

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra geoinformatiky

POTENCIÁL WEB SCRAPINGU PROSTOROVĚ LOKALIZOVANÝCH DAT

Diplomová práce

Bc. Adam JIRČÍK

Mgr. Karel MACKŮ, PhD

Olomouc 2026

Geoinformatika a kartografie

ANOTACE

Diplomová práce se zabývá potenciálem web scrapingu při získávání prostorově lokalizovatelných dat z webového prostředí. Cílem práce je analyzovat možnosti automatizovaného sběru dat z různých typů webových zdrojů a posoudit jejich využitelnost pro následné prostorové zpracování, analýzu či vizualizaci. Práce se zaměřuje na strukturovaná, polostrukturovaná i nestrukturovaná data, která mohou obsahovat prostorovou informaci buď přímo ve formě souřadnic či adres, nebo nepřímo prostřednictvím textových zmínek o lokalitách.

V rámci řešení byly testovány různé přístupy k získávání dat, zejména klasický web scraping, využití API a RSS kanálů. Pro praktické ověření byly zvoleny zdroje Bezrealitky.cz, ČT24, Deutsche Welle a The Guardian. Získaná data byla dále zpracována pomocí metod extrakce geografické informace, včetně regulárních výrazů, Named Entity Recognition, velkých jazykových modelů a geokódování. Výsledná prostorová data byla uložena do formátu GeoJSON, či prostorové databáze a vizualizována pomocí webových map.

Výsledky práce ukazují, že web scraping může být užitečným nástrojem pro získávání aktuálních prostorových dat z webu, zejména tam, kde nejsou data dostupná prostřednictvím otevřených dat nebo standardizovaných rozhraní. Současně práce upozorňuje na technické, právní a etické limity tohoto přístupu.

KLÍČOVÁ SLOVA

Web scraping; prostorová data; geoparsing; geokódování

Počet stran práce: 51

Počet příloh: 5 (z toho 5 volných)

ANNOTATION

The master thesis deals with the potential of web scraping for obtaining spatially localisable data from the web environment. The aim of the thesis is to analyse the possibilities of automated data collection from different types of web sources and to assess their usability for subsequent spatial processing, analysis and visualisation. The thesis focuses on structured, semi-structured and unstructured data, which may contain spatial information either directly in the form of coordinates or addresses, or indirectly through textual mentions of locations.

As part of the work, different approaches to data acquisition were tested, especially classic web scraping, the use of APIs and RSS feeds. The sources Bezrealitky.cz, ČT24, Deutsche Welle and The Guardian were selected for practical verification. The acquired data were further processed using methods for extracting geographical information, including regular expressions, Named Entity Recognition, large language models and geocoding. The resulting spatial data were stored in GeoJSON format or in a spatial database and visualised using web maps.

The results of the thesis show that web scraping can be a useful tool for obtaining up-to-date spatial data from the web, especially where such data are not available through open data sources or standardised interfaces. At the same time, the thesis highlights the technical, legal and ethical limitations of this approach.

KEYWORDS

Web scraping; spatial data; geoparsing; geocoding

Number of pages: 51

Number of appendixes: 5

Prohlašuji, že

- bakalářskou/diplomovou práci včetně příloh, jsem vypracoval(a) samostatně a uvedl(a) jsem všechny použité podklady a literaturu.

- jsem si vědom(a), že na moji bakalářskou/diplomovou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou/diplomovou práci užívat (§ 35 odst. 3),

- souhlasím, že údaje o mé bakalářské/diplomové práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské/diplomové práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské/diplomové práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

UNIVERZITA PALACKÉHO V OLOMOUCI

Přírodovědecká fakulta

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Bc. Adam JIRČÍK
Osobní číslo: R240570
Studijní program: N0532A330009 Geoinformatika a kartografie
Téma práce: Potenciál web scrapingu prostorově lokalizovatelných dat
Zadávající katedra: Katedra geoinformatiky

Zásady pro vypracování

Cílem práce je analyzovat potenciál web scrapingu při získávání prostorově lokalizovatelných dat a jejich následné využití. Student provede průzkum existujících datových zdrojů, které obsahují geografické informace, s důrazem na omezení vyplývající z licenčních a legislativních podmínek. Identifikuje konkrétní příklady webů či portálů, kde lze web scraping legálně aplikovat. Dále navrhne a realizuje techniky pro scraping prostorových dat z různých typů webů. Zohlední přitom odlišné formáty prezentace dat (HTML tabulky, JSON, API výstupy). Získaná data budou vhodně zpracována (např. geokódování a převedení prostorových formátů) pro další využití. Na základě extrahovaných prostorových dat student demonstruje jejich využití na příkladu reálného problému, například prostorové analýzy, vizualizace nebo modelování.

Text práce student zpracuje v souladu se závaznou šablonou pro kvalifikační práce KGI. O diplomové práci student vytvoří webovou stránku a poster. Celou práci (text, přílohy, výstupy, zdrojová a vytvořená data, poster a web) odevzdá student v digitální podobě na datové úložiště katedry. Do evidence STAG student odevzdá úplný text práce s přílohami, které určí vedoucí práce. Fyzicky student odevzdá pouze přílohy určené vedoucím práce.

Rozsah pracovní zprávy: max. 50 stran
Rozsah grafických prací: dle potřeby
Forma zpracování diplomové práce: elektronická

Seznam doporučené literatury:

- Brenning, A., & Henn, S. (2023). Web scraping: a promising tool for geographic data acquisition.
- Galvez-Hernandez, P., Gonzalez-Viana, A., Gonzalez-de Paz, L., Shankardass, K., & Muntaner, C. (2024). Generating Contextual Variables From Web-Based Data for Health Research: Tutorial on Web Scraping, Text Mining, and Spatial Overlay Analysis. *JMIR Public Health and Surveillance*, 10. <https://doi.org/10.2196/50379>
- Krotov, V., Johnson, L., & Silva, L. (2020). Legality and Ethics of Web Scraping. *Communications of the Association for Information Systems*, 47, 539–563. <https://doi.org/10.17705/1CAIS.04724>
- Lotfi, C., Srinivasan, S., Ertz, M., & Latrous, I. (2021). Web Scraping Techniques and Applications: A Literature Review. In *SCRS CONFERENCE PROCEEDINGS ON INTELLIGENT SYSTEMS* (pp. 381–394). Soft Computing Research Society. <https://doi.org/10.52458/978-93-91842-08-6-38>
- Pagallo, U., & Ciani, J. (2024). Anatomy of Web Data Scraping: Ethics, Standards, and the Troubles of the Law. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4707651>
- Souza, T. G. D., Fonseca, F. D. R., Fernandes, V. D. O., & Pedrassoli, J. C. (2021). EXPLORATORY SPATIAL ANALYSIS OF HOUSING PRICES OBTAINED FROM WEB SCRAPING TECHNIQUE. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2, 135–140. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-135-2021>

Vedoucí diplomové práce: Mgr. Karel Macků, Ph.D.
Katedra geoinformatiky

Datum zadání diplomové práce: 9. prosince 2024
Termín odevzdání diplomové práce: 4. května 2026

L.S.

prof. RNDr. Martin Kubala, Ph.D.
děkan



prof. RNDr. Vilém Pechanec, Ph.D.
vedoucí katedry

V Olomouci dne 10. prosince 2024

OBSAH

SEZNAM POUŽITÝCH ZKRATEK	8
ÚVOD	9
1 CÍLE PRÁCE	10
2 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	11
2.1 Porovnání s jinými metodami získávání dat z webu	11
2.2 Dělení web scrapingu	12
2.2.1 Podle architektury procesu	12
2.2.2 Podle způsobu získávání dat	12
2.2.3 Podle struktury dat	13
2.2.4 Podle metod extrakce	13
2.3 Využití web scrapingu	14
2.4 Získání prostorové informace z textových webových dat	15
2.5 Limity a problémy web scrapingu a získávání prostorových dat	16
3 METODY A POSTUP ZPRACOVÁNÍ	19
4 VLASTNÍ ŘEŠENÍ	22
4.1 Legislativní aspekt web scrapingu v ČR	22
4.1.1 Autorskoprávní rámec a ochrana databází	22
4.1.2 Zákonné licence a výjimky	23
4.1.3 Evropská právní úprava	23
4.1.4 Ochrana osobních údajů	24
4.1.5 Smluvní omezení a podmínky užití	24
4.1.6 Etické aspekty	24
4.1.7 Shrnutí legislativy a praktické vyvození	25
4.2 Bezrealitky.cz	26
4.3 ČT 24	32
4.4 Deutsche Welle (RSS)	37
4.5 The Guardian (API)	41
5 VÝSLEDKY	46
5.1 Výsledky scrapování webu bezrealitky.cz	46
5.2 Výsledky scrapování webu ČT24	46
5.3 Výsledky zpracování RSS feedu DW	47
5.4 Výsledky extrakce lokalit z API (The Guardian)	48
6 DISKUZE	49
7 ZÁVĚR	50
POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE	52
PŘÍLOHY	

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
AI	Artificial Intelligence
API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CSS	Cascading Style Sheets
DDL	Data Definition Language
DOM	Document Object Model
GIS	geografický informační systém
GPE	Geopolitical entity
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
LLM	Large Language Models
LOC	Location
NER	Named-entity recognition
NLP	Neurolingvistické programování
OGC	Open Geospatial Consortium
OSM	OpenStreetMap
RSS	Really Simple Syndication
URI	Unified Resource Identifier
URL	Uniform Resource Locator
WFS	Web Feature Service
XML	eXtensible Markup Language

ÚVOD

Webové stránky obsahují data, která mohou mít významný geografický rozměr. Tato data však často nejsou publikována ve formě otevřených datových sad ani prostřednictvím standardizovaných geoinformačních služeb. Mnohdy jsou dostupná pouze jako součást HTML stránek, textových článků nebo jiných webových struktur, které nejsou primárně určeny pro strojové zpracování. Právě zde se otevírá prostor pro využití web scrapingu jako metody automatizovaného získávání dat z webu.

Web scraping umožňuje získávat aktuální data z různorodých webových zdrojů a převádět je do podoby vhodné pro další zpracování. V kontextu geoinformatiky je jeho význam spojen především s možností získávat informace, které mají přímou nebo nepřímou prostorovou vazbu. Přímá prostorová informace může být uvedena například formou adresy nebo souřadnic, zatímco nepřímá prostorová informace se často vyskytuje v textu jako zmínka o lokalitě, městě, regionu nebo státu. Aby bylo možné taková data využít, je nutné je nejen získat, ale také správně identifikovat, extrahovat a geokódovat.

