Jde o krok, který předchází scrapingu, jelikož shromažďuje odkazy, které budou podrobeny extrakci. Nástroj, který provádí tuto činnost se nazývá crawler. Výstupem je seznam stránek a podstránek, které crawler získal dle zadaných podmínek [61].

Scraping je proces, který extrahuje z vybraných stránek hledaná data. Využívá předdefinované oblasti a elementy na konkrétních stránkách, do kterých se v iteracích dívá a stahuje z nich obsah. Ten formalizuje do datových souborů v libovolně definovaném formátu. Nástroj, který provádí tuto činnost se nazývá scraper [61].

Scrapy

Scrapy patří mezi nástroje vytěžovaní, které ve formě Python knihovny umožňují dolování dat z různých zdrojů. Z uživatelského hlediska, jde zejména o tyto případy uplatnění extrakce dat:

- **jeden zdroj** – klasické stažení dat z aktuálně prohlížené stránky do libovolně zvoleného formátu uložení;
- **kombinace zdrojů** – uživatel vykombinuje webové stránky, které podrobí extrakci [62].

Knihovna vyniká i svým technologickým pozadím a architekturou. Zatímco u alternativ je při konfiguraci potřeba doprogramovat mnoho tříd, funkcí a rozšíření, Scrapy zjednodušuje tyto povinnosti na několik funkcí a jednu konfiguraci. Vděčit za to může své *event-based* architektuře, tedy architektuře založené na posílání událostí. Ta umožňuje programátorům tvořit série operací,

které mohou extrahované data transformovat a dál ukládat do databáze bez degradace výkonu. Jinými slovy se *event-based* architektura dokáže vypořádat s latenci generovanou sítí, databází nebo procesem transformace dat během práce s mnoha otevřenými spojeními [62].

Jádrem celé architektury je pavouk (*spider*), kterého definuje uživatel. Jde o hlavní konfigurovatelný prvek, jelikož jeho životní cyklus funguje ve třech krocích: vytváří dotazy, zpracovává odpovědi a generuje výsledky zpracování. Vygenerované prvky lze dál libovolně modifikovat a transformovat dle potřeby projektu. Dále lze modifikovat procesy vytváření a ukončení pavouka, pokud je potřeba tyto průběhy provázat s dalšími činnostmi. Obrovským benefitem tohoto přístupu k scrapingu je jeho modulárnost – jednotlivé pavouky a transformace lze přepoužívat napříč projektem. Díky připravenosti Scrapy na produkční nasazení je interně také vyřešeno řízení správy cookies a cachovaní. Při naprogramování chování procesu autentizace dokáže tato knihovna řešit automaticky i tento krok [62].

Mimo propracované architektury nabízí tento nástroj i další přednosti vhodné povšimnutí. Během crawlingu si dokáže poradit s poškozenými HTML soubory, které nesplňují syntaktické pravidla. Benefitem je i přímá nativní integrace s knihovnou *Beautiful Soup* (seznámení v další kapitole) a možnost odkazovat se na elementy webu pomocí XPath a CSS selektorů. O tyto a další funkce se zasloužila rozsáhlá komunita uživatelů a přispívatelů, kteří udržují kód této knihovny dobře organizovaný a udržovaný [62].

Po pečlivém zvážení všech alternativ (výběr z dostupných alternativ probrán v sekci 3.3.4) byla tato knihovna vybrána pro použití ke získání datového souboru, nad kterým byla následně prováděna extrakce. Byly implementováni dva pavouci:

- **scraping seznamů Firmy.cz** – vstupem jsou adresy jednotlivých hlavních kategorií katalogu Firmy.cz, kde scraper postupně prochází jednotlivé stránky seznamu a ukládá adresy zobrazených firem pomocí odkazu na element obsahující URL;

- **crawling kontaktních stránek** – vstupem jsou získané webové adresy firem, které crawler prochází a dle regulérních výrazů hledá stránky obsahující kontaktní informace, obchodní podmínky, podmínky použití osobních údajů a v neposlední řadě i stránky zaměřené na představení společností.

Takto získané informace jsou ukládány lokálně v již popsaném schématu ve formátu CSV, který poskytuje vysokou kompatibilitu pro následný import do rozmanité škály nástrojů.

Alternativní nástroje

Mimo použitého Scrapy existuje několik spolehlivých komerčních i nekomerčních alternativ, pomocí kterých lze provádět crawling a scraping. Jelikož je nabídka možností rozmanitá, jednotlivé příklady nástrojů se mohou lišit cílovou skupinou, cenovými podmínkami i složitostí používání. Proto je potřeba všechny tyto aspekty při výběru vhodné technologie brát v potaz a najít rovnováhu mezi existujícími klady a zápory. Nutno zmínit, že se nejedná o vyčerpávající seznam, ale pouze o několik populárních prostředků vytěžování.

Mozenda je nástroj stejnojmenné společnosti, který využívá uživatelem vytvořené agenty k extrakci webových dat. Tito agenti dokážou rutinně pokrýt celý proces od extrakce, uložení až po distribuci na různá místa určení. Mozenda též umožňuje nasbíraná data různě transformovat a analyzovat, čím vytváří celý balíček možností práce s daty. Konfigurace scrapingu probíhá na dvou místech. Běh agentů spolu s prací na datech lze obsloužit z webové konzole. Tvorba samotných agentů probíhá ve samostatné aplikaci s názvem Agent Builder. Celá tato skupina nástrojů je zaměřena na firemní klienty, jelikož nabízejí několik úrovní cenové politiky limitující počet konkurenčních procesů a počet prohlížených stránek [63].

Automation Anywhere je nástroj využívající technologii SMART za účelem automatizace komplexních úkolů jako je například scraping. Jeho předností je možnost nahrávat pohyby kurzoru, tlačítka klávesnice a myši, z kterých následně vytvoří automatizovaný úkol. Aktuálně nabízí celou škálu nástrojů pro automatizaci jakýchkoliv rutinních úkolů. Jejich cílovým uživatelem jsou střední a větší firmy. [63, 64]

Rapid Miner je platforma spojující nástroje pro celou oblast datové vědy. Jde o pomůcky pro rychle prototypování, testování, vzdělávání a nasazování řešení do produkční praxe. Mimo zmiňovaných nadstaveb samozřejmě poskytuje i pestrou paletu nástrojů pro vytěžování dat z webů, kterých technické pozadí je postaveno na cloudové infrastruktuře [47].

ProWebScraper je nástroj pro scraping, kterého předností je ochrana proti blokování při velkém počtu dotazů na obsah webu. Je navržen pro extrakci velkého množství dat, proto je lehce škálovatelný. Dokáže pracovat jak se statickými tak dynamickými weby, co potvrzuje jeho přizpůsobitelnost k různým strukturám webů. Mezi vlastnosti tohoto nástroje patří schopnost listování v datech, možnost naplánovat pravidelné vytěžovaní ale i grafické prostředí sloužící ke konfiguraci extrakce. Cenová politika je postavena na modelu freemium, jelikož je možné si tento nástroj vyzkoušet, ale větší projekty jsou limitovány počtem vytěžených stránek [58].

WebScraper.io je doplněk prohlížeče Chrome, který slouží k scrapingu webů. Umí pracovat s navigací webu pomocí jeho mapy stránek – architektury webu v podobě XML dokumentu. Umožňuje paralelní běh jednotlivých scraperů, podporuje listování seznamů na webech a navíc obsahuje několik integrací pro jednodušší exporty dat. Pokud se uživatel rozhodne využívat ke běhu crapingu nabízený výpočetní výkon v jejich poskytovaném cloudu místo svého lokálního zařízení, bude mu účtován poplatek na základě počtu prohlížených stránek. [58].

Requests + Beautiful Soup kombinuje dvě Python knihovny, které poskytují jednoduchý, ale účinný nástroj pro scraping webových stránek. Knihovna *Requests* zajistí pomocí HTTP GET metody stažení obsahu webové stránky, který se uloží do textové podoby. Tento text se následně použije jako vstup do inicializační funkci knihovny *Beautiful Soup*, která textovou podobu HTML webu transformuje do stromové struktury. Pomocí stromové reprezentace HTML může uživatel provádět různé modifikace, čtení a hledání požadovaných elementů [65].

Apify je platforma poskytující nástroje pro scraping webových stránek. Tyto nástroje jsou určeny pokročilejším uživatelům, jelikož jejich použití se opírá o znalosti základů HTML a organizaci dat. Mimo obecného nástroje pro scraping jakékoliv webové stránky provozuje Apify i obchod doplňků, který nabízí již předpřipravené doplňky pro scraping obecně známých webů jako například Youtube, Google, Facebook... Za poplatek je možno vytěžovat informace z jednotlivých webů bez předchozí hlubší znalosti struktury a organizace daných stránek. Provoz těchto nástrojů se opírá o jejich infrastrukturu v cloudu, která navíc zajišťuje ochranu vůči blokaci vytěžování. Ta funguje díky vyššímu počtu proxy serverů, přes které jsou jednotlivé dotazy posílané [66].

Kapitola 4

Extrakce a zpracování dat

V této kapitole budou popsány konkrétní druhy extrakce a zpracování dat, které jsou založeny na třech různých principech. V jednotlivých sekcích budou přiblíženy konkrétní kroky vedoucí k extrakci a zpracování informací z metadatových protokolů. Dále budou popsána pravidla a způsoby sestavovaní jednotlivých regulárních výrazů, které byly použity ke získání základních informací o společnostech a podnikatelích. Jako alternativa k jednoduchým principům založených na pravidlech budou čitateli představeny i použité formy strojového učení. Při všech těchto způsobech budou v detailu přiblížené i knihovny a technologie aplikované během extrakce.

4.1 Použité principy

Zpracování dat je proces ukládaní a modifikace dat do použitelné a očekávané podoby. Při této činnosti se na data aplikuje skupina předpřipravených operací, které je upraví do požadované formy. Vstupem jsou extrahovaná data, které je potřeba vhodně transformovat a očistit od anomálií. Typickým příkladem je odstraňovaní nepřesných a nekompletních dat, kontrola chyb ve vstupních datech, normalizace formátů a konverze datových typů [67].

Před samotným zpracováním je potřeba získat vstupy, které budou podrobeny dalším procesům. Extrakce byla postavena na třech různých principech. První přímočarý vytěžuje informace z metadat dle použitých slovníku a protokolů. Pokročilejší způsob aplikuje připravené deterministická pravidla a regulární výrazy, díky kterým z textu webové stránky získává úseky podobající se určitým údajům dle sestavených šablon. Posledním způsobem jsou principy založené na strojovém učení, které prostřednictvím trénovacích dat získají představu o tom, jak mají jednotlivá extrahovaná data vypadat nebo kde se na stránce mají nacházet. V následujících sekcích budou tyto přístupy představeny spolu s technologiemi, které posloužily jako nástroje pro extrakci.

4.1.1 Metadata

Pro sběr informací o firmách byla použita čtveřice standardů, která na stránkách ukládá metadata. Konkrétně jde o mikroformáty, mikrodata, JSON-LD a RDFa. Tento reprezentativní vzorek byl vybrán na základě jeho četného rozšíření, a též z důvodu zajištění diverzity slovníků. Díky faktu, že mikroformáty používají vlastní slovník microformats2 a mikrodata spolu s JSON-LD a RDFa zase Schema.org, bylo možné prozkoumat obě formy zápisů v praxi. Open Graph byl z okruhu možností vyloučen, jelikož jeho syntaxe neposkytuje prostředky k zápisu atributů o firmách. Jeho hlavním cílem je totiž přidání sociálního objektu webovým stránkám, co není předmětem zkoumání této práce.