Tato práce se zaměřuje na možnosti využití web scrapingu při získávání prostorově lokalizovatelných dat z různých typů webových zdrojů. Pozornost je věnována jak klasickému scrapingu HTML stránek, tak alternativním způsobům získávání dat prostřednictvím API a RSS kanálů. Zároveň jsou řešeny metody extrakce geografické informace z textových dat, zejména pomocí metod zpracování přirozeného jazyka, velkých jazykových modelů a regulárních výrazů. Součástí práce je také převod získaných lokalit na souřadnice a jejich následná webová vizualizace pomocí interaktivních mapových aplikací.

Praktická část práce ověřuje navržené postupy na vybraných zdrojích, které reprezentují různé typy webových dat. Jedná se o realitní portál Bezrealitky.cz, zpravodajský web ČT24, RSS kanál Deutsche Welle a API deníku The Guardian. Tyto zdroje umožňují porovnat rozdílné přístupy ke sběru dat i rozdílné způsoby získávání prostorové informace. Výsledkem práce je posouzení využitelnosti web scrapingu jako nástroje pro získávání aktuálních prostorových dat a zároveň zhodnocení jeho technických, datových, právních a etických omezení.

1 CÍLE PRÁCE

Cílem této diplomové práce je analyzovat potenciál web scrapingu při získávání prostorově lokalizovatelných dat a jejich následném využití. Práce se proto zaměřuje nejen na samotné získávání dat, ale také na způsoby extrakce a interpretace geografické informace z různých typů webových zdrojů.

V rámci hlavního cíle byly stanoveny následující dílčí cíle:

Prvním dílčím cílem je provést rešerši a analýzu existujících datových zdrojů dostupných na webu, které obsahují geografickou informaci. Součástí této analýzy je i posouzení omezení vyplývajících z licenčních podmínek, legislativního rámce a etických aspektů využití web scrapingu.

Druhým dílčím cílem je identifikovat a vybrat konkrétní webové zdroje reprezentující různé způsoby publikace dat, tedy strukturovaná data (např. JSON), polostrukturovaná data (HTML) i alternativní přístupy, jako jsou RSS feedy a API. Tyto zdroje slouží jako základ pro návrh a implementaci jednotlivých metod získávání dat.

Dalším dílčím cílem je navrhnout a implementovat techniky pro automatizované získávání dat z webového prostředí. V práci jsou testovány různé přístupy ke scrapingu, včetně klasického parsování HTML pomocí knihoven, jako je BeautifulSoup, a přístupu založeného na simulaci chování uživatele pomocí nástroje Selenium. U nestrukturovaných textových dat je nutné lokalitu extrahovat pomocí různých metod, konkrétně pomocí regulárních výrazů, metod zpracování přirozeného jazyka (NER) s využitím knihovny spaCy a pomocí velkých jazykových modelů (LLM). Tyto přístupy jsou následně porovnány z hlediska přesnosti a použitelnosti.

Posledním dílčím cílem je demonstrace využití získaných dat na konkrétních příkladech. V práci je ukázáno využití dat pro prostorovou analýzu a vizualizaci, prostřednictvím webových map a aplikací. Tím je ověřen praktický přínos web scrapingu jako nástroje pro získávání aktuálních prostorových dat.

2 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

V posledních desetiletích dochází k významné proměně způsobů získávání geografických dat, která úzce souvisí s rozvojem webu a digitálních technologií. Vedle tradičních zdrojů, jako jsou terénní měření, statistické databáze nebo oficiální registry, se stále více prosazují data dostupná na webu. Web se tak stává důležitým prostředím pro získávání informací o prostorových jevech, přičemž důležitou metodou jejich získávání může být web scraping.

Web scraping představuje proces automatizovaného získávání dat z webových stránek, který umožňuje extrahovat informace z různých typů online zdrojů. Na rozdíl od tradičních metod sběru dat poskytuje scraping možnost získat velké množství dat relativně rychle s nízkými náklady a umožňuje tento proces provádět opakovaně automaticky. Obrovskou předností této metody je především schopnost získávat data, která nejsou dostupná prostřednictvím standardních rozhraní, jako jsou API nebo otevřená data. Mnoho webových stránek obsahuje cenné informace s prostorovým kontextem, avšak tyto informace jsou často publikovány pouze ve formě HTML stránek a nejsou přímo určeny ke strojovému zpracování (Brenning a Henn, 2023.)

Web scraping představuje klíčový nástroj pro získání dat, která nejsou primárně určená k extrakci a využití pro pozdější analýzy. Brenning a Henn (2023) tvrdí, že data získaná z webu mohou zachycovat aktuální stav různých procesů, například vývoj cen nemovitostí, prostorové rozložení služeb nebo mediální pokrytí událostí. Mohou doplňovat tradiční datové zdroje, které bývají často časově zpožděné nebo agregované.

2.1 Porovnání s jinými metodami získávání dat z webu

Velmi rozšířenou metodou získávání dat z webového prostředí je využití aplikačních rozhraní (API). API poskytují přístup k datům prostřednictvím předem definovaných dotazů a vracejí data ve strukturovaných formátech, jako je JSON nebo XML, což usnadňuje jejich automatizované zpracování. Mitchell (2018) uvádí, že API představují stabilnější a spolehlivější způsob získávání dat než web scraping, protože nejsou přímo závislá na struktuře HTML dokumentu. Nevýhodou API je skutečnost, že rozsah dostupných dat je omezen na to, co poskytovatel služby zpřístupní. Kahlon a Singh (2025) uvádějí, že API často obsahují omezení v podobě limitů počtu požadavků nebo omezeného přístupu k datům, což může představovat překážku při rozsáhlejším sběru dat.

Další možností získávání dat z webového prostředí jsou otevřená data. Otevřená data jsou data, která jsou volně dostupná pro opakované použití a sdílení, obvykle bez významných právních nebo technických omezení (Open Knowledge Foundation, 2015). V praxi jsou tato data publikována prostřednictvím otevřených datových portálů nebo webových služeb a jsou zpravidla dostupná ve strukturovaných formátech, jako jsou CSV, JSON nebo XML, což usnadňuje jejich další zpracování. Attard et al. (2015) uvádějí, že otevřená data jsou navržena tak, aby byla strojově čitelná a snadno integrovatelná do různých aplikací. To představuje jejich hlavní výhodu oproti web scrapingu. Dostupnost otevřených dat ale závisí na rozhodnutí poskytovatele, přičemž publikovaná data nemusí vždy pokrývat všechny relevantní informace nebo být dostatečně aktuální.

Specifickou oblastí prostorových webových dat jsou syndikační formáty, například GeoRSS. GeoRSS rozšiřuje běžné RSS nebo Atom kanály o geografickou informaci a umožňuje tak předávat lokalizovaný obsah ve formě webového feedu (GeoRSS, 2006). Standard OGC GeoRSS Encoding definuje GeoRSS jako jednoduchý způsob, jak geograficky označovat obsah v RSS a Atom kanálech, tedy jak do webového obsahu zakódovat informaci o poloze (Open Geospatial Consortium, 2017). Výhodou GeoRSS je jeho jednoduchost a interoperabilita, nevýhodou je omezení na zdroje, které tento formát poskytují.

2.2 Dělení web scrapingu

Web scraping zahrnuje široké spektrum metod a přístupů, které se liší zejména způsobem získávání a zpracování dat. V odborné literatuře neexistuje jednotná klasifikace, nicméně jednotlivé studie se shodují na několika základních kritériích dělení, mezi které patří míra automatizace, fáze procesu, použité techniky a charakter zpracovávaných dat.

2.2.1 Podle architektury procesu

Z hlediska architektury je web scraping často chápán jako dvoufázový proces složený z web crawlingu a samotné extrakce dat. Kobayashi a Takeda (2000) popisují rozdíl následovně:

Web crawling představuje systematické procházení webových stránek za účelem jejich identifikace a stažení. Tento proces zahrnuje práci s URL adresami, správu požadavků a navigaci mezi stránkami pomocí hypertextových odkazů. Crawling tedy prochází web a vyhledává relevantní stránky, které vyhovují filtrům, či obsahují požadovanou informaci.

Na něj navazuje samotný scraping, který zahrnuje extrakci konkrétních informací z obsahu webových stránek. Tato fáze pracuje s již získaným HTML nebo jiným formátem dat a zaměřuje se na identifikaci relevantních prvků. Mutlu et al. (2024) uvádí, že oddělení těchto dvou fází je klíčové pro návrh škálovatelných scrapingových systémů, protože umožňuje optimalizovat každou část procesu samostatně.

2.2.2 Podle způsobu získávání dat

Způsob získávání dat z webového prostředí lze rozlišit podle toho, zda jsou požadovaná data dostupná přímo v odpovědi serveru, nebo zda vznikají až po vykreslení stránky v prohlížeči. Tento rozdíl je pro web scraping zásadní, protože přímo ovlivňuje volbu technického postupu i použitých nástrojů. Brenning a Henn (2023) upozorňují, že zvolená technologie scrapingu závisí na charakteristice webu a na tom, jakým způsobem jsou data zpřístupněna uživateli. Článek Kahlona a Singha (2025) navíc zdůrazňuje, že moderní weby často kombinují více způsobů načítání obsahu, což zvyšuje nároky na návrh scrapingového řešení.

V případě statických webových stránek je obsah dostupný přímo ve zdrojovém kódu HTML, který server vrací jako odpověď na HTTP požadavek. V takové situaci lze data získat

stažením HTML dokumentu a jeho následnou analýzou. Technicky se zde běžně používají knihovny pro odesílání HTTP požadavků a parsování odpovědi. Knihovna *Requests* slouží pro práci s HTTP komunikací v Pythonu. *Beautiful Soup* je určena pro získávání dat z HTML a XML dokumentů. Pro rozsáhlejší nebo systematické scrapování je často využíván framework *Scrapy*, který je určen pro extrakci dat z webů a zároveň poskytuje nástroje pro řízení crawlingu a zpracování odpovědi (Brenning a Henn, 2023).

Pro statický obsah je výhodné, že struktura dat bývá dostupná přímo v HTML nebo v odpovědích serveru ve formátu JSON. Prakticky to znamená, že uživatel může identifikovat relevantní HTML elementy, tabulky, nadpisy nebo textové bloky a extrahovat je pomocí selektorů nebo parseru. Takový postup bývá vhodný zejména tehdy, pokud cílový web nevyžaduje uživatelskou interakci a pokud se obsah nenačítá až dodatečně po načtení stránky. Brenning a Henn tento typ technické strategie zmiňují jako efektivní zejména tam, kde je cílem získat větší množství webových dat s rozumnými náklady.