Extrakce a zpracování informací z metadat probíhá v několika krocích:

1. extrakce metadat,
2. detekce přítomných protokolů,
3. zpracování tagů dle slovníků,
4. transformace a normalizace získaných informací,
5. agregace a uložení vytěžených informací.

K extrakci metadat z HTML kódu stránek je používaná Python knihovna Extruct, s kterou se díky dobře připravenému API pracuje velmi jednoduše. Použitím jedné funkce dokáže v unifikovaném formátu vyextrahovat metadata ze všech zmiňovaných protokolů. Výstupem tohoto kroku je rozsáhlý JSON obsahující hodnoty přiřazené ke klíčům pojmenovaných podle jednotlivých vyextrahovaných protokolů. V této fázi je možné zjistit, které syntaxe byly na stránce zastoupeny a podrobit je dalšímu zjišťování. Po detekci přítomných vyextrahovaných dat přichází na řadu zpracování dat (*parsing*) dle správně zvoleného slovníku.

Využitím vhodných klíčů byla data iterativně podrobena poptávce po jednotlivých hodnotách týkajících se firemních dat. I když oba slovníky nabízejí sofistikovaný model vnořování informací dle typů hlavních objektů, je nutno konstatovat, že tyto standardy a předpisy byly ve značné míře nedodržované. Proto bylo přijato rozhodnutí ve prospěch kvantity získaných dat, které nekontroluje datový typ jednotlivých objektů. Pro účely extrakce je tedy dostatečným znakem přítomnost dané informace bez nutnosti jejího správného zařazení. V případě duplicit jednotlivých údajů se ukládají heuristicky první nalezené položky. Po úspěšném nalezení všeho přítomného se přistoupí ke transformaci a normalizaci hodnot do podob následující datový soubor pro experimenty.

Posledním krokem je agregace výsledků získaných z jednotlivých protokolů. Jelikož pro předmět práce není důležité poznat konkrétní syntaxi, ve které se metadata nacházely, je přirozeným postupem získaná data agregovat do jednoho obsáhlejšího datového souboru, který bude dál podroben experimentům.

Agregace uplatňuje dvě jednoduchá pravidla pro případy, kde je pro jednotlivé datové body přítomných vícero hodnot. Pokud jde o kategorie, které ze své podstaty připouštějí víceré hodnoty (adresy, telefonní čísla, e-maily), agregace uplatní pravidlo spojení a odstranění duplikátů. V ostatních případe je opět použita heuristika, která vybírá dřív extrahované hodnoty. Takto připravený datový soubor je následné uložen v již popsané formě (kopíruje schéma datového souboru pro experimenty 2.1.2) ve formátu CSV.

4.1.2 Regulární výrazy a pravidla

Spojením regulárních výrazů a dodatečných pravidel lze vytvořit deterministický způsob extrakce jednotlivých informací. Regulární výraz (*regex*) je definován jako sekvence znaků, který popisuje hledaný textový vzor [68]. Tento prvek má klíčovou roli v popisu hledaných informací. Většina z nich (popřípadě alespoň jejich část) má předpokládanou formu, kterou lze popsat regulárním výrazem.

Při extrakci za pomoci regulárních výrazů je použita Python knihovna Beautiful Soup, která vybírá z parsovaného HTML nebo XML dokumentu jeho strukturu elementů a organizuje ji do stromu. Ten je možno rekurzivně procházet, prohledávat i modifikovat. K parsingu lze použít jeden ze čtveřice nástrojů, které je nutno externě doinstalovat. Práce s touto knihovnou je vysoce intuitivní, protože její API je postaveno na procházení struktury stromu [65].

Na základě těchto předpokladů byly realizovány následující kroky extrakce a zpracování:

1. sestavení regulárních výrazů,
2. identifikace webových elementů, jako míst pro hledání,
3. extrakce prvků se shodou v regulárních výrazech,
4. uplatnění rozhodčích pravidel,
5. transformace a normalizace získaných informací,
6. agregace a uložení vytěžených informací.

Cílem této sekce je přiblížit principy, na kterých stojí regulární výrazy pro extrakci jednotlivých informací. Ty jsou použity spolu s dodatečnými pravidly, které představují rozhodčí prvek v atypických případech umístnění nebo formátování dat. Shrnutí těchto přístupů lze nalézt v následujícím seznamu:

Jméno společnosti – regex je postaven na předpokladu, že jméno společnosti má u sebe některou ze zkratek popisující typ společnost (s.r.o., a.s., v.o.s.…). Slabinou tohoto přístupu je fakt, že není schopen odhalit podnikatele figurující pod svým občanským jménem.

IČ – regulární výraz předpokládá výskyt osm ciferného řetězce, který může obsahovat první mezeru po třech cifrách a další mezeru po dvou cifrách. Tento formát je jeden ze často se vyskytujících zápisů identifikačního čísla podnikatele. Po vyhledání této sekvence je uplatněno dodatečné pravidlo kontroly v podobě hledání textového řetězce IČ/IČO v okolí nalezeného elementu. Okolím se rozumí text, v kterém bylo IČ nalezeno, nebo elementy, které jsou předkem/sousedem ve stromové struktuře HTML dokumentu. Tato kontrola je vykonávána z důvodu zamezení zájmen s jinými typy identifikačních čísel.

Adresy – jediná část adresy, která má ustálenou strukturu je poštovní směrovací číslo. Právě tento údaj je hledán pomocí regulárního výrazu. Po nalezení sekvence pěti cifer je uplatňován komplexní soubor podmínek pro extrakci. Obecně jde rozlišit mezi dvěma případy umístnění adres na stránce. V prvním z nich je adresa součástí souvislého bloku textu, a proto není nutno prohledávat její okolo ve smyslu předků a sousedů na úrovni stromové struktury. Druhý případem je adresa rozdělena po řádcích, kde jsou tyto řádky reprezentovány jako samostatné elementy v HTML struktuře. I když mechanika výběru je u elementů a textu odlišná, cíl je v obou případech stejný. Vezmou se všechny texty předcházející poštovnímu směrovacímu číslu (jelikož to je zvykem uvádět na konci adresy) a na ně se uplatní regulární výrazy, které hledají obecná klíčová slova jako adresa, kontakt, firma spolu s konkretními názvy společností. Tyto slova v praxi určují začátek sekvence, kterou součástí je celá adresa. Na základě nalezených klíčových slov se dál rozlišuje druh adresy na adresu společnosti a další adresy skladů, provozoven a kanceláří, které nejsou v rámci teto práce rozlišovány. Pokud se v hledaném okolí nenajde ani jeden z předpokládaných vzorů, za adresu se určí předcházející okolí PSČ. To bylo empiricky určeno na dvojici údajů v sousedních elementech nebo textu odděleném jakýmkoliv separátorem. Dalším pravidlem pro dělení adres je přistup, kdy se první nalezená adresa považuje za adresu firemní. Toto pravidlo se uplatňuje jen v případě, že nebyla nalezena adresa explicitně označena jako adresa společnosti. Tyto přístupy bohužel připouští i jistou chybovost, jelikož se do sekvence mohou dostat nepředpokládané textové výrazy, nebo může nastat chyba při uplatnění jedné ze zmíněných heuristik.

Otevírací hodiny – pro extrakci je uplatňovaná komplexní soustava regulárních výrazů a dodatečných pravidel. Prvním krokem je nalezení dvojice časů, které představují časový interval otevíracích hodin. Tyto údaje mohou mít různou formu oddělovačů i formát zapsání hodin, proto tyto skutečnosti zohledňuje regex s mnoha možnostmi. Po nalezení intervalu se v okolí hledá údaj odkazující na den v týdnu nebo jinou formu specifikace pro otevírací hodiny. Po nalezení této dvojice se proces opa-

kuje až do vyčerpání vhodného okolí prvního nálezu. Tento iterativní přistup má za cíl pokrýt co nejvíc dní v týdnu.

Telefonní čísla – snahou regulární výrazů je vyhledat sekvenci číslic, která splňuje podmínky jednoho z přijatelných zápisů telefonního čísla. Poradí si jak s mezinárodním formátem, tak i často přítomnou lokálním formou. Aby se předcházelo extrakci jakýchkoliv telefonních čísel ze stránky, jsou uplatněny podmínky, které hlídají četnost výskytů daného stylu zápisu čísla. Zadáním je najít jen obecné firemní číslo bez specifických kontaktů na zaměstnance. Ty jsou zpravidla zapsány v unifikovaných seznamech s mnoha záznamy. Tyto katalogy čísel lze na základe jejich elementů detekovat a z extrahovaných údajů vyfiltrovat.

Emailové adresy – regex je postaven na přímočaré myšlence obecně akceptovatelného formátu pro e-mailovou adresu. Tyto pravidla se zejména týkají výskytu znaku zavináče a hlídaní podoby domény, která po zavináči nasleduje. Stejně jako u telefonních čísel se zde uplatňuje pravidlo pro odstraněni nadbytečných adres, které se na stránkách mohou vyskytovat.

Odkazy na sociální sítě – regulární výrazy pro sociální sítě kontrolují přítomnost doménové adresy (nebo jejich zkrácených alternativ) v prohledávaných odkazech. Pro zamezení extrakce odkazů z tlačítek pro sdílení jednotlivých příspěvků místo odkazu na stránku samotnou, je zde kontrolována a netolerována přítomnost řetězců, které v URL označují parametry ke sdílení obsahu.

Takto sestavené regulární výrazy se uplatňuji na HTML elementy vybraných typů. Empiricky byly vybrány elementy, pomocí kterých je na web umisťován text: `<tr>`, `<div>`, `<p>`, `<span>`, `<li>`, `<h1>`, ..., `<h5>`. Pro zamezení výběru větší části textu než je potřeba, jsou elementy prohledávané s omezením na takové, které nemají ve své struktuře žádné další potomky. Jedinou výjimkou jsou tagy formátující jednotlivé části textu. Toto filtrování při prohledávání umožňuje knihovní funkce `findAll(...)`, která je součástí *Beautiful Soup*. Po uplatnění regexů a pravidel, které byly popsané v předcházející sekci, přichází na řadu transformace a normalizace formátů. Ta se využívá zejména u telefonních čísel, adres a otevíracích hodin. Všechny konečné formy pro jednotlivé datové body následují formáty popsané u datového souboru pro experimenty.

Posledním krokem je agregace výsledků hledání napříč všemi shromážděnými stránkami pro jednotlivé adresy domén. Zde nastávají podobné případy jako u agregaci metadata. Pro kategorie dat připouštějící vícero hodnot se uplatňuje stejný mechanizmus sdružování a filtrace duplikátů. Drobnou změnou prochází heuristika výběru z hodnot, které nepřipouští vícero výsledků. Ta upřednostňuje výsledky z hlavních stránek, jelikož pracuje s předpokladem, že na hlavní stránce firemního webu jsou uvedeny zpravidla obecné firemní údaje. Po agregaci probíhá uložení do formátu CSV, který následuje dříve popsané schéma.

4.1.3 Strojové učení

Použití technik vytěžování za pomoci strojového učení bude zaměřeno na data, které samy o sobě nemají ustálenou strukturu. Mezi takové případy patří jméno společnosti, adresa, IČ a otevírací doba. Právě pro tyto kategorie údajů bylo nutno vytvořit soubor regulárních výrazů a pravidel, který se opíral nejen o formát dané informace, ale navíc i o předcházející a nadcházející textové data. Zmiňované řešení není dostatečně robustní, jelikož nelze spolehlivě pokrýt všechny případy zápisů. Proto by úsilí směřující k zlepšení výsledků extrakce mělo směřovat k pokročilejší identifikaci HTML elementů, které mohou obsahovat hledané informace. Pro úkol identifikace jednotlivých elementů z hlediska jejich obsahu může vhodně posloužit klasifikátor.

Správné klasifikované elementy lze dál podrobit selekci dílčích částí textu, které reprezentují zvolené informace. I pro tento proces lze použít pokročilejší metodu než jen pravidla a regulární výrazy. Vhodným kandidátem je proces rozpoznávaní pojmenovaných entit, aneb Named Entity Recognition (NER). Tato metoda dokáže v textu lokalizovat a klasifikovat vybranou část do předem známých kategorií jako například osoby, názvy firem nebo adresy [69].

Bude tedy použita dvojice metod strojového učení, konkrétně klasifikátor a NER, a to pro různé častí procesu extrakce. Tyto metody byly zvoleny i vzhledem k dostupným datovým souborům a jejich obsahu. Jsou postaveny na principech zpracování textu, který je velice citlivý na předzpracování dat. Z tohoto důvodu bude následující sekce věnována dostupným technikám předzpracování textu i přes fakt, že některé způsoby byly okrajově zmíněné již v předcházející kapitole v rámci vytěžování dat webových stránek.

Předzpracovaní lze interpretovat jako posloupnost procesů, která vede na převod textových dat do numerické podoby. Tuto číselnou reprezentaci následně využívají modely strojového učení k tréninku a evaluaci výsledků. Součásti předzpracovaní jsou zpravidla tyto kroky:

- **tokenizace** – rozdělení souvislého textu na části, tzv. tokeny, aneb sekvence znaků svou strukturou připomínající slova, které tvoří sémantickou jednotku vhodnou pro další zpracování; během tokenizace se zahazují znaky bez sémantického významu, jako například interpunkce [70];
- **stemming a lemmatizace** – oba tyto procesy mají za cíl získat slovo v základním tvaru; během stemmingu dochází k odstraňování předpon a přípon, čím se odvozené slovo převádí na základní tvar; to je v kontrastu s lemmatizací, která invertuje proces skloňování s cílem získat základ slova prostřednictvím slovníkové a morfologické analýzy [71];
- **mazání stop slov** – odstranění slov s malou sémantickou hodnotou vzhledem k celému textu; jejich výskyt lze detekovat prostřednictvím slovníků, nebo na základě frekvence jejich výskytů v textu; v českém jazyce jsou stop slova zastoupeny zejména částicemi, spojkami a předložkami [72];

- **vektorizace** – převod textových dat do numerické podoby; konkrétní přístupy již představeny v sekci 3.3.3.

Pro shrnutí záměru lze postup seřadit do několika kroků:

1. trénování a evaluace klasifikátorů na testovacích datech,
2. výběr nejlepšího klasifikátoru a jeho použití místo regulárních výrazů pro identifikaci hledaných HTML elementů,
3. zpracování textu pomocí NER (pro nalezení firmy a adresy),
4. uplatnění rozhodčích pravidel,
5. transformace a normalizace získaných informací,
6. agregace a uložení vytěžených informací.

Klasifikace

Vzhledem k omezené dostupnosti trénovacích dat, bude klasifikátor trénován na sesbíraných vektorizovaných textových datech, které obsahují jednotlivé hledané údaje. Jde o elementy v HTML struktuře, které dále neobsahují jiné než textové potomky (jsou listem v stromu HTML). Jelikož se výskyt jména společnosti, adresy a IČ zvykne soustředit ve stejné části webu (detaily již zmíněné v sekci 2.1.3), klasifikátor bude rozpoznávat tři třídy:

- elementy obsahující firemní informace,
- elementy obsahující otevírací hodiny,
- ostatní elementy.

V rámci předzpracování byla využita řada nástrojů s podporou českého jazyka. Prvním z nich je knihovna `nltk`, která provádí tokenizaci. Následuje lemmatizace, kterou zařizuje nástroj `majka`, za použití českého morfologického korpusu [73]. Soubor pravidel pro stemming v českém jazyce je uložen a na text aplikován jako funkce v jazyce Python, pro kterou bylo pokladem implementace v jazyce Java [74]. V závěru byly vyzkoušené dva typy vektorizace z knihovny `sklearn`, konkrétně `CountVectorFeaturizer` a `TfidfVectorizer`. Jejich konkrétní nastavení bylo spolu se zvolenými klasifikátory předmětem optimalizace hyperparametrů, co bude spolu s konkrétním výběrem a popisem klasifikátorů diskutováno v sekci 5.3.3.

Takto natrénovaný klasifikátor bude použit k výběru hledaných elementů za pomoci knihovny *Beautiful Soup*, která ve své funkci `findAll(...)` připouští prostřednictvím parametru jakoukoliv vlastní filtrovací funkci. Z důvodu dosažení výsledků ve srovnatelných podmínkách se budou prohledávat elementy se stejnými vlastnostmi jako bylo popsáno v sekci pro regulární výrazy.

Z vyfiltrovaných prvků bude získaný text zbaven nadbytečných separátorů normalizován a uložen jako souvislý textový dokument. Ten lze poslat jako vstup do metody NER, nebo podrobil části již navržených pravidel a regulárních výrazů. Pro výběr adresy a jména společnosti bude dál uplatňován výběr za pomoci NER, pro zbylé údaje jako otevírací doba a IČ budou použity části pravidel, které se na text uplatňují po identifikaci.

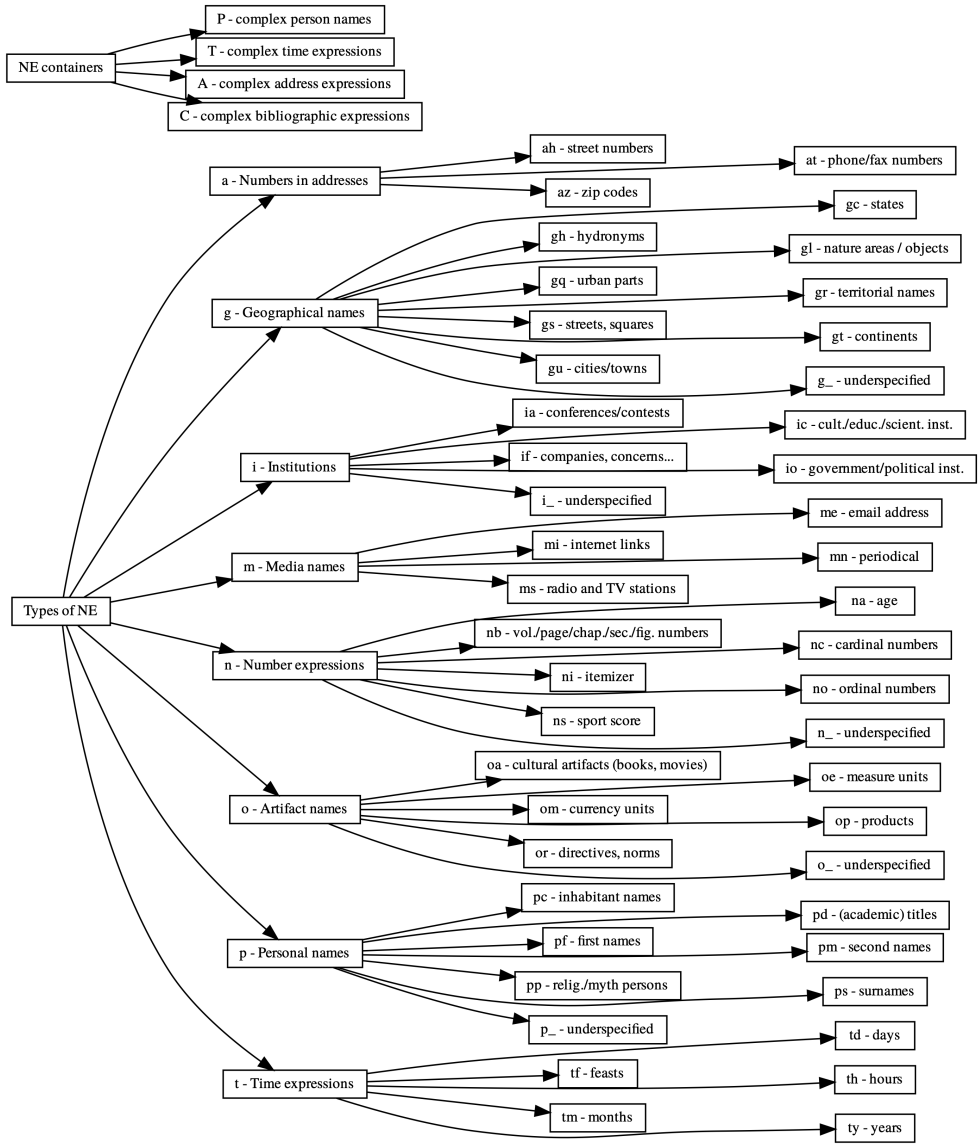
Nutno již ve fázi návrhu postupu zmínit, že absence většího objemu a diverzity hledaných informací výrazně ovlivní úspěšnost použití klasifikátoru v praxi. Pro navýšení robustnosti by bylo vhodné použít vetší počet dat a též rozlišovat vícero kategorií pro zamezení jejich prolínání. Klasifikátor by bylo možno rozšířit i o další vstupy, jako okolí zkoumaného textu, pro poskytnutí lepšího kontextu polohy elementu na stránce.

Named Entity Recognition (NER)

Pojem entita lze chápat jako slovo nebo slovní spojení, které odkazuje na známou věc či jev. Proces NER modelu je rozložen do dvou kroků: detekce části textu, která může být entitou a následné zařazení mezi definované kategorie. Vstupem pro tento proces jsou speciálně označovaná trénovací data. Jejich formát se označuje pojmem Inside-outside-beginning (IOB), který jednotlivým částem připojuje označení `I, O, B` dle jejich příslušnosti k dané entity. Od trénovacích dát dále velmi závisí úspěšnost konečné klasifikace, která přímo koreluje s relevancí dat pro dané téma. Je proto obecně prospěšné provádět trénink na textu, který je svým stylem a úpravou podobný následně testovanému textu. Pro účely práce by bylo proto vhodné natrénovat model na textech z webových stránek, sociálních sítí nebo jiných marketingových materiálů společností. [75, 76]

Kvůli absenci dostatečného množství trénovacích dat ve specifickém formátu bylo přijato rozhodnutí použit již natrénovaný model pro NER. Aktuální *state-of-art* řešení s názvem `NameTag 2` natrénováno na korpusu `Czech Named Entity Corpus 2.0 (CNEC)` je použito díky poskytovanému API přímo od tvůrců modelu. Korpus je sestaven z téměř 9000 vět v českém jazyce, které obsahují více než 35000 manuálně označených entit. Entity jsou rozděleny na osm větších skupin a celkově obsahují 46 kategorií. Je plusem, že mezi tyto kategorie patří i adresy, názvy společností a osobností, co bude klíčové pro cíl extrakce této práce. Detailní přehled entit a jejich označení lze nalézt na obrázku 4.1 [77, 78].