Naopak u dynamických webových aplikací nebývá celý obsah přítomen v původní HTML odpovědi. Část dat se může načítat až po inicializaci stránky prostřednictvím JavaScriptu nebo následných asynchronních požadavků. V takovém případě samotné stažení HTML dokumentu nestačí, protože výsledná podoba stránky vzniká až v prostředí prohlížeče. Scrapy ve své oficiální dokumentaci proto doporučuje, aby se u dynamického obsahu nejprve zjistilo, zda data nepřicházejí v dalších síťových požadavcích, a pokud ne, lze použít nástroje schopné vykreslit stránku a spustit JavaScript (Scrapy, 2026).

Technicky se tento problém řeší nástroji pro automatizaci prohlížeče, simulací chování reálného uživatele v prohlížeči. Mezi nejčastěji používané nástroje patří *Selenium* a *Playwright*. *Scrapy* jej ve své dokumentaci uvádí jako vhodnou volbu pro práci s dynamicky načítaným obsahem. Tyto nástroje umožňují načíst stránku, počkat na vykreslení obsahu, případně kliknout na prvky rozhraní, posunout stránku nebo vyvolat další požadavky, a teprve poté extrahovat výsledný obsah.

2.2.3 Podle struktury dat

Z hlediska charakteru dat lze web scraping rozdělit podle úrovně jejich strukturovanosti.

Strukturovaná data jsou typicky reprezentována tabulkami nebo databázemi a jejich extrakce je relativně jednoduchá. Polostrukturovaná data, jako jsou HTML dokumenty, vyžadují analýzu struktury stránky a identifikaci relevantních prvků pomocí selektorů.

Největší výzvu představují nestrukturovaná data, zejména textové dokumenty. V těchto případech není možné spoléhat pouze na strukturu stránky a je nutné využít pokročilejší metody, například zpracování přirozeného jazyka. Tento typ dat je typický například pro zpravodajské články nebo obsah sociálních médií (Saurkar et al., 2018).

2.2.4 Podle metod extrakce

Zatímco způsob získávání dat určuje, jakým způsobem je získán obsah webových stránek, metody extrakce se zaměřují na samotné získání konkrétních informací z tohoto obsahu.

Volba vhodné metody extrakce závisí především na struktuře dat, stabilitě webové stránky a požadované míře automatizace. V odborné literatuře lze identifikovat několik základních přístupů, které se liší svou komplexitou i oblastí využití (Mitchell, 2018).

Nejjednodušší přístup představují metody založené na využití HTML struktury dokumentu a předem definovaných pravidel. Tyto metody využívají selektory, například CSS nebo XPath, případně regulární výrazy, pro identifikaci požadovaných prvků stránky. Jak uvádí Mitchell (2018), tyto techniky umožňují cíleně získávat data z konkrétních částí webové stránky na základě její struktury.

V praxi jsou tyto přístupy realizovány pomocí nástrojů pro parsování HTML dokumentu, které převádějí stránku do stromové reprezentace (DOM). Extrakce dat následně probíhá pomocí selektorů nebo dotazovacích jazyků, například XPath. Tyto metody jsou běžně implementovány v knihovnách, jako jsou *Beautiful Soup* nebo *lxml*. Jejich výhodou je jednoduchost, vysoká rychlost a nízká výpočetní náročnost, díky čemuž jsou vhodné zejména pro extrakci dat ze stránek s dobře definovanou a stabilní strukturou. Nevýhodou je jejich citlivost na změny HTML kódu, které mohou vést k nefunkčnosti procesu.

Specifickou kategorií představují metody založené na extrakci dat z nestrukturovaného textu. V těchto případech není možné spoléhat pouze na strukturu HTML dokumentu a je nutné využít pokročilejší přístupy. Hu et al., (2022) zmiňují metody zpracování přirozeného jazyka (NLP), které umožňují identifikovat relevantní informace přímo v textovém obsahu.

Pokročilé přístupy zahrnují také využití strojového učení, které umožňuje automaticky identifikovat relevantní informace na základě trénovacích dat. Tyto metody jsou flexibilnější než pravidlové přístupy a dokáží se lépe přizpůsobit různorodým strukturám dat. Na druhou stranu vyžadují větší množství dat pro trénování a jsou výpočetně náročnější.

V literatuře se často objevují přístupy, které kombinují více metod extrakce dat. Například strukturovaná data mohou být získána pomocí analýzy HTML nebo DOM struktury, zatímco nestrukturovaný text je dále zpracován pomocí metod zpracování přirozeného jazyka. Hu et al., (2022) doporučují kombinaci těchto přístupů v rámci jednoho systému.

2.3 Využití web scrapingu

Web scraping nachází široké uplatnění v různých oblastech výzkumu i praxe, přičemž jeho hlavní přínos spočívá v možnosti získávat velké objemy dat z webového prostředí. Odborná literatura ukazuje, že scraping je využíván napříč disciplínami, od ekonomie přes sociální vědy až po geografii, přičemž konkrétní aplikace se liší podle typu získávaných dat a cílů analýzy. Ve své práci jej využili Cavallo (2018), Mutlu (2024), Hu (2022) či Middleton (2018).

Jednou z nejčastějších oblastí využití web scrapingu je ekonomie a analýza trhů. Cavallo (2018) ukazuje, že web scraping může být efektivně využit pro sledování cen produktů v online obchodech. Na základě dat získaných z webových stránek e-commerce platform lze analyzovat dynamiku cen, inflaci nebo cenové rozdíly mezi regiony. Tento přístup umožňuje získat detailnější a aktuálnější data než tradiční statistické zdroje, které bývají publikovány s časovým zpožděním.

Další významnou oblastí je analýza realitního trhu. Web scraping je zde využíván pro získávání dat o cenách nemovitostí, jejich lokalitě a charakteristikách. Například studie využívající data z realitních portálů od Brenninga a Henna (2023) ukazují, že scraping umožňuje analyzovat prostorové vzorce cen nemovitostí a jejich vývoj v čase. Tato data jsou následně často importována do GIS, což umožňuje jejich prostorovou analýzu.

Web scraping je rovněž široce využíván při analýze sociálních médií a online komunikace. Middleton et al. (2018) uvádějí, že data získaná z webových zdrojů, jako jsou sociální sítě nebo zpravodajské weby, mohou být využita pro identifikaci událostí a jejich lokalizaci. Tento přístup je významný zejména v oblasti krizového řízení, kde umožňuje sledovat vývoj situace v reálném čase a získávat informace z různých zdrojů současně.

Další oblastí využití je analýza textových dat a médií. Web scraping umožňuje získávat velké množství textových dokumentů, například zpravodajských článků, které lze následně analyzovat pomocí metod zpracování přirozeného jazyka. Hu et al. (2022) uvádějí, že tyto metody umožňují extrahovat informace o lokalitách, událostech nebo aktérech přímo z textu, což rozšiřuje možnosti využití webových dat v různých oblastech výzkumu.

Specifickou oblastí je také využití web scrapingu v akademickém výzkumu. Mutlu et al. (2024) popisují aplikaci web scrapingu pro automatizované vyhledávání a sběr vědeckých publikací, což umožňuje efektivnější práci s odbornou literaturou. Tento přístup ukazuje, že scraping není pouze nástrojem pro získávání dat z webových stránek, ale může být využit i pro podporu samotného výzkumného procesu.

Z uvedených příkladů vyplývá, že web scraping představuje univerzální nástroj pro získávání dat z webového prostředí, který nachází uplatnění v širokém spektru aplikací. Společným jmenovatelem těchto přístupů je schopnost pracovat s velkým množstvím dat, jejich aktuálnost a flexibilita. Zároveň však jednotlivé studie upozorňují na nutnost řešit problémy spojené s kvalitou dat, jejich strukturou a dostupností, což představuje důležitý aspekt při jejich dalším zpracování.

2.4 Získání prostorové informace z textových webových dat

Prostorová informace může být v datech uvedena přímo, například ve formě geografických souřadnic nebo adres, případně nepřímo prostřednictvím textových zmínek o geografických názvech. Tento rozdíl ovlivňuje způsob jejího zpracování a vede k využití odlišných metod pro její získání a interpretaci.

Přímá prostorová informace je reprezentována přímo pomocí geografických souřadnic nebo jednoznačně identifikovatelných adres. V takových případech lze lokalizaci získat relativně jednoduše, například geokódováním. Tento přístup je však omezený na data, která již prostorovou informaci obsahují ve strukturované podobě. Naopak nepřímá lokalizace vyžaduje její identifikaci z textového obsahu, což Hu et al. (2022) považuje za výrazně složitější úlohu.

Proces získávání lokace z textu je v literatuře často označován jako geoparsing. Tento proces zahrnuje dvě hlavní fáze: identifikaci geografických entit v textu a jejich následné přiřazení ke konkrétní lokaci. První fáze je realizována pomocí metod rozpoznávání pojmenovaných entit (Named Entity Recognition, NER), které umožňují identifikovat výskyty geografických názvů v textu, například názvy měst, států nebo regionů. Druhá fáze zahrnuje určení konkrétního geografického objektu, ke kterému se daný název vztahuje (Hu et al., 2022).

Význam geoparsingu je patrný zejména při práci s nestrukturovanými textovými daty, jako jsou zpravodajské články nebo příspěvky na sociálních médiích. Middleton et al. (2018) uvádějí, že kombinace extrakce textových dat z webu a jejich následného zpracování pomocí NLP metod umožňuje identifikovat události a jejich lokalizaci. Tento přístup je využíván například v oblasti krizového řízení, kde je důležité získávat aktuální informace o probíhajících událostech z různých zdrojů.

Geokódování, podle Middletona et al. (2018), představuje další důležitou součást procesu získávání prostorové informace. V případě, že je lokalita identifikována například ve formě názvu místa nebo adresy, je možné ji převést na geografické souřadnice pomocí geokódovacích služeb. Tento krok je nezbytný pro další prostorovou analýzu dat, například v prostředí GIS. Přesnost geokódování však závisí na kvalitě vstupních dat a dostupnosti referenčních databází.

Získávání lokace z textu je však spojeno s řadou problémů. Jedním z hlavních je nejednoznačnost geografických názvů, kdy jeden název může označovat více různých míst. Například název „Chodov“ může odkazovat na pražskou část, ale i na menší město v západních Čechách. Dalším problémem je variabilita jazykových struktur, kdy mohou být lokace v textu uvedeny různými způsoby, například pomocí zkratk nebo neformálních názvů.

Další výzvou je kontextová závislost, kdy správná interpretace lokace závisí na širším kontextu textu. Například zmínka o názvu města nemusí nutně znamenat, že se daná událost odehrála právě v této lokalitě. Middleton et al. (2018) uvádějí, že pro zvýšení přesnosti lokalizace je proto vhodné kombinovat informace z více částí textu a využívat kontextové informace.

2.5 Limity a problémy web scrapingu a získávání prostorových dat

Přestože web scraping představuje efektivní nástroj pro získávání dat z webového prostředí, jeho využití je spojeno s řadou technických, datových i metodologických omezení. Hu et al. (2022) uvádějí, že tyto problémy se dále prohlubují v případě práce s prostorově lokalizovanými daty, zejména pokud je lokalita obsažena pouze nepřímo v textu, viz kapitola Limity při získávání prostorové informace.

Technické limity

Jedním z hlavních problémů web scrapingu je závislost na struktuře webových stránek. Pravidlové metody extrakce, například využívající XPath nebo CSS selektory, jsou citlivé na změny HTML struktury. I drobná úprava webové stránky může vést k nefunkčnosti scrapingového řešení, což vyžaduje jeho průběžnou údržbu.

Další technický problém představuje dynamický obsah, vysvětlený v kapitole 2.2.2. Brenning a Henn (2023) považují tento přístup za výpočetně náročnější, který zpomaluje proces sběru dat.

Významným omezením web scrapingu jsou také ochranné mechanismy webových stránek, které jsou navrženy k detekci a blokování automatizovaného přístupu. Jak uvádí Martínéz Llamas et al. (2025), moderní webové aplikace využívají pokročilé metody detekce botů, které analyzují chování uživatele, charakteristiky požadavků nebo síťovou komunikaci s cílem odlišit automatizované skripty od reálných uživatelů.

Mezi běžně používané techniky patří například omezení počtu požadavků, blokování IP adres nebo využití interaktivních ověřovacích mechanismů, jako jsou CAPTCHA, které slouží k rozlišení člověka a automatizovaného programu. Tyto přístupy mohou výrazně komplikovat nebo zcela znemožnit automatizovaný sběr dat, a jejich přítomnost je jedním z klíčových faktorů ovlivňujících návrh scrapingového řešení.

Datové limity

Data získaná pomocí web scrapingu jsou často nestrukturovaná, neúplná nebo obsahují chyby. Webové stránky nejsou primárně navrženy pro strojové zpracování, což znamená, že získaná data mohou vyžadovat rozsáhlé čištění a transformaci (Brenning a Henn, 2023).

Dalším problémem je heterogenita datových zdrojů. Různé webové stránky mohou prezentovat podobná data odlišným způsobem, což komplikuje jejich sjednocení a následnou analýzu. Kahlon a Singh (2024) uvádějí, že tato variabilita představuje významnou výzvu při návrhu univerzálních scrapingových nástrojů.

Limity při získávání prostorové informace

Specifickou skupinu problémů představují limity spojené se získáváním prostorové informace z textových dat. Jak uvádějí Middleton et al. (2018), jedním z hlavních problémů je nejednoznačnost geografických názvů, kdy jeden název může označovat více různých lokalit. Správné určení lokace proto vyžaduje zohlednění kontextu textu.

Middleton et al. (2018) dále upozorňují, že zmínka o určité lokalitě v textu nemusí nutně znamenat, že se daná událost odehrála právě v této lokalitě. Správná interpretace prostorové informace proto vyžaduje analýzu širšího kontextu.

Právní a etické aspekty

Brenning a Henn (2023) upozorňují, že web scraping je rovněž spojen s právními a etickými otázkami. Mnoho webových stránek omezuje automatizovaný přístup prostřednictvím podmínek užívání nebo technických opatření. Porušení těchto pravidel může vést k právním důsledkům.

Dalším aspektem je ochrana osobních údajů, zejména při práci s daty ze sociálních médií nebo jiných zdrojů obsahujících osobní informace. Je proto nutné zohlednit legislativní rámec a etické principy při získávání a zpracování dat.

Více se touto problematikou zabývá kapitola Legislativní aspekt web scrapingu v ČR.

3 METODY A POSTUP ZPRACOVÁNÍ

Hlavní metodou práce je web scraping, tedy automatizované získávání dat z webového prostředí. V rámci řešení web scraping zahrnuje různé techniky získávání dat podle struktury zdrojových. Byly implementovány následující přístupy:

- prostřednictvím web scrapingu (Bezrealitky),
- prostřednictvím web scrapingu (ČT24),
- prostřednictvím API (Guardian),
- prostřednictvím RSS kanálu (Deutsche Welle).

Tyto přístupy odpovídají běžně využívaným metodám získávání dat v prostředí webu, přičemž jejich vhodnost závisí na dostupnosti strukturovaného rozhraní. Jedná se o metody klasického scrapingu a byly vyzkoušeny i alternativní metody pro možné porovnání.

Z technického hlediska byly jednotlivé varianty web scrapingu realizovány pomocí různých knihoven jazyka Python. Pro komunikaci s webovými službami byla využita knihovna *requests*, která umožňuje odesílání HTTP požadavků a práci s odpověďmi serveru. V případě zpracování HTML dokumentů byla použita knihovna *BeautifulSoup*, která umožňuje parsování DOM struktury a extrakci konkrétních prvků na základě tagů a selektorů. Pro zpracování RSS kanálů byla využita knihovna *feedparser*, která převádí XML strukturu na objektovou reprezentaci.

V případě webu ČT24 byl využit nástroj *Selenium WebDriver*, který umožňuje simulovat chování uživatele v prohlížeči a získat výsledný obsah stránky po vykonání JavaScriptu. *Selenium* bylo provozováno v headless režimu a doplněno o nástroj *webdriver-manager* pro správu ovladače. Tento přístup je výpočetně náročnější, ale umožňuje získat kompletní obsah moderních webových stránek.

Pro optimalizaci scrapingu webu ČT24 byla využita XML sitemap, která obsahuje seznam URL adres článků. Tento přístup umožňuje získání relevantních stránek bez nutnosti procházení webu pomocí odkazů a současně umožňuje filtrování článků podle data publikace.

Heuristická extrakce

V případě dat z Guardian API byla použita metoda založená na regulárních výrazech, která identifikuje jazykové vzory typu „in London“ nebo „from Paris“. Extrakce probíhá z více částí článku (byline, headline, text).

NLP přístup

V případě RSS dat Deutsche Welle byla použita metoda Named Entity Recognition (NER) pomocí knihovny *spaCy*. Model identifikuje entity typu *GPE* a *LOC*, které odpovídají geografickým lokalitám.

Extrakce byla doplněna o ohodnocení:

- vyšší váha titulku,
- preference víceslovných názvů,
- penalizace krátkých výrazů.

AI přístup

U dat z ČT24 byla použita metoda založená na využití velkých jazykových modelů (LLM), konkrétně služeb OpenAI, Gemini a Claude. Modely byly využity pro extrakci lokalit na základě promptu, který požaduje návrat strukturovaného JSON výstupu.

Geokódování

Ve všech přístupech byla použita metoda geokódování, tedy převod textového názvu lokality na geografické souřadnice pomocí služby Nominatim (OpenStreetMap). Geokódování lze obecně chápat jako proces transformace textového popisu lokace (např. názvu místa nebo adresy) na geografické souřadnice (φ , λ) (Goldberg et al., 2007). Tento proces lze vyjádřit jako zobrazení:

$$f: L \rightarrow (\varphi, \lambda)$$

kde L je název lokality a (φ, λ) její geografická poloha.

Databázové a publikační metody

V případě realitních dat (Bezrealitky) byla data ukládána do databáze PostgreSQL/PostGIS a následně agregována pomocí materialized views.

Data byla publikována prostřednictvím serveru GeoServer jako služba typu WFS, což odpovídá standardům OGC (OGC, 2019). Tento přístup umožňuje oddělení datové a prezentační vrstvy.

Kartografická vizualizace

Vizualizace dat byla realizována pomocí knihovny *Leaflet*, která umožňuje tvorbu interaktivních webových map. Byly použity metody:

- metoda bodových znaků,
- shlukování prvků (marker clustering),
- tematická klasifikace podle kategorií.

Použitá data

Byly využity následující datové zdroje:

- **Guardian API** – strukturovaná data článků,
- **RSS kanál Deutsche Welle** – XML data,
- **web ČT24** – sitemap + HTML obsah,
- **Bezrealitky.cz** – realitní data získaná scrapingem,
- **OpenStreetMap (Nominatim)** – geokódovací data,
- **ArcČR 500 verze 3.3** – geometrie krajů, okresů a ČR.

Použité programy

Zpracování bylo realizováno v následujícím prostředí:

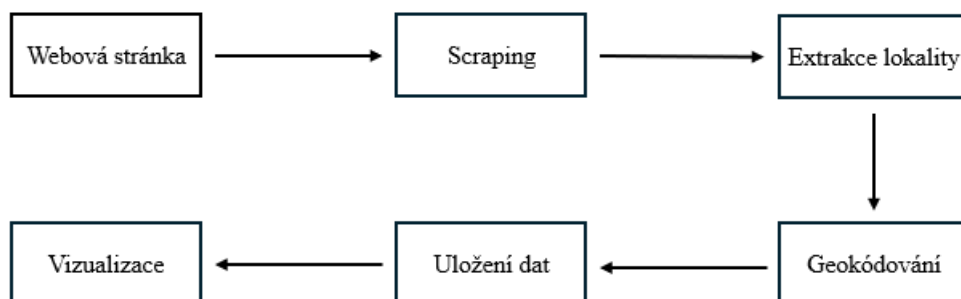
- **Python 3.13.2**
 - Knihovny: requests, BeautifulSoup, feedparser, Selenium, spaCy,
- **Flask** – API pro webovou aplikaci,
- **PostgreSQL 18 + PostGIS 3.6.0** – ukládání dat,
- **GeoServer** – publikace dat,
- **Leaflet** – webová vizualizace,
- **OpenAI, Gemini, Claude API** – AI extrakce lokalit.

Postup zpracování

Zpracování dat se lišilo podle typu datového zdroje, ale mělo společnou logiku.

Nejprve byla získána vstupní data (API, RSS, scraping). Následně byl z textových částí článků extrahován seznam kandidátních lokalit pomocí jedné ze tří metod (regex, NER, AI). Poté byly tyto lokality geokódovány a převedeny na souřadnice.