Model je používán jako webová služba pomocí dostupného API rozhraní. Součástí volání je text, který vznikne vyskládáním správně klasifikovaných HTML elementů. Služba vrátí tento text s nalezenými entitami formátu `XML`. Tento výstup je pomocí knihovny `Beautiful Soup` (výběr nástroje byl ovlivněn záměrem zachováním konsistence se zpracování HTML dokumentu) zprasován do podoby stromové struktury, v které se dají vyhledávat jednotlivé tagy z dokumentace 4.1. Pro nalezení adresy se ve výstupu hledá kontainer označen



Obrázek 4.1: Entity součástí korpusu CNEC [79]

tagem `A`, který je následně podroben dalšímu rozboru. V prvním kroku je z kontaineru odstraněn případný nález jména společnosti, kterého přítomnost v adrese není žádaná. Následné se validita adresy ověřuje na základě přítomnosti údajů o městě, ulici a číselných údajů (PSČ, orientační a popisné číslo). Pokud jsou tyto dílčí informace součástí kontaineru, jeho obsah je považován za relevantní adresu, která se uloží do vyextrahovaného datasetu. Obdobní proces je uplatněn při hledání názvu společnosti. Ten je pro navýšení relevantnosti

pro danou stránku hledán nejdřív v kontaineru adresy. Pokud se zde takový údaj nenachází přistoupí se k hledání jména pomocí tagu `P`, jelikož firma může být reprezentována též jménem fyzické osoby podnikatele. Takto získané údaje jsou dále agregovány za stejných podmínek jako tomu bylo v případě extrakce pomocí regulárních výrazů a uloženy do formátu CSV.

Kapitola 5

Experimenty a jejich vyhodnocení

Poslední kapitola práce se bude věnovat měření úspěšnosti extrakce, interpretaci získaných výsledků a představení vykonaných experimentů. V první části se čitatel seznámí s použitými metrikami, které sloužily k vyhodnocení jednotlivých druhů měření. Samotné měření bude rozděleno na tři sekce dle druhů extrakce. Každá ze sekcí bude obsahovat prezentaci výsledku, diskuzi závěrů a možné další kroky směřující k rozšíření a zlepšení použitých postupů. Sekce s klasifikátorem bude navíc obsahovat popis provedených experimentů a výběru vhodného druhu, který byl následně použit.

5.1 Metriky pro extrakci

Měření úspěšnosti extrakce probíhalo na manuálně anotovaném datovém souboru pro experimenty 2.1.2. Počty zastoupení vzorků v jednotlivých kategorií jsou rozepsány v tabulce 5.1.

U každé kategorie bude měřena průměrná míra shody. Výpočet tohoto údaje se liší napříč jednotlivými případy, které je možno zařadit do čtyř kategorií:

1. **absolutní shoda** – srovnání dvou textových řetězců, které nabývá hodnoty `1` v případě absence rozdílu, nebo `0` pokud se texty neshodují v celém svém rozsahu; tímto způsob jsou srovnávány údaje, které ztrácí svojí informační hodnotu již při odlišnosti jednoho znaku (IČ, jednotlivé dny otevíracích hodin, odkazy na sociální sítě);

2. **Levenshteinova vzdálenost** – tato metrika zjemňuje dopad rozdílných znaků v textech, a proto je používaná ke srovnání údajů, které v sobě nesou přidanou hodnotu i přes jistou míru odlišnosti (jméno společnosti a adresa sídla);

	Počet výskytů
Jméno společnosti	194
IČ	171
Adresa společnosti	197
Adresy	43
Otevírací doba	111
Telefonní čísla	197
E-maily	187
Facebook	123
Instagram	75
Youtube	55
Twitter	18
Linkedin	14
Celkově webů	202
Celkově stránek	593

Tabulka 5.1: Počty zastoupení jednotlivých kategorií údajů získaných ze stránek v experimentálním datovém souboru

3. **Jaccardův index** – používá se ke srovnání obsahu v množinách za účelem kontroly přebývajících údajů (telefonní čísla a emaily);
4. **Jaccardův index s využitím Levenshteinovi vzdálenosti** – speciální případ srovnání obsahu dvou množin, kde se u prvků připouští jistá míra odlišnosti nalezených textů (adresy).

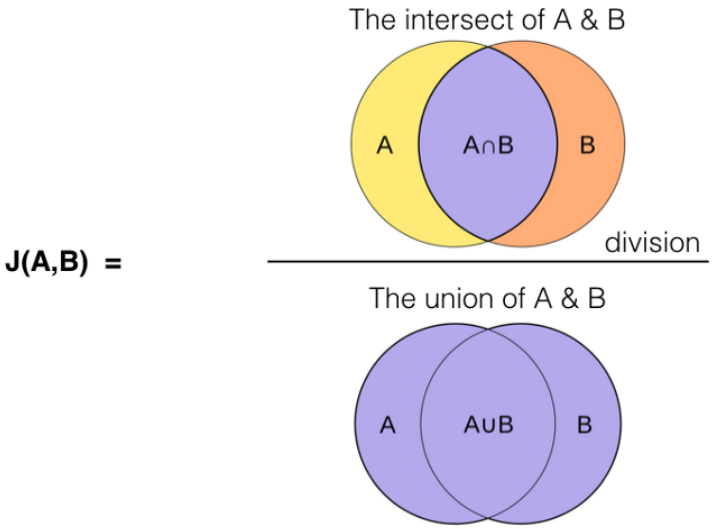
Levenshteinova vzdálenost je metrika určena ke srovnávání dvou textových řetězců, která vyjadřuje počet nutných operací vedoucích na převod jednoho textového řetězce na druhý. Mezi takové operace patří substituce, odstranění nebo přidání znaků v textu. Její přesný výpočet je uveden na obrázku 5.1. Využívá se k vyjádření podobnosti textových řetězců [80]. Pro srovnávání zmiňovaných údajů je použita skórovací funkce, kterou základ tvoří právě Levenshteinova vzdálenost. Konkrétně se jedná o metriku `token_set_ratio` srovnávající textové řetězce jako množiny tokenů z Python knihovny TheFuzz, která je normalizována na škálu `0-1`. Jelikož výpočtu předchází tokenizace, do úvahy tedy nevstupuje diakritika, velikost písma ani přítomnost speciálních znaků v jednotlivých slovech. Navíc, proces převodu posloupnosti jednotlivých tokenů do množin odstraňuje ze vstupu výpočtu pořadí a četnost jednotlivých slov.

Jaccardův index je poměr mezi průnikem a sjednocením dvou množin, který se používá k posouzení podobnosti nebo diverzity jejich vzorků. Ve své podstatě tento koeficient měří v jaké míře se překrývají dvě množiny. Pro lepší vizualizaci této metriky souží obrázek 5.2 [82].

Jaccardův index s využitím Levenshteinovi vzdálenosti je vlastní implementace Jaccardova indexu, která používá vlastní srovnávací funkci

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Obrázek 5.1: Výpočet Levenshteinovi vzdálenosti [81]



Obrázek 5.2: Vizualizace výpočtu Jaccardova indexu [83]

pro získání průniku dvou množin. Implementovaná srovnávací funkce považuje dva textové řetězce za shodné v případě, že jejich normalizované `token_set_ratio` překročí empiricky stanovenou hranici `0.7`. Hranice byla stanovena na této hodnotě díky srovnání několika různých zápisů stejné adresy, během kterého se pozorovalo jejich vzájemné skóre. Po získaní takto definované množiny průniky výpočet pokračuje dle definice Jaccardova indexu.

Ze všech napočítaných metrik se dělá aritmetický průměr napříč celou kategorií údajů. Do výpočtu nevstupují případy, kdy se v obou srovnávaných kategoriích nenachází žádný známý údaj. Na druhou stranu, případy, kdy je vyplněna jen jedna ze srovnávaných částí jsou automaticky hodnocené nulovou shodou a vstupují do celkového výpočtu.

Mimo měření shody nalezených údajů je potřeba měřit i pokrytí extrahované množiny. K těmto účelům se používají vybrané části matice záměn, která v obecné rovině slouží pro tabelární vyjádřeni míry nebo absolutního počtu správně a nesprávně klasifikovaných informací pomocí čtyř kategorií. V případě extrakce bude pomocí této metodiky hodnocena přítomnost extrahovaných dat. Mezi zmiňované kategorie matice záměn patří:

1. **skutečně pozitivní (TP)** – na webu se ve skutečnosti nachází hledaný údaj a extrakce také získala nějaký údaj,
2. **skutečně negativní (TN)** – na webu se nenachází hledaný údaj a extrakce nic nezískala,
3. **falešně pozitivní (FP)** – na webu se nenachází hledaný údaj ale extrakce něco získala (chyba prvního typu),
4. **falešně negativní (FN)** – na webu se ve skutečnosti nachází hledaný údaj, ale extrakce nezískala nic (chyba druhého typu) [84].

Z těch kategorií bude počítaná chybovost obou typů normalizovaná vzhledem k celkovému počtu přítomných vzorků.

5.2 Metriky pro experimenty

Měřeni úspěšnosti experimentu výběru vhodné kombinace vektorizéra a klasifikátora bylo prováděno na datasetu pro trénování 2.1.3. Ten byl rozdělen pomocí funkce `train_test_split` z knihovny scikit-learn. Bylo použito předurčené nastavení pro velikost testovací množiny, konkrétně 25% z celkového vyváženého počtu vzorků pro jednotlivé kategorie. Konkrétní počty pro trénovací a testovací množiny jsou uvedeny v tabulce 5.2.

	Kategorie 0 Otevírací hodiny	**Kategorie 1 Informace o společnostech**	**Kategorie 2 Ostatní**
Trénovací část	278	575	733
Testovací část	94	158	255
Celkem	372	733	988

Tabulka 5.2: Počty zastoupení jednotlivých kategorií v datovém souboru pro trénink a evaluaci

Je zřejmé, že datový soubor není vyvážený co do počtu vzorků jednotlivých kategorií. Pro vyhodnocování úspěšnosti je proto potřeba používat metriku, která tuto skutečnost zohlední. Možným přístupem je použití *vážené* varianty vybrané metriky, která napříč vícero klasifikačními třídami počítá vážený aritmetický průměr dané metriky. V případě práce to je vážená verze metriky F1-skóre, která provádí poměrně spolehlivou evaluaci i nad nevyváženími datasety.

F1-skóre je metrika používána k evaluaci binární klasifikace. Jedná se o harmonizovaný průměr hodnot *precision* a *recall*, který lze modifikovat k použití i pro měření úspěšnosti klasifikátoru s více klasifikačními třídami. Při experimentech bude použita varianta váženého průměru, který bude počítán jako součást výstupu funkce `classification_report`

z knihovny scikit-learn s parametrem *weighted*. Ten se stará o převažování výsledků na základě počtu pozitivních vzorků v jednotlivých kategoriích.

Výpočet F1-skóre je definován jako

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

kde hodnota *precision* je získaná jako poměr počtu skutečně pozitivních vzorků a součtu skutečně pozitivních vzorků s počtem vzorků falešně pozitivních ($\frac{TP}{TP+FP}$). Na druhé straně *recall* je poměrem počtu skutečně pozitivních vzorků a součtu skutečně pozitivních vzorků s počtem vzorků falešně negativních ($\frac{TP}{TP+FN}$) [85].

5.3 Měření

Měření a vyhodnocení bude probíhat na lokálních zdrojích, konkrétně se jedná o zařízení s procesorem `Apple Silicon M1`, pamětí RAM o velikosti `16 GB` a prostředí `Python 3.9.9`. Evaluace probíhá dle stanovených pravidel na odpovídajících datových souborech jak bylo popsáno v úvodních částech sekcí o metrikách.

5.3.1 Metadata

Evaluaci úspěšnosti extrakce informací z metadat bude předcházet rozbor původu získaných informací. Původem se v tomto případě myslí využitý formát zápisu. Pomocí tohoto jednoduchého měření lze získat vhled do aktuálních preferencí alternativ označování metadat v kontextu firemních stránek. Tabulka 5.3 ukazuje absolutní dominanci standardu JSON-LD, co je v souladu s aktuálními trendy z přehledového grafu 3.1. Překvapivá je úplná absence zápisu RDFa a mikroformátů, a navíc slabé zastoupení mikrodat i přes jejich obecnou popularitu.

Získané počty lze vysvětlit faktem, že společnost Google podporuje za účelem indexování JSON-LD jako preferovaný způsob [21]. I přes fakt, že se indexování opírá o přítomnost označených údajů, jejich výskyt je u měřeného vzorku spíš výjimkou než pravidlem. Měření výsledků extrakce proto nedosahuje dostačující výsledky jak ukazuje tabulka 5.4. Pro automatizovaný sběr informací o firmách z webu proto nepředstavuje spolehlivý způsob extrakce. Obecně mohou tyto údaje (pokud jsou přítomné) sloužit jako alternativní validace dat sesbíraných jiným způsobem, pokud není dostupný anotovaný datový soubor se správnými výsledky. Jelikož tyto údaje musel někdo během správy stránek ručně označit a umístit na web, lze na ně nahlížet jako na korektní za předpokladu, že se nejedná o druh dat vysoce proměnlivý v čase.

	JSON-LD	Mikroformáty	Mikrodata	RDFa
Jméno společnosti	22	0	0	0
IČ	2	0	0	0
Adresa společnosti	10	0	3	0
Adresy	0	0	0	0
Otevírací doba	3	0	0	0
Telefonní čísla	11	0	2	0
E-maily	9	0	3	0
Facebook	10	0	0	0
Instagram	8	0	0	0
Youtube	4	0	0	0
Twitter	2	0	0	0
Linkedin	2	0	0	0

Tabulka 5.3: Počty zastoupení jednotlivých kategorií údajů získaných z různých standardů pro zápis metadat

	Průměrná míra shody	Počet extrahovaných údajů	Míra falešně pozitivních údajů	Počet přítomných údajů	Míra falešně negativních údajů
Jméno společnosti	0.087245	22	0.009901	194	0.861386
IČ	0.011696	2	0.000000	171	0.836634
Adresa společnosti	0.049442	13	0.000000	197	0.910891
Adresy	0.000000	0	0.000000	43	0.212871
Otevírací doba	0.007585	3	0.009901	111	0.544554
Telefonní čísla	0.049069	13	0.000000	197	0.910891
E-maily	0.028947	12	0.014851	187	0.881188
Facebook	0.065041	10	0.000000	123	0.559406
Instagram	0.106667	8	0.000000	75	0.331683
Twitter	0.052632	2	0.004950	18	0.084158
Linkedin	0.066667	2	0.004950	14	0.064356
Youtube	0.054545	4	0.000000	55	0.252475

Tabulka 5.4: Úspěšnost extrakce s použitím metadat

5.3.2 Regulární výrazy a pravidla

Extrakce pomocí regulárních výrazů a dodatečných pravidel představuje základní přístup k automatizovanému získávaní dat. Výsledky extrakce jsou zaznamenány v tabulce 5.5.

Tento přístup dosahuje nejlepších výsledků v rozmezí kategorií, pro kterých zápis lze definovat jasná pravidla (telefonní čísla, adresy, odkazy na sociální sítě). Mírný pokles přesnosti byl zaznamenán u e-mailových adres a telefonů, co lze připsat nedostatečnému filtrování získaných údajů, jelikož dle definice smí extrakce připouštět jen obecné adresy a telefonní čísla. Během detailního prozkoumání výsledků extrakce lze ve většině případů extrahovaných dat poznat převažující případy, kdy byla z kontaktních stránek extrahována nadmnožina přípustných hodnot. Tuto skutečnost by šlo zlepšit buď heuristiku výběru z výsledkům, nebo omezením prohledávaného prostoru.

	Průměrná míra shody	Počet extrahovaných údajů	Míra falešně pozitivních údajů	Počet přítomných údajů	Míra falešně negativních údajů
Jméno společnosti	0.718402	149	0.000000	194	**0.222772**
IČ	0.680233	134	0.004950	171	**0.188119**
Adresa společnosti	0.750254	158	0.000000	197	**0.193069**
Adresy	0.457333	40	0.034653	43	0.049505
Otevírací doba	0.475855	135	**0.153465**	111	0.034653
Telefonní čísla	**0.786551**	182	0.000000	197	0.074257
E-maily	**0.702846**	154	0.000000	187	0.163366
Facebook	**0.960630**	126	0.019802	123	0.004950
Instagram	**0.986667**	74	0.000000	75	0.004950
Twitter	**0.772727**	21	0.019802	18	0.004950
Linkedin	**1.000000**	14	0.000000	14	0.000000
Youtube	**0.964286**	55	0.004950	55	0.004950

Tabulka 5.5: Úspěšnost extrakce s použitím regulárních výrazů

Druhý výraznějším nedostatkem trpí údaje, které nemají v celém svém rozsahu unikátní s ničím nezaměnitelnou strukturu. Konkrétně jde o adresu, jméno společnosti, IČ a otevírací dobu. Extrakce těchto údajů je postavena na nalezení jejich části, kterou lze popsat regulárním výrazem a následném dohledání zbývajících informací. To se podepsalo na výchylce v poměru falešně negativních a pozitivních případů. I proto bylo další směrovaní upřimné na vylepšení klasifikace oblasti, kde se tyto údaje mohou nacházet s cílem pokusit se tyto údaje lépe lokalizovat.

I přes tyto nedostatky je tento způsob díky své jednoduchosti, přímočarosti, vysvětlitelnosti a rychlé rozšiřitelnosti vhodným kandidátem pro základní extrakci údajů například pro vytvoření datového souboru k další manuální anotaci. Tým anotátorů tak může dostat poměrně spolehlivý základ, který lze po korekci použit pro trénink modelu využívajícího přístupy strojové učení.

5.3.3 Strojové učení

Při sestavování experimentu byly vybrány dva druhy vektorizérů a tři druhy klasifikátorů. Cílem bylo najít při vhodném nastavení parametrů takovou kombinaci zvolených prostředků, které dosáhnou nejvyšší hodnotu váženého průměru f1-skóre. Mimo již zmíněných dvou typů vektorizace z knihovny `sklearn`, konkrétně `CountVectorFeaturizer` a `TfidfVectorizer`, byly z této knihovny zvoleny klasifikátory `LinearSVC`, `MultinomialNB` a neposledné řadě `GradientBoostingClassifier`. Tyto konkrétní klasifikátory byly vybrány s ohledem na charakter dat (textová data) a jejich četnost (malé zastoupení jednotlivých kategorií). Mimo hodnotícího kritéria byly pro tyto postupy spočítané i další metriky jako *precision* a *recall* jako součást obsáhlejšího reportu funkce `classification_report` z knihovny `sklearn`.

Výběr hyperparametrů, které vedly k nejlepším výsledkům byl postaven na zkoušení všech možných kombinací dostupných variant pomocí funkce `GridSearchCV`. Během těchto pokusů používá k optimalizaci zvolenou *cross-*

validaci, konkrétně `StratifiedKFold`, kde byl počet rozdělení určen hodnotou `k=5`. Ta rozděluje množinu množinu dat na pětiny s ohledem na četnosti zastoupení v jednotlivých kategoriích, kde $\frac{1}{5}$ je použita jako množina validační a zbylé části jsou trénovací. Konkrétní hodnoty pro dostupné hyperparametry jsou zaznamenány v tabulce 5.6.

	Parameter	**Hodnoty**
TfidfVectorizer	analyzer	word, char, char_wb
CountVectorizer	analyzer	word, char, char_wb
LinearSVC	loss multi_class	hinge, squared_hinge ovr, crammer_singer
MultinomialNB	alpha fit_prior	0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0 True, False
GradientBoostingClassifier	learning_rate n_estimators	0.001, 0.01, 0.05, 0.1 50, 100, 200, 500, 1000

Tabulka 5.6: Použité hodnoty pro zvolené hyperparametry k ladění

Po doběhnutí všech měření byly vyhodnoceny výsledky dle dříve stanovených kritérií. Jako nejlepší byla dle výsledků zvolena kombinace vektorizace a klasifikátoru `TfidfVectorizer` + `GradientBoostingClassifier`, jak lze vidět v tabulce 5.7. Takto natrénované modely byly uloženy do formátu `pickle`, který byl následně uložen do extrakční knihovny pro použití při klasifikaci jednotlivých HTML elementů. Dle vlastností připraveného datasetu lze konstatovat, že klasifikátor je i přes rozdělení trénovací a testovací množiny pravděpodobně přeučen, co naznačuje poměrně vysoký výsledek měřeného f1-skóre. Z důvodu omezeného počtu vzorků v datasetu nebylo možné dosáhnout dostatečnou diverzitu, která by zabránila přeučení. To se s velkou pravděpodobností projeví i na výkonu klasifikátoru v praxi při identifikaci neznámých textů z typů webových stránek, které dataset nepokrývá. Dalším možným zlepšením je použít pokročilejší metodu vektorizace, s využitím korpusu českého jazyka. Tímto krokem by byla také snížena závislost od dodaného počtu a druhu trénovacích vzorků.