Výsledná data byla transformována do formátu GeoJSON nebo uložena do databáze. V případech zpracování dat z webů ČT24 a Bezrealitky.cz byla data dále kartograficky vizualizována pomocí webové mapy. Pro scraping dat z webu ČT24 byla vytvořena webová aplikace.



Obr. 1 Workflow skriptů

4 VLASTNÍ ŘEŠENÍ

Tato kapitola popisuje praktickou část práce zaměřenou na automatizované získávání dat z webových zdrojů a následné zpracování jejich prostorové informace. Ta se na webu může vyskytovat v různých podobách. Může být explicitně uvedena pomocí souřadnic či adresy, nebo jako zmínka o lokalitě uprostřed textu. V případě textových dat je proto nutné tuto informaci nejprve extrahovat, a to různými metodami, které se liší svou přesností i schopností zachytit kontext. V práci jsou porovnány přístupy založené na velkých jazykových modelech (AI), nástrojích pro zpracování přirozeného jazyka, konkrétně knihovně spaCy, a také jednodušší metodě využívající regulární výrazy.

Nejprve je vymezen legislativní a etický rámec web scrapingu v prostředí České republiky. Následně jsou popsány jednotlivé vytvořené skripty, které využívají odlišné přístupy ke sběru dat. V rámci klasického web scrapingu jsou použity techniky založené na knihovnách BeautifulSoup a Selenium, které se liší způsobem načítání a zpracování webových stránek. Vedle toho jsou představeny také RSS feed a API jako alternativní způsoby získávání dat, které v některých případech umožňují přístup ke strukturovanějším informacím bez nutnosti přímého parsování HTML.

U každého řešení jsou popsány použité knihovny, vstupní požadavky, princip fungování skriptu, způsob extrakce prostorových informací, geokódování a forma výsledného výstupu.

4.1 Legislativní aspekt web scrapingu v ČR

V současné době neexistuje právní předpis, který by web scraping explicitně upravoval, jeho legalita je proto odvozována z obecnějších právních institutů, jako jsou autorské právo, ochrana databází, ochrana osobních údajů nebo smluvní závazky. Tento stav vede k tomu, že web scraping je v odborné literatuře často označován jako „šedá zóna“, jejíž právní posouzení závisí na konkrétních okolnostech daného případu (Krotov; Silva, 2018).

Vedle právních aspektů hraje významnou roli také rovina etická, která reflektuje dopady této činnosti na jednotlivce, organizace i webové prostředí jako celek. Jak upozorňují Krotov a Silva (2018), i jednání, které je formálně v souladu s právem, může vyvolávat etické otázky, zejména v souvislosti s ochranou soukromí, férovým využitím dat a ekonomickými dopady.

4.1.1 Autorskoprávní rámec a ochrana databází

Základním právním předpisem v českém právním řádu je Zákon č. 121/2000 Sb., autorský zákon, který upravuje ochranu autorských děl i databází.

Obsah webových stránek může být chráněn jako autorské dílo, pokud splňuje podmínku jedinečnosti. Z hlediska práva je významné zejména to, že automatizované získávání dat zpravidla zahrnuje jejich rozmnožování, což je činnost vyhrazená nositeli práv. Neoprávněné nakládání s chráněným obsahem tak může vést k porušení autorského práva (Telec; Tůma, 2019).

Vedle ochrany jednotlivých děl upravuje autorský zákon také tzv. zvláštní právo pořizovatele databáze, které chrání investici vloženou do jejího vytvoření. Toto právo umožňuje pořizovateli kontrolovat využití databáze jako celku nebo její podstatné části.

Z pohledu web scrapingu je tato úprava zásadní, neboť omezuje možnosti systematického získávání dat z rozsáhlých webových zdrojů.

4.1.2 Zákonné licence a výjimky

Autorský zákon ale obsahuje výjimky, které umožňují užití chráněného obsahu bez souhlasu nositele práv. Klíčové ustanovení představuje § 92, který upravuje bezúplatné zákonné licence.

Podle § 92 Zákon č. 121/2000 Sb., autorský zákon:

Do práva pořizovatele jím zpřístupněné databáze též nezasahuje oprávněný uživatel, který užívá podstatnou část obsahu databáze

- a) pro svou osobní potřebu; ustanovení § 30 odst. 3 zůstává nedotčeno,
- b) pro účely vědecké nebo vyučovací, uvede-li pramen, v rozsahu odůvodněném sledovaným nevýdělečným účelem, a
- c) pro účely veřejné bezpečnosti nebo správního či soudního řízení. „...nezasahuje oprávněný uživatel do práva pořizovatele databáze, pokud užívá podstatnou část databáze pro účely vědecké nebo vyučovací, uvede-li pramen, v rozsahu odůvodněném sledovaným nevýdělečným účelem.“

Toto ustanovení představuje významný nástroj vyvažování mezi ochranou práv a veřejným zájmem na přístupu k informacím. Jeho aplikace však vyžaduje splnění podmínky přiměřenosti, která je posuzována individuálně s ohledem na konkrétní účel užití. Jak uvádí Telec a Tůma (2019), zákonné licence nelze chápat jako obecné oprávnění k neomezenému využívání dat, ale jako výjimku, která musí být interpretována restriktivně.

4.1.3 Evropská právní úprava

Evropský právní rámec je doplněn Směrnicí 2019/790, která zavádí právní úpravu text and data miningu. Směrnice umožňuje reprodukci a extrakci dat pro účely jejich analýzy, zejména v kontextu vědeckého výzkumu. Tato úprava odráží technologický vývoj a potřebu pracovat s velkými objemy dat. Jak uvádí Rosati (2024), směrnice DSM reaguje na digitální proměnu autorského práva v Evropské unii a snaží se nastavit rámec, který umožňuje využívání digitálního obsahu pro nové technologické účely, včetně text and data miningu, při současném zachování ochrany práv nositelů autorských práv.

Je však třeba zdůraznit, že tato úprava se vztahuje především na samotnou analýzu dat a nepředstavuje obecné oprávnění k jejich neomezenému získávání. Právní posouzení proto nadále závisí na kombinaci evropské a národní úpravy.

4.1.4 Ochrana osobních údajů

Další významnou oblastí je ochrana osobních údajů, upravená zejména General Data Protection Regulation a Zákon č. 110/2019 Sb., o zpracování osobních údajů.

Zpracování osobních údajů podléhá přísným pravidlům, a to i v případě, že jsou data veřejně dostupná. Jak zdůrazňují Voigt a von dem Bussche (2017), dostupnost dat neznamená jejich volnou použitelnost.

Zásadní jsou zejména tyto principy:

- zákonnost a transparentnost,
- účelové omezení,
- minimalizace údajů,
- omezení doby uchování,
- zabezpečení dat.

Porušení těchto principů může vést k významným právním důsledkům, včetně sankcí ze strany dozorových orgánů.

4.1.5 Smluvní omezení a podmínky užití

Právní rámec web scrapingu je dále ovlivněn smluvními vztahy mezi uživatelem a provozovatelem webové služby. Podmínky užití mohou obsahovat omezení týkající se automatizovaného přístupu k datům a jejich dalšího využití, viz Obr. 2.

62. Je zakázáno: (i) neoprávněně přistupovat či manipulovat s daty na službě Sreality.cz; (ii) užívat softwarových či jiných počítačových programů za účelem vytěžení databáze služby Sreality.cz (zejména prostřednictvím scrapování dat nebo jiným obdobným způsobem); (iii) provádět tzv. URL redirect, kdy URL adresa v inzerátu na službě Sreality.cz nesmí směřovat zejména na komerční realitní servery, resp. servery se stejným či obdobným tematickým obsahem; (iv) vytvářet podmínky pro vznik nebo zneužití jakýchkoliv bezpečnostních slabín technického rozhraní služby Sreality.cz; (v) zasahovat do technických a/nebo obsahových prvků služby Sreality.cz (např. užitím spyware, virů, trojských koní, počítačových červů a obdobných programů porušujících integritu služby Sreality.cz).

Obr. 2 Podmínky na webu sreality.cz zakazující scraping

Podle Krotov a Silva (2018) může porušení těchto podmínek vést k odpovědnosti za porušení smlouvy, zejména pokud uživatel s podmínkami výslovně souhlasil. Vedle toho je relevantní i technický standard robots.txt, který sice není právně závazný, ale jeho respektování je vnímáno jako projev korektního chování v digitálním prostředí.

4.1.6 Etické aspekty

Etické otázky web scrapingu přesahují rámec právní regulace a týkají se širších dopadů na jednotlivce a organizace. Jak uvádí Krotov a Silva (2018), etické aspekty web scrapingu jsou dosud nedostatečně prozkoumány, přesto však zásadní. Autoři zdůrazňují, že uživatelé by měli zohledňovat možné důsledky svého jednání a posuzovat nejen jeho legalitu, ale i etickou přijatelnost.

Ochrana soukromí

Automatizované získávání dat může vést k narušení soukromí jednotlivců, zejména pokud jsou data kombinována s dalšími zdroji a umožňují identifikaci konkrétních osob. Problémem je také absence informovaného souhlasu subjektů údajů.

Ochrana organizací

Web scraping může odhalit citlivé informace o fungování organizací, včetně obchodních strategií nebo interních procesů. Takové jednání může mít negativní dopady na jejich konkurenceschopnost a reputaci.

Ekonomické dopady

Scraping může ovlivnit ekonomický model webových služeb, například obcházením reklam nebo vytvářením konkurenčních datových produktů. To může vést k nerovnováze mezi tvůrci dat a jejich uživateli.

4.1.7 Shrnutí legislativy a praktické vyvození

Posouzení přípustnosti web scrapingu nelze provádět obecně, ale vždy v závislosti na konkrétní kombinaci dvou základních faktorů – povaze získávaných dat a způsobu jejich následného využití. Právě tyto dvě kritéria určují, zda se dané jednání pohybuje v akceptovatelném rámci, nebo zda představuje zvýšené právní či etické riziko.

Z hlediska povahy dat je třeba rozlišovat zejména mezi obsahem, který má charakter tvůrčího vyjádření (např. texty či popisy), systematicky uspořádanými daty (např. katalogy, seznamy či inzeráty), a dále mezi samotnými fakty či metadata, jako jsou ceny, časové údaje nebo geografické informace. Významnou roli hraje rovněž skutečnost, zda zpracovávaná data obsahují informace vztahující se ke konkrétním osobám, neboť i veřejně dostupná data mohou mít citlivý charakter a jejich další využití může vést k nechtěné identifikaci jednotlivců (Voigt a von dem Bussche, 2017).