Po zvolení klasifikátoru byla měřena jeho úspěšnost při extrakci spolu s využitím NER pro výběr jména společnosti a adresy, eventuálně pravidel a regulárních výrazů pro selekci IČ a otevírací doby. Výsledky v tabulce 5.8, zejména míry falešné pozitivních a negativních údajů vypovídají o navýšení schopnosti nalézt oblast, ve které je dál potřeba hledat jednotlivé informace. Pokles míry falešně negativních údajú v případě IČ a jména společnosti souvisí s navýšením absolutním počtem extrahovaných údajů. Naopak sníženi počtu, a tedy upřesnění extrahovaných částí pro otevírací dobu souvísí s poklesem falešné pozitivních vzorků. U otevíracích dob si lze všimnout i mírné zlepšení průměrné míry shody, na druhou stranu všechny ostatní kategorie údajů zaznamenaly pokles. Nejmarkantnější zhoršení lze registrovat u adresy společnosti, která se propadla jak v míře shody, tak v pokrytí extrahovaných záznamů. Vysvětlení

Vektorizér	Klasifikátor	Hyperparametry	Precision	Recall	F1-skóre
CountVectorizer	LinearSVC	analyzer: char multi_class: ovr loss: hinge	0.9496	0.9494	0.9495
TfidfVectorizer	LinearSVC	analyzer: char multi_class: crammer_singer loss: hinge	0.9536	0.9513	0.9518
CountVectorizer	MultinomialNB	analyzer: word fit_prior: True alpha: 0.05	0.8878	0.8708	0.8713
TfidfVectorizer	MultinomialNB	analyzer: char fit_prior: False alpha: 0.001	0.8781	0.8783	0.8769
CountVectorizer	GradientBoostingClassifier	analyzer: char_wb learning_rate: 0.1 n_estimators': 1000	0.9616	0.9607	0.9608
TfidfVectorizer	**GradientBoostingClassifier**	**analyzer: char_wb learning_rate: 0.1 n_estimators': 1000**	**0.9636**	**0.9625**	**0.9628**

Tabulka 5.7: Výsledky experimentu hledání vhodného nastavení a kombinace vektorizéru a klasifikátoru

těchto jevů možno rozdělit do dvou částí dle použité techniky. U NERu lze nepřesnost extrakce připsat absenci kontroly nad trénovacími daty pro model, jelikož byl NER použit pouze jako služba pomocí API. K zlepšení by bylo možné směrovat při použití vlastních trénovacích dat, které vycházejí ze stylu komunikace použitém na firemních webových stránkách. Pokles přesnosti u IČ je způsoben zejména navýšením počtu extrahovaných údajů, kde se do výsledků dostaly identifikátory nacházející se na stránkách nepatřící dané společnosti.

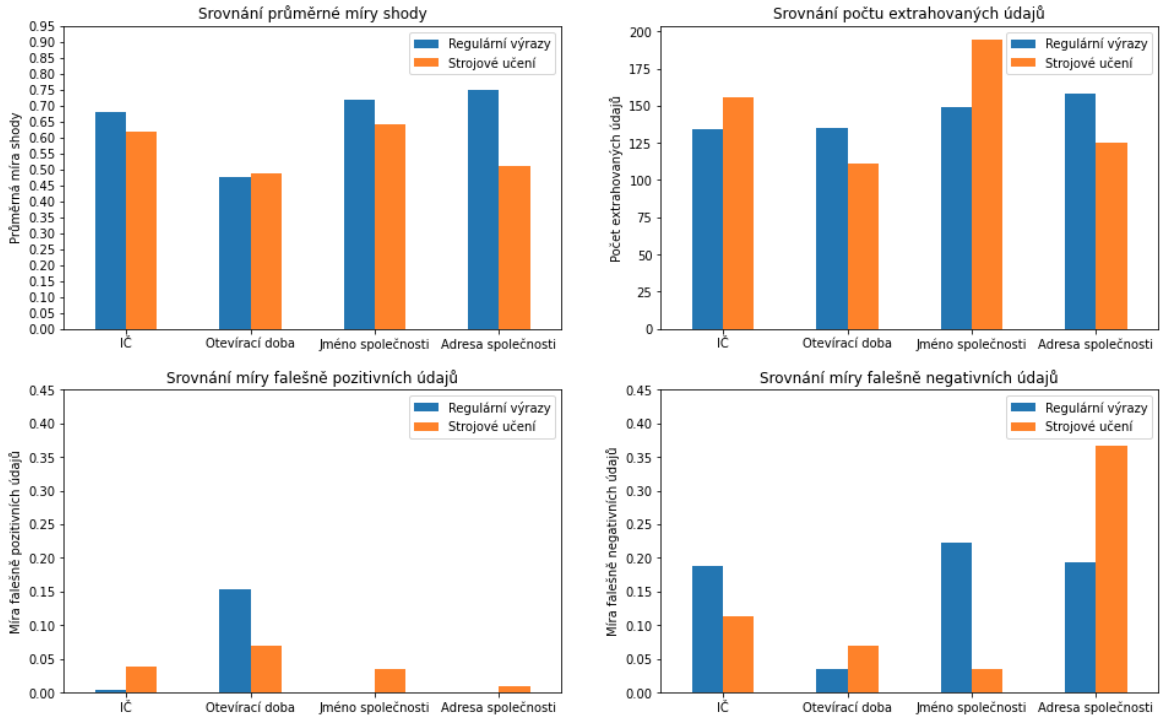
	Průměrná míra shody	Počet extrahovaných údajů	Míra falešně pozitivních údajů	Počet přítomných údajů	Míra falešně negativních údajů
Jméno společnosti	0.640896	194	**0.034653**	194	**0.034653**
IČ	0.620112	156	**0.039604**	171	**0.113861**
Adresa společnosti	0.511307	125	**0.009901**	197	**0.366337**
Otevírací doba	0.485714	111	**0.069307**	111	**0.069307**

Tabulka 5.8: Úspěšnost extrakce s použitím strojového učení

Pro lepší přehled ve změnách výsledků jsou měřené ukazovatele vizualizovány v grafech na obrázku 5.3.

5.3.4 Další rozvoj

Celý proces sestavování experimentů a měření narážel na limit dostupnosti vhodných dat. Jelikož neexistuje volně přístupný dataset, který by splňoval požadavky na využitelnost pro tuto práci, bylo nutností přistoupit k vytvoření vlastního datového souboru. Tento proces byl s ohledem na absenci lidských zdrojů a vhodných nástrojů pro manuální anotaci vysoce časově náročný. Podepsalo se to na celkovém počtu vzorků a tím i na výsledcích měření. Pro další rozvoj tohoto tématu je proto nutností získat obsáhlejší a širší datové soubory,



Obrázek 5.3: Srovnání úspěšnosti vybraných způsobů extrakce

které budou vhodné k použití pro trénování modelů v oblasti strojového učení. Za účelem použití metod odlišných od těch, které se omezují jen na textové informace, by bylo prospěšné získat obrazový datový soubor s anotovanými sekcemi na webu pro pokročilejší vizuální zpracování pomocí neuronových sítí. Vizuální stránka jednotlivých HTML elementů by mohla dopomoct k zvýšení přesnosti klasifikace jednotlivých sekcí. Dalším vhodným adeptem pro možný rozvoj v oblasti extrakce informací při absenci velkého množství vhodných dat je *transfer learning*. Jedná se o celou podoblast strojového učení, která se zaobírá přizpůsobováním již existujících modelů pro nové specifické úkoly. Šlo by tak použít již existující modely a datasety, které nebyly vytvořené primárně pro extrakci dat z webu.

Pro ulehčení práce při prototypování a měření výkonu alternativních způsobů extrakce dokáže pomoct vzniklá knihovna `cw_information_extractor` přístupná na webu https://github.com/tomasis98/cw-information-extractor. Její architektura je uzpůsobena další rozšiřitelnosti a unifikované práci s daty. Je také připravena na měření úspěšností extrakce získaných datových souborů. Plusem je též dostupnost vícero možností pro její použití, například pomocí konzole nebo pomocí funkcí v prostředí *Jupyter Notebook*. Více informací o konkretním rozhraní knihovny je uvedeno v přiložené dokumentaci A.

Závěr

Cílem práce bylo automaticky zpracovávat informace o firmách z jejich webových stránek. Součástí tohoto procesu bylo získat vhled do problematiky charakteristiky firem a tyto poznatky zapracovat během koncepce postupů extrakce. Prvním krokem proto bylo sestavení datových souborů pro extrakci informací, evaluaci výsledků i trénování modelu pro vyzkoušení přístupu založeném na strojovém učení. Dalším krokem bylo aplikovat základní i pokročilejší postupy extrakce na stránky uložené v získaném datasetu. Během tohoto procesu vznikla extrakční knihovna v jazyce *Python*, která byla použita i pro evaluaci výsledků. Součástí této knihovny je i klasifikátor používaný k identifikaci HTML elementů, kterého výběr byl řízen experimentálním měření výsledků na vlastním datasetu.

Lze konstatovat, že se podařilo splnit všechny stanovené cíle. Teoretická stránka práce plní přehledový účel a tím poskytuje nutné znalosti pro vstup do problematicky extrakce informací z firemních webových stránek. Naplňování dílčích cílů bylo ale doprovázeno hendikepem v podobě absence možnosti získat obsáhlejší datové soubory, co v jisté míře ovlivnilo kvalitu experimentů. Výsledkem je proto práce, která představuje přehled možných přístupů k extrakci na omezenější množině stránek. Na druhou stranu, byl během implementace jednotlivých způsobů extrakce v rámci knihovny kladen důraz na její další rozvoj a rozšířitelnost o alternativní přístupy. Zde vidím příležitost pro prozkoumání pokročilejších technik extrakce, jelikož implementací knihovny vzniklo prostředí, ve kterém je možno rychlé prototypování a snadná evaluace výsledků.

Literatura

[1] Kočí, P.: Co je to obchodní firma? 2014. Dostupné z: https://www.mka-nosko.cz/posts/co-je-to-obchodni-firma/

[2] Chválová, J.: Co je Obchodní společnost. Dostupné z: https://www.penize.cz/slovnik/obchodni-spolecnost

[3] Podnikatel aneb vše, co potřebujete vědět o tomto pojmu – přehledně na jednom místě! 2018. Dostupné z: https://comeflexoffice.cz/podnikatel-aneb-vse-co-potrebujete-vedet-na-jednom-miste/

[4] IČO, nebo IČ? 2021. Dostupné z: https://www.idoklad.cz/blog/ico-nebo-ic-co-to-vlastne-je-a-jak-ho-ziskat

[5] Kdy si vystačíte s místem podnikání a kdy potřebujete provozovnu? 2019. Dostupné z: https://www.pruvodcepodnikanim.cz/clanek/kdy-si-vystacite-s-mistem-podnikani-a-kdy-potrebujete-provozovnu/

[6] Dublino, J.: 12 Tips for Building an Effective Business Website. 2021. Dostupné z: https://www.businessnewsdaily.com/9811-effective-business-website-tips.html

[7] 10. díl: Co vše by měl splňovat firemní web? 2020. Dostupné z: https://www.pruvodcepodnikanim.cz/clanek/co-vse-by-mel-splnovat-firemni-web/

[8] ABSTORE. 2022. Dostupné z: https://www.abstore.cz/

[9] ACI - Auto Components International. 2022. Dostupné z: https://aci.cz/

[10] UNI HOBBY, a.s. 2022. Dostupné z: https://unihobby.cz/kontakty.html

[11] Firmy.cz. 2022. Dostupné z: https://www.firmy.cz/

[12] Key:opening_hours. 2022. Dostupné z: https://wiki.openstreetmap.org/wiki/Key:opening_hours

[13] Web page. Dostupné z: https://www.thefreedictionary.com/web+page

[14] Ubah, K.: Learn Web Development Basics. Dostupné z: https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners/

[15] HTML Tags. 2011. Dostupné z: https://www.javatpoint.com/html-tags

[16] What is metadata and why is it as important as the data itself? 2021. Dostupné z: https://www.opendatasoft.com/en/blog/what-is-metadata-and-why-is-it-important-data/