Neméně důležitým faktorem je účel, pro který jsou data využívána. V případě vědeckého nebo výzkumného využití je scraping považován za legální. Podobně i čistě osobní využití, které nevede k dalšímu šíření dat, představuje nízké riziko, pokud nedochází k zásahům do práv třetích osob. Naopak v případě komerčního využití se míra rizika výrazně zvyšuje. Pokud jsou data využívána k vytvoření produktu nebo služby, která konkuruje původnímu zdroji dat, může takové jednání vyvolávat otázky nejen právní, ale i etické.

Z praktického hlediska lze shrnout, že web scraping je obecně přípustný v případech, kdy je zaměřen na získávání dat v omezeném rozsahu, která jsou dále využívána především pro analytické, výzkumné nebo osobní účely, a kdy nedochází k přímému využívání obsahu v jeho původní podobě ani k narušení hodnoty původního zdroje. Naopak problematické jsou situace, kdy dochází k rozsáhlému získávání systematicky uspořádaných dat, jejich dalšímu šíření nebo využití k vytvoření konkurenční služby či produktu. V těchto případech je nezbytné věnovat zvýšenou pozornost nejen právním omezením, ale i etickým důsledkům takového jednání.

4.2 Bezrealitky.cz

Tato část práce popisuje skript určený pro automatizované získávání dat z realitních inzerátů na portálu Bezrealitky.cz. Skript je parametrický, lze spouštět různé kombinace typu transakce a typu nemovitosti, například pro pronájem bytů, prodej bytů, pronájem domů nebo prodej domů. Hlavním cílem skriptu je stáhnout dostupné inzeráty z webového rozhraní, extrahovat jejich základní atributy, převést adresu na bodová data geokódováním a následně tato data uložit do PostgreSQL databáze.

Softwarové a datové předpoklady

Před spuštěním skriptu je nutné mít připravené odpovídající programové prostředí. Skript zahrnuje moduly *os*, *json*, *time*, *argparse*, *pathlib* a *urllib.parse*. Tyto moduly jsou součástí standardní knihovny Pythonu, a proto by je nemělo být nutné instalovat samostatně.

Naopak je nutné doinstalovat následující externí balíčky:

- **requests** – slouží k odesílání HTTP požadavků na webový server a geokódovací službu,
- **beautifulsoup4** – slouží k parsování HTML kódu stránky,
- **lxml** – parser použitý knihovnou BeautifulSoup pro efektivní zpracování HTML,
- **sqlalchemy** – zajišťuje komunikaci s databází PostgreSQL,
- **psycopg2** nebo **psycopg2-binary** – databázový ovladač pro PostgreSQL používaný přes SQLAlchemy.

Vedle Pythonu je nutné mít nainstalovaný databázový systém PostgreSQL s rozšířením PostGIS. Skript se při inicializaci pokouší automaticky vytvořit rozšíření postgis, ale to je možné pouze v případě, že je PostGIS v systému již nainstalován.

Dále je nutné být připojen k internetu, protože skript komunikuje se dvěma externími službami. První z nich je samotný web Bezrealitky.cz, odkud jsou stahována data o inzerátech. Druhou službou je Nominatim z projektu OpenStreetMap, která slouží ke geokódování adres.

Databázové připojení je ve skriptu řešeno pomocí připojovacího řetězce uloženého v proměnné DB_URI, viz Obr. 3. Před spuštěním je nutné změnit jméno a heslo pro přihlášení do databáze, případně adresu serveru nebo port.

```
# Připojovací řetězec k databázi PostgreSQL/PostGIS.  
# Nutné změnit jméno a heslo, případně port a adresu serveru  
DB_URI = os.getenv(  
    "DB_URI",  
    "postgresql+psycopg2://jméno:heslo@localhost:5432/bezrealitky"  
)
```

Obr. 3 Připojení do databáze

Na počítači musí existovat databáze s názvem bezrealitky. Databázové spojení je následně vytvořeno pomocí funkce `create_engine()` z knihovny `SQLAlchemy`.

Pro agregaci dat je nezbytné mít v databázi vrstvu krajů a okresů. V této práci jsou využita data z ArcČR 500 v. 3.3. Je také nutné vytvořit šest materialized views podle pro všechny kombinace prodeje a pronájmu bytů a domů viz Obr. 5.

Základní princip fungování skriptu

Nejprve se na základě vstupních parametrů určí, jaký typ inzerátů má být zpracován. Následně se sestaví URL výpisu inzerátů na portálu Bezrealitky.cz. Poté skript postupně prochází jednotlivé stránky výpisu, z každé stránky extrahuje dostupné inzeráty, získává jejich atributy, validuje je, případně geokóduje a ukládá výsledná data do databáze. Po ukončení sběru dat skript aktualizuje vybrané materialized views v databázi.

Konfigurační část skriptu

Na začátku skriptu jsou definovány základní konfigurační konstanty. Proměnná `BASE` obsahuje základní URL adresu portálu Bezrealitky.cz. Dále je definován User-Agent, který identifikuje klienta přistupujícího k serveru. Nastavena je také maximální doba čekání na odpověď (`REQUEST_TIMEOUT`) a prodleva mezi načítáním jednotlivých stránek (`DELAY_BETWEEN_PAGES`). Tato prodleva omezuje příliš rychlé opakované požadavky na server a snižuje riziko blokace nebo přetěžování služby.

Samostatně je definována konfigurace geokódování. Proměnná `NOMINATIM_URL` obsahuje adresu služby Nominatim a `GEOCODE_CACHE_FILE` určuje cestu k JSON souboru, ve kterém se ukládají výsledky geokódování. Díky této cache není nutné opakovaně dotazovat stejnou adresu, což snižuje počet volání externí služby a urychluje opakované spuštění skriptu.

Režimy spuštění

Velmi důležitou součástí skriptu je dictionary `MODE_MAP`. Ten propojuje kombinaci dvou vstupních parametrů, tedy typu transakce a typu nemovitosti, s konkrétní cestou na webu a názvem databázové tabulky. Například kombinace ("pronajem", "byt") odpovídá URL cestě `/vypis/nabidka-pronajem/byt` a data se ukládají do tabulky `pronajem_byt`. Stejným způsobem jsou nastaveny i další kombinace pro domy a prodej.

Generování databázové struktury

Skript obsahuje funkci `build_ddl(table)`, která generuje SQL příkazy pro vytvoření cílové tabulky. Tabulka obsahuje následující sloupce: id inzerátu, adresu, cenu, měnu, výměru, URL adresu, číslo stránky výpisu, geometrii, čas vložení do databáze a čas poslední aktualizace prvku. Prostorová data jsou ukládána do sloupce `geom` jako `geometry(Point, 4326)`, tedy bodová geometrie ve WGS 84. Současně se vytváří prostorový index typu `GIST` nad sloupcem `geom` a index nad sloupcem `updated_at`, který zrychluje databázové dotazy.

Druhá důležitá funkce, `build_upsert(table)`, generuje SQL dotaz typu `INSERT ... ON CONFLICT DO UPDATE`. Umožňuje vložení nového záznamu, nebo aktualizaci již existujícího záznamu se stejným primárním klíčem. V praxi to znamená, že pokud se

inzerát ve objeví znovu, neuloží se duplicitně, ale aktualizují se jeho atributy a čas poslední změny. Pro geometrii jsou použity funkce `ST_MakePoint()` a `ST_SetSRID()`, které z dvojice souřadnic vytvoří bodový prvek.

Získávání souřadnic a geokódování

Získání prostorové informace o poloze nemovitosti je řešeno dvoustupňovým způsobem. Primárně se skript snaží využít souřadnice přímo dostupné ve zdrojových datech. Tyto souřadnice jsou součástí JSON struktury (konkrétně v objektu `gps`) a obsahují hodnoty zeměpisné šířky (`lat`) a délky (`lng`). Pokud jsou tyto hodnoty k dispozici, jsou použity přímo bez dalšího zpracování.

V případě, že souřadnice ve zdrojových datech chybí nebo nejsou validní, skript využívá jako záložní mechanismus geokódování textové adresy pomocí služby Nominatim OSM. Geokódování je aplikováno ve třídě `NominatimGeocoder`, která zajišťuje komunikaci se službou a správu lokální cache.

Při inicializaci této třídy je zadán kontaktní e-mail a minimální časová prodleva mezi jednotlivými požadavky. Právě uvedení e-mailu nastavení prodlevy na jednu sekundu odpovídá doporučením pro využívání Nominatim geokódovacích služeb. Třída si zároveň ukládá čas posledního dotazu, aby bylo možné zajistit dodržení této prodlevy.

Samotná metoda `geocode()` nejprve ověří, zda vstupní adresa není prázdná, a následně kontroluje, zda již není uložena v cache. Pokud se adresa v cache nachází, je vrácen uložený výsledek bez nutnosti opakovaného dotazu na externí službu. Pokud adresa neobsahuje označení státu, skript k ní automaticky doplní text „Česko“. Cílem je získat data pouze pro Českou republiku a zvyšuje se tím přesnost geokódování.

Následně je odeslán HTTP požadavek na API Nominatim, kde jsou specifikovány parametry jako formát odpovědi (`jsonv2`), omezení počtu výsledků na jeden a filtrování podle státu (`countrycodes=cz`). Pokud je geokódování úspěšné, získané souřadnice jsou uloženy do cache a vráceny zpět do procesu skriptu. V opačném případě je do cache uložena hodnota `None`, aby se zabránilo opakovanému dotazování na neúspěšné adresy.

Stahování a zpracování jednotlivých stránek

Pro stažení jedné stránky výpisu slouží funkce `fetch_page()`. Ta sestaví URL adresu, případně doplní parametr `page` pro další stránky stránkovaného výpisu. Poté odešle HTTP požadavek a z přijatého HTML dokumentu pomocí `BeautifulSoup` vyhledá element `__NEXT_DATA__`, který obsahuje strukturovaná data ve formátu JSON. Výsledkem funkce je slovník obsahující naparsovaná data, nebo hodnota `None`, pokud data nelze získat.

Vyhledání inzerátů v datové struktuře

Funkce `find_advert_refs()` prohledává JSON strukturu a snaží se nalézt část obsahující seznam inzerátů. Konkrétně se orientuje na objekt `apolloCache`. Ve větvi `ROOT_QUERY` hledá klíč začínající textem `listAdverts`, který reprezentuje výsledek dotazu na seznam inzerátů. Pokud je takový klíč nalezen, funkce vrátí seznam referencí na jednotlivé inzeráty a celý objekt cache, ze kterého lze později získat detailní informace o každém záznamu.

Extrakce atributů inzerátu

Pro každý nalezený inzerát skript získává jeho id, adresu, cenu, měnu, výměru a relativní URI detailu. Všechny tyto atributy jsou dostupné v JSON, viz Obr. 4. Pomocná funkce `get_address_from_advert()` řeší skutečnost, že adresa může být v datech uložena pod jazykovými variantami, například v lokalizované podobě `address({"locale":"CS"})`. Funkce `to_int_or_none()` převádí vybrané atributy na celočíselný datový typ a v případě neplatných nebo chybějících hodnot vrací `None`. Tím se předchází chybám při ukládání do databáze.

```
"Advert:769121": {
  "__typename": "Advert",
  "id": "769121",
  "uri": "769121-nabidka-pronajem-bytu-stredni-brno",
  "estateType": "BYT",
  "offerType": "PRONAJEM",
  "disposition": "GARSONIERA",
  "landType": "UNDEFINED",
  "imageAltText({"locale":"CS"})": "Pronájem bytu Garsoniéra 32 m², Střední, Brno",
  "mainImage": { ...
},
  "publicImages({"limit":3})": [ ...
],
  "address({"locale":"CS"})": "Střední, Brno - Ponava, Jihomoravský kraj",
  "surface": 32,
  "surfaceLand": null,
  "tags({"locale":"CS"})": [ ...
],
  "price": 13500,
  "charges": 2650,
  "currency": "CZK",
  "petFriendly": false,
  "reserved": false,
  "highlighted": false,
  "roommate": false,
  "project": null,
  "gps": {
    "__typename": "GPSPoint",
    "lat": 49.2170531,
    "lng": 16.6027547
  },
}
```

Obr. 4 Náhled JSON struktury vybraného inzerátu

Součástí zpracování je i validace dat. Skript ponechává pouze inzeráty v českých korunách (CZK), aby se vyfiltrovaly pouze nemovitosti na území České republiky. Dále vyžaduje, aby užitná plocha byla uvedena a měla kladnou hodnotu. Inzeráty, které těmto podmínkám nevyhovují, jsou přeskočeny a tato skutečnost je zaznamenána do konzole. Je to z toho důvodu, že při zachování nemovitostí s nulovou plochou, nastaly problémy při následné agregaci a přepočtu na ceny na m².

Hlavní scrapovací smyčka

Klíčová logika skriptu je implementována ve funkci `scrape_all_pages()`. Tato funkce postupně prochází všechny dostupné stránky výpisu. Pro každou stránku nejprve stáhne data, následně v nich vyhledá odkazy na inzeráty a zpracuje každý dosud neviděný záznam. Skript si vede množinu `seen_ids`, která přeskakuje duplicitní inzeráty. Tím je zajištěno, že tentýž inzerát nebude během jednoho běhu zpracován vícekrát.

Zajištěno je také omezení nekonečného procházení inzerátů. Skript sleduje několik stavových ukazatelů. Pokud se dvakrát po sobě nepodaří načíst data nebo stránka neobsahuje žádné inzeráty, je procházení ukončeno. Dále se porovnává tzv. signatura

stránky, tedy posloupnost referencí na inzeráty. Pokud více stránek po sobě vrací totožný obsah, skript předpokládá, že bylo dosaženo konce stránkování. Posledním ochranným mechanismem je ukončení běhu po několika stránkách bez nových inzerátů.

Po zpracování inzerátů na jedné stránce skript vytváří batch záznamů, který je následně uložen do databáze jedním hromadným dotazem. Po dokončení každé stránky skript vypisuje do konzole informaci o počtu nově nalezených a uložených záznamů.

Aktualizace materialized views

Po ukončení sběru dat se volá funkce `refresh_materialized_views()`. Ta na základě kombinace transakce a typu nemovitosti vybere odpovídající materialized views a provede nad nimi příkaz `REFRESH MATERIALIZED VIEW`. Tím se zajistí, že agregace podle krajů a okresů budou obsahovat aktuální data.

Zpracování parametrů příkazové řádky

Skript je navržen pro spuštění z příkazové řádky. Povinnými parametry jsou `--typ`, `--transakce` a `--contact`. Parametr `--typ` určuje, zda se mají zpracovávat byty nebo domy. Parametr `--transakce` určuje, zda jde o pronájem nebo prodej. Parametr `--contact` obsahuje kontaktní e-mail použitý při geokódování. Volitelným parametrem je `--geocode-delay`, který umožňuje nastavit prodlevu mezi dotazy na službu Nominatim. Výchozí hodnota je jedna sekunda.

Spuštění může vypadat například takto:

```
python bezrealitky.py --typ byt --transakce pronajem --contact  
jmeno@email.xy
```

Tvorba agregovaných dat

Pro následnou analýzu jsou bodová data inzerátů prostorově propojena s polygonovými vrstvami administrativních jednotek (kraje a okresy). K tomuto účelu je využita funkce `ST_Within`, která testuje, zda se bod reprezentující nemovitost nachází uvnitř daného polygonu.

Na základě tohoto prostorového propojení jsou v databázi vytvářeny materialized views, které obsahují přepočítanou cenu na m². Příkaz k vytvoření je na Obr. 5.

```
SELECT k.kod_kraj,  
       st_transform(k.geom, 4326) AS geom,  
       k.naz_cznuts,  
       avg(b.cena / b.vymena_m2) AS cena_m2,  
       count(b.id) AS pocet_bytu  
FROM   kraje k  
       LEFT JOIN prodej_byt b ON st_within(b.geom, st_transform(k.geom, 4326))  
GROUP BY k.kod_kraj, k.geom, k.naz_cznuts;
```

Obr. 5 Ukázka SQL kódu na vytvoření materialized view

Publikace dat pomocí GeoServeru

Výsledná agregovaná data jsou publikována prostřednictvím serveru GeoServer, který umožňuje jejich zpřístupnění pomocí standardizovaných webových služeb.

Databázové tabulky a materialized views jsou v GeoServeru připojeny jako datové zdroje (PostGIS datastore). Na jejich základě jsou vytvořeny vrstvy, které reprezentují prostorová data krajů a okresů doplněná o vypočtené statistiky. Pro publikaci dat je využita služba WFS (Web Feature Service), která umožňuje přístup k vektorovým datům včetně jejich atributů.

Tvorba webové mapy

Poslední částí řešení je tvorba webové mapy, která slouží k vizualizaci zpracovaných dat o cenách nemovitostí. Ta je vytvořena pomocí knihovny Leaflet.

Základ aplikace tvoří HTML struktura, ve které je definován mapový kontejner a doplňující prvky, jako je název mapy nebo popis. Samotná mapa je inicializována v JavaScriptu, kde je nastaven výchozí pohled na území České republiky a přidány základní kompoziční prvky: mapové pole, měřítko, legenda, tiráž a název mapy. Stylování aplikace je řešeno pomocí CSS, které upravuje vzhled mapy i interaktivních prvků.

Do mapy jsou načítána prostorová data ve formátu GeoJSON, která pocházejí z publikovaných WFS vrstev v prostředí GeoServer. Tato data obsahují již agregovanou hodnotu ceny za m², která je vizualizována formou kartogramu. Uživatel může přepínat mezi různými typy dat (např. byty, domy) a úrovněmi administrativního členění (kraje, okresy).

Mapa dále obsahuje interaktivní prvky, jako jsou vyskakovací okna s informacemi o vybrané oblasti (průměrnou cenu a počet nemovitostí) nebo dynamicky generovaná legenda.

Webovou mapu je možné spustit lokálně po stažení všech souborů (HTML, CSS, JavaScript). V případě přímého napojení na WFS služby GeoServeru je nutné řešit omezení typu CORS (Cross-Origin Resource Sharing), které vzniká při komunikaci mezi různými porty (např. localhost:8000 a localhost:8080). Z tohoto důvodu je doporučeno mapu vložit v adresářové struktuře přímo v rámci GeoServeru a spustit na adrese např.: <http://localhost:8080/geoserver/www/index.html>. Tím je zajištěno, že webová aplikace i datové služby běží ve stejném prostředí a nedochází k blokování požadavků ze strany webového prohlížeče.

Součástí práce je také odkaz na vizuálně totožnou webovou mapu, který je součástí webových stránek práce. Pro účely této prezentace jsou zde využita data uložená ve formátu GeoJSON, která jsou přímo nahraná v aplikaci, nikoli dynamicky načítaná z databáze či WFS služby. Z tohoto důvodu se jedná o statickou ukázkou, kde zobrazená data nemusí odpovídat aktuálnímu stavu databáze.

4.3 ČT 24

Druhý skript je určený pro získávání článků ze zpravodajského portálu ČT24, extrakci jejich základních atributů, identifikaci hlavních geografických lokalit zmíněných v textu a převod těchto lokalit na prostorová data. Uživatel při spuštění zadává časový rozsah, volí poskytovatele jazykového modelu a doplňuje příslušný API klíč. Výstupem skriptu je datová vrstva ve formátu GeoJSON obsahující geokódované lokality přiřazené k jednotlivým článkům.

Softwarové a datové předpoklady

Skript je opět napsaný v Pythonu a využívá standardní knihovny i externí balíčky. Ze standardních knihoven jsou použity moduly `os`, `re`, `json`, `time`, `urllib.request`, `urllib.parse`, `xml.etree.ElementTree`, `datetime` a `typing`. Ty zajišťují práci se soubory, regulárními výrazy, časem, HTTP komunikací a zpracováním XML dat.

Pro správnou funkčnost skriptu je nutné doinstalovat následující externí balíčky:

- **requests** – slouží k odesílání HTTP požadavků na webový server a geokódovací službu,
- **selenium** – umožňuje automatizované načítání a zpracování obsahu článků v prostředí webového prohlížeče,
- **webdriver-manager** – zajišťuje automatickou správu odpovídající verze ovladače pro Google Chrome,
- **python-dotenv** – slouží k načítání proměnných prostředí ze souboru `.env`,
- **openai** – používá se v případě volby modelu ChatGPT.

Kromě Pythonu je nutné mít v systému nainstalovaný webový prohlížeč Google Chrome, protože Selenium při běhu skriptu vytváří instanci prohlížeče v headless režimu.

Dále je vyžadováno internetové připojení. Skript komunikuje s několika externími zdroji:

- sitemapou portálu ČT24, odkud získává seznam článků,
- samotnými stránkami jednotlivých článků,
- vybranou AI službou,
- geokódovací službou Nominatim.

Základní princip fungování skriptu

Po spuštění skriptu uživatel zadá časový rozsah a zvolí AI poskytovatele. Skript následně načte sitemap index portálu ČT24 a získá seznam URL článků. Ty jsou dále filtrovány podle data.