[17] What's in the head? Metadata in HTML. 1998. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML

[18] Microdata. 1998. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTML/Microdata

[19] Microformats. 1998. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTML/microformats

[20] Sanders, A.: A Guide to JSON-LD for Beginners. 2022. Dostupné z: https://moz.com/blog/json-ld-for-beginners

[21] Understand how structured data works. 2022. Dostupné z: https://developers.google.com/search/docs/advanced/structured-data/intro-structured-data

[22] The Open Graph protocol. Dostupné z: https://ogp.me/

[23] Tag your website with RDFa according to Schema.org's guidelines. 2022. Dostupné z: https://www.ionos.com/digitalguide/websites/website-creation/tutorial-rdfa-markup-with-schemaorg/

[24] Bizer, C.; Meusel, R.; Primpeli, A.; aj.: Web Data Commons - Microdata, RDFa, JSON-LD, and Microformat Data Sets. 2014. Dostupné z: http://webdatacommons.org/structureddata/

[25] Abhishek, R.: DOM (Document Object Model). Dostupné z: https://www.geeksforgeeks.org/dom-document-object-model/

[26] Introduction to the DOM. 1998. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

[27] The HTML DOM Tree of Objects. Dostupné z: https://www.w3schools.com/js/pic_htmltree.gif

[28] EM, C.: Data Collection: The complete Guide. 2020. Dostupné z: https://www.easyearnedmoney.com/data-collection/

[29] Co jsou otevřená data. 2015. Dostupné z: https://web.archive.org/web/20150626164553/http://www.otevrenadata.cz/otevrena-data/co-jsou-otevrena-data/

[30] Getting Data from the Web. 2021. Dostupné z: https://datajournalism.com/read/handbook/one/getting-data/getting-data-from-the-web

[31] Snijders, C.; Matzat, U.; Reips, U.-D.: "Big Data": Big Gaps of Knowledge in the Field of Internet Science. *International Journal of Internet Science*, ročník 2012, č. 7, 2012: s. 1–5, ISSN 1662-5544, doi:1662-5544.

[32] Mehta, A.: A Complete Guide to API Development – Importance, Working, Tools, Terminology and best Practices. 2022. Dostupné z: https://appinventiv.com/blog/complete-guide-to-api-development/

[33] Hughes, K.: API Development. 2021. Dostupné z: https://www.karllhughes.com/posts/api-development

[34] Web Mining. Dostupné z: https://www.techopedia.com/definition/15634/web-mining

[35] Bharanipriya; Prasad, K.: WEB CONTENT MINING TOOLS. *International Journal of Information Technology and Knowledge Management*, ročník 4, č. 1, 2011: s. 211–215. Dostupné z: http://www.csjournals.com/IJITKM/PDF%204-1/43.V.%20Bharanipriya1%20&%20V.%20Kamakshi%20Prasad2.pdf

[36] Finn, A.; Kushmerick, N.; Smyth, B.: Fact or Fiction: Content Classification for Digital Libraries. In *DELOS*, 2001.

[37] Content Extraction from Webpages Using Machine Learning. 2017. Dostupné z: https://webis.de/downloads/theses/papers/yunis_2017.pdf

[38] Cai, D.; Yu, S.; Wen, J.-R.; aj.: VIPS: a Vision-based Page Segmentation Algorithm. Technická Zpráva MSR-TR-2003-79, November 2003. Dostupné z: https://www.microsoft.com/en-us/research/publication/vips-a-vision-based-page-segmentation-algorithm/

[39] Unstructured Data Mining. 2022. Dostupné z: https://www.techopedia.com/definition/30576/unstructured-data-mining

[40] Johnson, F.; Gupta, S. K.: Web Content Mining Techniques. *International Journal of Computer Applications*, ročník 47, č. 11, 2012: s. 44–50, ISSN 0975 - 8887. Dostupné z: https://research.ijcaonline.org/volume47/number11/pxc3880266.pdf

[41] Fan, W.; Wallace, L.; Rich, S.; aj.: Tapping the power of text mining. *Communications of the ACM*, ročník 49, č. 9, 2006: s. 76–82, ISSN 0001-0782, doi:10.1145/1151030.1151032. Dostupné z: https://dl.acm.org/doi/10.1145/1151030.1151032

[42] Gupta, V.; Lehal, G. S.: A Survey of Text Mining Techniques and Applications. *Journal of Emerging Technologies in Web Intelligence*, ročník 1, č. 1, 2009-08-01: s. 60–76, ISSN 1798-0461, doi:10.4304/jetwi.1.1.60-76. Dostupné z: http://ojs.academypublisher.com/index.php/jetwi/article/view/11

[43] Prasad, D.; Madhusudanan, S.; Jaganathan, S.: UCLUST. *ARPN Journal of Engineering and Applied Sciences*, ročník 10, č. 5, 2015: s. 2108–2117, ISSN 1819-6608. Dostupné z: https://www.researchgate.net/publication/282739560_uCLUST-a_new_algorithm_for_clustering_unstructured_data

[44] Madani, A.; Boussaid, O.; Zegour, D. E.: Semi-structured Documents Mining. *Procedia Computer Science*, ročník 22, 2013: s. 330–339, ISSN 18770509, doi:10.1016/j.procs.2013.09.110. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S1877050913009034

[45] Ribeiro-Neto, B.; Laender, A.; da Silva, A.: Top-down extraction of semi-structured data. *6th International Symposium on String Processing and Information Retrieval. 5th International Workshop on Groupware (Cat. No.PR00268)*, 1999: s. 176–183, doi:10.1109/SPIRE.1999.796593. Dostupné z: http://ieeexplore.ieee.org/document/796593/

[46] Pol, K.; Patil, N.; Patankar, S.; aj.: A Survey on Web Content Mining and Extraction of Structured and Semistructured Data. *2008 First International Conference on Emerging Trends in Engineering and Technology*, 2008: s. 543–546, doi:10.1109/ICETET.2008.251. Dostupné z: http://ieeexplore.ieee.org/document/4579960/

[47] Pujar, M.; Mundada, M. R.: A Systematic Review Web Content Mining Tools and its Applications. *International Journal of Advanced Computer Science and Applications*, ročník 12, č. 8, 2021: s. 752–759, ISSN 1662-5544. Dostupné z: https://thesai.org/Downloads/Volume12No8/Paper_86-A_Systematic_Review_Web_Content_Mining_Tools.pdf

[48] Ganesan, K.: What is Term Frequency? 2019. Dostupné z: https://www.opinosis-analytics.com/knowledge-base/term-frequency-explained/#.Yj3aeprMLDT

[49] Papineni, K.: Why inverse document frequency? *Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001 - NAACL '01*, 2001: s. 1–8, doi:10.3115/1073336.1073340. Dostupné z: http://portal.acm.org/citation.cfm?doid=1073336.1073340

[50] Christian, H.; Agus, M. P.; Suhartono, D.: Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, ročník 7, č. 4, 2016-12-31: s. 285–294, ISSN 2476-907X, doi: 10.21512/comtech.v7i4.3746. Dostupné z: https://journal.binus.ac.id/index.php/comtech/article/view/3746

[51] Ali, Z.: A simple Word2vec tutorial. Dostupné z: https://medium.com/@zafaralibagh6/a-simple-word2vec-tutorial-61e64e38a6a1

[52] Piryonesi, S. M.; El-Diraby, T. E.: Role of Data Analytics in Infrastructure Asset Management. *Journal of Transportation Engineering, Part B: Pavements*, ročník 146, č. 2, 2020, ISSN 2573-5438, doi: 10.1061/JPEODX.0000175. Dostupné z: http://ascelibrary.org/doi/10.1061/JPEODX.0000175

[53] Mughal, M. J. H.: International Journal of Advanced Computer Science and Applications. *International Journal of Advanced Computer Science and Applications*, ročník 9, č. 6, 2018: s. 208–215, ISSN 1662-5544. Dostupné z: https://thesai.org/Downloads/Volume9No6/Paper_30-Data_Mining_Web_Data_Mining_Techniques.pdf

[54] The Ultimate Guide to Decision Trees for Machine Learning. 2020. Dostupné z: https://www.keboola.com/blog/decision-trees-machine-learning

[55] Harrison, O.: Machine Learning Basics with the K-Nearest Neighbors Algorithm. 2018. Dostupné z: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[56] Gandhi, R.: Support Vector Machine. 2018. Dostupné z: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[57] Fumo, D.: A Gentle Introduction To Neural Networks Series — Part 1. 2017. Dostupné z: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc

[58] 10 Most Popular Web Mining Tools and Softwares Compared. Dostupné z: https://prowebscraper.com/blog/web-mining-tools/

[59] Casey, K.: How to explain Robotic Process Automation (RPA) in plain English. 2020. Dostupné z: https://enterprisersproject.com/article/2019/5/rpa-robotic-process-automation-how-explain

[60] Library. 2021. Dostupné z: https://www.computerhope.com/jargon/l/library.htm

[61] Colm, K.: Web crawling vs web scraping. Dostupné z: https://www.zyte.com/learn/difference-between-web-scraping-and-web-crawling/

[62] Kocman, T.: Představení knihovny Scrapy pro tvorbu web crawlerů. 2019. Dostupné z: https://www.root.cz/clanky/predstaveni-knihovny-scrapy-pro-tvorbu-web-crawleru/

[63] Herrouz, A.; Khentout, C.; Djoudi, M.: Overview of Web Content Mining Tools. *CoRR*, ročník abs/1307.1024, 2013, 1307.1024. Dostupné z: http://arxiv.org/abs/1307.1024

[64] Automation Anywhere. 2022. Dostupné z: https://www.automationanywhere.com/

[65] Richardson, L.: Beautiful Soup Documentation¶. 2004. Dostupné z: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[66] Barton, D.: How to scrape any website (for beginners). 2022. Dostupné z: https://blog.apify.com/how-to-scrape-any-website-for-beginners/

[67] Duggal, N.: What Is Data Processing: Cycle, Types, Methods, Steps and Examples. 2009. Dostupné z: https://www.simplilearn.com/what-is-data-processing-article

[68] Chong, J.: Regular Expressions Clearly Explained with Examples. Dostupné z: https://towardsdatascience.com/regular-expressions-clearly-explained-with-examples-822d76b037b4

[69] Li, S.: Named Entity Recognition with NLTK and SpaCy. Dostupné z: https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da

[70] Chakravarthy, S.: Tokenization for Natural Language Processing. Dostupné z: https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4

[71] Beri, A.: Stemming vs Lemmatization. Dostupné z: https://towardsdatascience.com/stemming-vs-lemmatization-2daddabcb221

[72] Janík, M.: Stop slova. Dostupné z: https://www.michaljanik.cz/oblibene/stop-slova

[73] Šmerk, P.; Horák, A.: Fast Morphological Analysis of Czech. *RASLAN 2009*, 2009: s. 13–16, ISSN 978-80-210-5048-8. Dostupné z: https://nlp.fi.muni.cz/raslan/2009/papers/13.pdf

[74] Savoy, J.: IR Multilingual Resources at UniNE. 2005. Dostupné z: http://members.unine.ch/jacques.savoy/clef/

[75] Marshall, C.: What is named entity recognition (NER) and how can I use it? Dostupné z: https://medium.com/mysuperai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d

[76] Gupta, M.: NLP — IOB tags. Dostupné z: https://www.geeksforgeeks.org/nlp-iob-tags/

[77] Straková, J.; Straka, M.; Hajič, J.: Neural Architectures for Nested NER through Linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, ISBN 978-1-950737-48-2, s. 5326–5331.

[78] Ševčíková, M.; Žabokrtský, Z.; Krůza, O.: Named Entities in Czech: Annotating Data and Developing NE Tagger. In *Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue*, *Lecture Notes in Computer Science*, ročník 4629, editace V. Matoušek; P. Mautner, Berlin / Heidelberg: Springer, 2007, ISBN 978-3-540-74627-0, ISSN 0302-9743, s. 188–195.

[79] Ševčíková, M.; Žabokrtský, Z.; Straková, J.; aj.: NE type hierrarchy. 2022. Dostupné z: https://ufal.mff.cuni.cz/cnec/cnec2.0

[80] Venditama, D.: Levenshtein Distance for Dummies. Dostupné z: https://medium.com/analytics-vidhya/levenshtein-distance-for-dummies-dd9eb83d3e09

[81] Venditama, D.: Levenshtein Distance Equations. Dostupné z: https://miro.medium.com/max/1400/1*F8qZjU7QtFePNMHclpeB4w.png

[82] Understand Jaccard Index, Jaccard Similarity in Minutes. 2017. Dostupné z: https://medium.com/data-science-bootcamp/understand-jaccard-index-jaccard-similarity-in-minutes-25a703fbf9d7

[83] Visualize the Jaccard Similarity. 2017. Dostupné z: https://miro.medium.com/max/700/1*XiLRKr_Bo-VdgqVI-SvSQg.png

[84] Suresh, A.: What is a confusion matrix? Dostupné z: https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

[85] Wood, T.: What is the F-score? Dostupné z: https://deepai.org/machine-learning-glossary-and-terms/f-score

Příloha A

Dokumentace knihovny `cw_information_extractor`

Tato knihovna umožňuje extrakci informací o firmách z jejich webových stránek pomocí tří dostupných alternativ. Podporovány jsou extrakce s využitím:

1. metadat,
2. regulárních výrazů a pravidel,
3. kombinace klasifikátoru a NER/regulárních výrazů.

První dva způsoby představují základ, který dokáže ze stránek extrahovat tyto údaje:

- **jméno společnosti**,
- **IČ**,
- **adresa společnosti**,
- ostatní adresy (provozovny, prodejny, sklady...),
- **otevírací doba**,
- telefonní čísla
- e-maily
- odkazy na soc. sítě (Facebook, Instagram, Youtube, Twitter, Linkedin).

Přístup využívající strojové učení extrahují pouze hrubě vyznačenou podmnožinu, jelikož tyto případy nejsou dostatečně pokryty regulárními výrazy.

A.1 Instalace

Knihovna používá jazyk `Python 3.9+` a balíčkovací prostředí `pip 21.3.1`. Před její instalací je doporučeno vytvořit na svém lokálním zařízení nové virtuální prostředí. Po vytvoření a aktivaci tohoto prostředí vykonejte tyto kroky:

```
git clone git@github.com:tomasis98/cw-information-extractor.git
cd cw-information-extractor
pip install .
```

A.2 Používání

Po úspěšné instalaci by do prostředí měl přibýt balíček `cw-information-extractor`, který lze spouštět jak za pomocí konzole, tak prostřednictvím kódu. Výběr nabízených funkcí pro obě případy jsou rozepsány v následujících sekcích.

A.2.1 Konzole

Z konzole je přístupná čtveřice příkazů, kde tři z nich slouží k různým typům extrakce a poslední k měření výsledků. Jednotlivé příkazy používají jako vstup a výstup cestu k souborům ve formátu `csv`. Výchozím nastavením souboru je použití `;` jako oddělovače hodnot (`sep`), `'` pro označování textových řetězců (`quotechar`) a `utf8` jako kódování souboru (`encoding`). Tyto nastavení lze dodatečně měnit díky specifikaci možností přímo v příkazu.

Extrakční příkazy očekávají na vstupu soubor ve formátu:

- **id**: id webové stránky ve formátu `nazevfirmy_cz`,
- **url**: konkrétní url, ke které se váže uložený obsah,
- **encoding**: kódování získané stránky,
- **content**: html kód uložené webové stránky.

Výstupem extrakce je soubor ve formátu:

- **id**: id webové stránky ve formátu `nazevfirmy_cz`,
- **company_name**: jméno společnosti nebo podnikatele,
- **company_in**: IČ podnikatele
- **company_address**: adresa sídla společnosti
- **phones**: seznam telefonních čísel v mezinárodním formátu

- **emails**: seznam e-mailů
- **addresses**: seznam adres provozoven, prodejen, skladů a další přidružených nemovitostí,
- **opening_hours**: otevírací hodiny ve standardizovaném formátu,
- **facebook**: odkaz na Facebook stránku,
- **instagram**: odkaz na Instagram profil,
- **twitter**: odkaz na Twitter profil,
- **linkedin**: odkaz na Linkedin profil,
- **youtube**: odkaz na Youtube kanál.

Extrakce z metadata

Příkaz extrahuje informace o firmách z dostupných metadat webových stránek na vstupu. Podporovány jsou standardy `json-ld`, `microdata`, `micformat` a `rdfa`.

```
python -m cw_information_extractor metadata [OPTIONS] INPUT_PATH OUTPUT_PATH
```

Na vstupu příkaz očekává dvojici argumentů:

- **INPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, který obsahuje obsah webů pro extrakci uloženém ve zmiňované podobě;
- **OUTPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, do kterého se uloží výstup extrakce v popsané podobě.

Extrakce za použití regulárních výrazů

Příkaz extrahuje informace o firmách z dostupných webových stránek na vstupu pomocí regulárních výrazů a dodatečných pravidel. Regulární výrazy jsou uzpůsobené českým firmám a jejich webům.

```
python -m cw_information_extractor regex [OPTIONS] INPUT_PATH OUTPUT_PATH
```

Na vstupu příkaz očekává dvojici argumentů:

- **INPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, který obsahuje obsah webů pro extrakci uloženém ve zmiňované podobě;
- **OUTPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, do kterého se uloží výstup extrakce v popsané podobě.

Extrakce za použití strojového učení

Příkaz extrahuje informace o firmách z dostupných webových stránek na vstupu pomocí klasifikace vhodných částí HTML stránky a následném použití NER nebo dodatečných pravidel. Tento příkaz je závislý na externím webovém API poskytující NER, který vznikl během práce Straková et al. 2019.

```
python -m cw_information_extractor classifier [OPTIONS] INPUT_PATH OUTPUT_PATH
```

Na vstupu příkaz očekává dvojici argumentů:

- **INPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, který obsahuje obsah webů pro extrakci uloženém ve zmiňované podobě;
- **OUTPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, do kterého se uloží výstup extrakce v popsané podobě.

Měření výsledků

Příkaz měří výsledky extrakce za pomoci srovnávání s datovým souborem, obsahujícím anotované údaje z webových stránek. Výstupem je statistika průměrné míry shody, počty a míry falešné pozitivních a negativních vzorků pro každou kategorii.

```
python -m cw_information_extractor measure_results [OPTIONS] EXTRACTION_INPUT_PATH
↪  GROUND_TRUTH_INPUT_PATH OUTPUT_PATH
```

Na vstupu příkaz očekává trojici argumentů:

- **EXTRACTION_INPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, který obsahuje obsah webů pro extrakci uloženém ve zmiňované podobě;
- **GROUND_TRUTH_INPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, který obsahuje anotovaný obsah webů ve shodné podobě s výstupem extrakce;
- **OUTPUT_PATH**: absolutní/relativní cesta k souboru ve formátu `csv`, do kterého se uloží výstup měření.

A.2.2 Kód

Funkce spouštěné z kódu se liší od konzolových zejména ve vstupních parametrech. Jsou přizpůsobené na práci v prostředí `Jupyter` a proto na vstupu očekávávají `pandas.Dataframe` místo cesty k souborům.

Načítaní datových souborů

Převod uložených `csv` souboru do reprezentace pomocí `pandas.Dataframe` v očekávaném formátu.

```
cw_information_extractor.load_input_dataframe(input_path, sep, quotechar, encoding)
cw_information_extractor.load_extracted_dataframe(input_path, sep, quotechar,
↪  encoding)
```

Extrakce

Extrakční funkce fungují na stejných principech jako konzolové funkce. Rozdílem mimo vstupu je i výstup, který se vrací opět v podobě `pandas.Dataframe` místo uložení do souboru.

```
cw_information_extractor.extraction_using_metadata(input_df)
cw_information_extractor.extraction_using_regex(input_df)
cw_information_extractor.extraction_using_classifier(input_df)
```

Měření výsledků

Funkce pro měření funguje na stejných principech jako konzolové funkce. Rozdílem mimo formy vstupu je i výstup, který se vrací opět v podobě `pandas.Dataframe` místo uložení do souboru.

```
cw_information_extractor.create_metrics_dataframe(extraction_df, ground_truth_df)
```

A.3 Licence

Extraktor je kvůli použití NER jako webové služby licencován pod CC BY-NC-SA.

Příloha B

Seznam použitých zkratek

API Application Programming Interface

B2B Business to business

B2C Business to customer

CBOW Continuous Bag of Words

CSS Cascading Style Sheets

CSV Comma-separated values

FB Facebook

FN False negatives

FP False positives

HTML Hypertext Markup Language

IDF Inverse document frequency

IG Instagram

IOB Inside–outside–beginning

JSON JavaScript Object Notation

k-NN k-nearest neighbors

NER Named entity recognition

RDF Resource Description Framework

SVM Support vector machine

TF Term frequency

TN True negatives

TP True positives

URL Uniform Resource Locator

VISP Vision-based Page Segmentation Algorithm

XML Extensible Markup Language

YTB Youtube

Příloha C

Obsah přiložené paměťové karty

- `readme.txt` ..... stručný popis obsahu paměťové karty
- `src` ..... adresář se zdrojovými kódy implementace
  - `crawler` ..... adresář s jednoduchým crawlerem webu Firmy.cz
    - `spiders` .... adresář s implementací jednotlivých kroků crawlování
  - `datasets` ..... adresář se vstupními a výstupními datovými soubory
    - `ground_truth.csv` ..... anotovaný dataset údajů z firemních webů
    - `pages.csv` ..... dataset pro extrakci údajů uložených webů
    - `training.csv` ..... dataset pro trénování klasifikátoru
    - `extraction_metadata.csv` ... dataset extrakce s využitím metadat
    - `extraction_ml.csv` .. dataset extrakce s využitím strojového učení
    - `extraction_regex.csv` .... dataset extrakce s využitím regulárních výrazů
  - `extractor` ..... adresář s knihovnou pro extrakci
  - `notebooks`
    - `classifier.ipynb` ..... předzpracování dat, trénování a výběr klasifikátoru
    - `metrics.ipynb` ..... měření úspěšnosti extrakce
- `text` ..... adresář s textem práce
  - `latex` ..... adresář se zdrojovými kódy práce v LaTeXu
  - `DP_Stanovcak_Tomas_2022.pdf` ..... práce ve formátu PDF