V další fázi jsou jednotlivé články načítány pomocí Selenium, z jejich obsahu jsou extrahovány základní atributy (titulek, text, abstrakt, datum publikace a kategorie). Následně je text článku předán jazykovému modelu, který identifikuje hlavní geografické lokality. Tyto lokality jsou následně geokódovány a uloženy do výstupního souboru GeoJSON.

Konfigurační část skriptu

Na začátku skriptu jsou definovány základní konfigurační konstanty. Proměnná `SITEMAP_INDEX_URL` obsahuje adresu sitemap indexu článků. Dále jsou nastaveny HTTP hlavičky (User-Agent, Accept-Language, Accept), které zajišťují korektní komunikaci se serverem.

Skript dále definuje časové limity:

- `SELENIUM_TIMEOUT`,
- `AFTER_NAV_SLEEP`,
- `ARTICLE_PAUSE`,
- `TIMEOUT`.

Součástí konfigurace je také dictionary `CATEGORY_MAP`, který převádí segment URL adresy na jednotný název kategorie článku.

Volba AI poskytovatele a zadání API klíče

Skript umožňuje volbu jazykového modelu. Funkce `prompt_ai_settings()` nabídne uživateli výběr mezi třemi poskytovateli: ChatGPT, Gemini a Claude. Následně uživatel zadá API klíč, který je použit pro autentizaci při volání příslušné služby.

Získávání seznamu článků pomocí sitemap

Skript využívá XML sitemapu portálu ČT24. Funkce `fetch_sitemap_index_urls()` načte index sitemap souborů a získá jejich adresy. Následně funkce `fetch_article_candidates_from_sitemap()` prochází jednotlivé sitemap dokumenty a vybírá URL odpovídající článkům.

Práce s datem a filtrování článků

Skript umožňuje zadat počáteční a koncové datum ve formátu YYYY-MM-DD. K tomu slouží funkce `prompt_date_range()`, která načte vstup od uživatele a převede jej na objekt typu `date` pomocí pomocné funkce `parse_user_date()`. V případě, že uživatel zadá neplatný formát nebo je počáteční datum větší než koncové, skript vyvolá chybu a běh se ukončí. Filtrace článků probíhá ve dvou na sebe navazujících krocích. Nejprve se využívá údaj `lastmod` získaný ze sitemap, který představuje datum poslední změny článku. Ve druhém kroku je po otevření konkrétního článku zjištěno skutečné datum publikace. To je získáváno z JSON-LD struktury vložené v HTML dokumentu, konkrétně z atributu `datePublished`.

Načítání článků pomocí Selenium

Pro načtení jednotlivých článků skript využívá knihovnu Selenium. Funkce `build_driver()` vytváří instanci prohlížeče Google Chrome v headless režimu, tedy bez grafického rozhraní. Funkce `wait_body()` čeká, dokud se v dokumentu neobjeví element `<body>`, čímž se minimalizuje riziko, že parsování začne ještě před úplným načtením stránky. Samotné zpracování článku probíhá ve funkci `scrape_article()`. Ta z otevřené stránky získává titulek, text článku, abstrakt, kategorii a datum publikace. Text článku je sestavován z odstavců nalezených pomocí sady CSS selektorů uložených v proměnné `PARA_SELECTORS`.

V případě, že se nepodaří získat dostatečně dlouhý text (například kvůli odlišné struktuře stránky), skript využívá jako záložní řešení obsah meta tagu description.

Extrakce kategorie

Kategorie článku je určována funkcí `extract_category()`, která ji odvozuje z URL adresy. URL struktura portálu ČT24 obsahuje tematickou kategorii jako jednu z částí adresy, například:

- [https://ct24.ceskatelevize.cz/clanek/**svet**/komise-zaluje-cesko-a-madarsko-kvuli-evropskemu-zatykacimu-rozkazu-372934](https://ct24.ceskatelevize.cz/clanek/svet/komise-zaluje-cesko-a-madarsko-kvuli-evropskemu-zatykacimu-rozkazu-372934),
- [https://ct24.ceskatelevize.cz/clanek/**ekonomika**/uohs-zastavil-spravni-rizeni-s-energetickymi-firmami-zakazane-dohody-neprokazal-372938](https://ct24.ceskatelevize.cz/clanek/ekonomika/uohs-zastavil-spravni-rizeni-s-energetickymi-firmami-zakazane-dohody-neprokazal-372938),
- [https://ct24.ceskatelevize.cz/clanek/**kultura**/muzum-se-spatne-vyjadruji-emoce-rika-tvurce-serialu-o-toxicke-maskulinite-372927](https://ct24.ceskatelevize.cz/clanek/kultura/muzum-se-spatne-vyjadruji-emoce-rika-tvurce-serialu-o-toxicke-maskulinite-372927).

Extrakce geografických lokalit pomocí jazykových modelů

Klíčovou částí skriptu je extrakce geografických lokalit z textu článků. Ta je v tomto případě řešena pomocí jazykových modelů, které z textu identifikují hlavní místa, o nichž článek skutečně pojednává.

Pro přípravu vstupu modelu slouží funkce `build_location_prompt()`, viz Obr. 6, která vytváří instrukce v českém jazyce. Model by měl vrátit pouze hlavní lokality, nikoli všechny zmíněné názvy. Lokality nejsou vráceny pouze jako seznam řetězců, ale jako strukturované objekty. Každá lokalita obsahuje:

- název,
- typ (např. city, country, natural_area, region apod.),
- případně nadřazené jednotky (např. stát nebo region).

Tato struktura je snahou pro přesnější následné geokódování.

```
"Z textu článku extrahuj pouze hlavní geografické lokality, o kterých článek skutečně je. "
"Nevracej všechny jen okrajově zmíněné lokality.\n\n"
"Vrať výhradně JSON pole objektů. Nevracej žádný jiný text, žádné vysvětlení, žádný markdown a žádné bloky ```json.\n\n"
"Každý objekt musí obsahovat:\n"
'- "name": název lokality v základním tvaru vhodném pro geokódování\n'
'- "type": typ lokality\n\n'
'Možné hodnoty "type":\n'
'- "street_or_square"\n'
'- "municipality"\n'
'- "district"\n'
'- "region"\n'
'- "country"\n'
'- "natural_area"\n'
'- "historical_or_broad_area"\n'
'- "local_place"\n'
'- "other"\n\n'
"Dále může objekt volitelně obsahovat:\n"
'- "municipality"\n'
'- "region"\n'
'- "country"\n\n'
"Pravidla:\n"
"- Vrať jen hlavní dějiště článku nebo hlavní místa, která jsou pro článek podstatná.\n"
"- Pokud je uvedena podrobnější lokalita, nevracej zároveň její nadřazenou jednotku. Například při výskytu Brno nevracej zároveň Česko.\n"
"- 'natural_area' použij pro přírodní oblasti, pohoří, chráněná území, parky a podobná území. U nich doplň stát nebo region.\n"
"- 'historical_or_broad_area' použij pro širší nebo historické oblasti, které nejsou běžnou správní jednotkou. U nich doplň stát nebo region.\n"
"- 'local_place' použij pro lokální názvy menších míst, která sama o sobě nemusí být jednoznačná. U nich doplň obec, region nebo stát podle potřeby.\n"
"- 'other' použij jen výjimečně, pokud lokalitu nelze smysluplně zařadit.\n"
"- Ulice a náměstí vrať v základním tvaru a doplň obec, pokud je z textu zřejmé.\n"
"- Obce a města vrať v základním tvaru. Kraj nebo stát přidej jen pokud je to nutné pro jednoznačnost, jinak je nepřidávej.\n"
"- Názvy lokalit vrať v základním tvaru vhodném pro geokódování, ne ve skloňovaném tvaru. Použij například 'Jižní Korea' místo 'Jižní Koreu'.\n"
"- Rozlišuj stejné názvy podle kontextu článku.\n"
"- Nevymýšlej informace, které nejsou v textu nebo jasně odvoditelné.\n"
"- Nevracej duplicity.\n\n"
f"Title: {title}\n"
f"Text:\n{snippet}"
```

Obr. 6 Prompt zadání AI modelu

Skript podporuje tři samostatné varianty:

- `gpt_extract_locations()` pro model ChatGPT,
- `gemini_extract_locations()` pro Google Gemini,
- `claude_extract_locations()` pro Anthropic Claude.

Nad těmito funkcemi stojí sjednocující funkce `extract_locations()`, která podle volby uživatele zvolí příslušného poskytovatele.

Zpracování odpovědi AI modelu

Odpověď jazykového modelu nemusí být vždy vrácena ve zcela čisté podobě. V praxi se ukázalo, že modely ChatGPT a Claude sice vracejí požadovanou JSON strukturu, ale často ji doplňují o formátovací prvky, například markdown bloky typu ``json. Z tohoto důvodu skript obsahuje pomocnou funkci `normalize_json_block()`, která tyto prvky odstraní a převede výstup na čistý text obsahující JSON.

Následně se tento text zpracuje pomocí funkce `json.loads()`, která jej převede na datovou strukturu Pythonu. Výsledkem je list objektů reprezentujících jednotlivé lokality. Pokud převod selže nebo výstup neodpovídá očekávanému formátu, je vrácen prázdný seznam. Součástí této fáze je také deduplikace lokalit a kontrola datových typů.

Geokódování lokalit

Po extrakci lokalit následuje jejich převod na zeměpisné souřadnice. Geokódování zajišťuje stejně jako v předchozím skriptu služba Nominatim.

Funkce `build_geocode_query()` sestavuje dotaz podle typu lokality. Například pro města může být dotaz rozšířen o stát, zatímco pro přírodní oblasti je použita obecnější formulace. Funkce `geocode_location()` odesílá HTTP požadavek na API služby Nominatim a z odpovědi získá souřadnice ve formě zeměpisné šířky a délky. Skript během jednoho běhu využívá cache, která ukládá již geokódované lokality.

Hlavní scrapovací smyčka

Hlavní logika programu je implementována ve funkci `main()`. Ta nejprve načte uživatelské parametry, následně získá seznam kandidátních článků ze sitemap a odstraní duplicity pomocí množiny `seen`.

Skript poté postupně načítá jednotlivé články a provádí jejich zpracování. U každého článku se kontroluje:

- zda má platné datum publikace,
- zda spadá do požadovaného intervalu,
- zda obsahuje dostatečně dlouhý text.

Články, které těmto podmínkám nevyhovují, jsou přeskočeny a tato skutečnost je zaznamenána pomocí počítadla. U validních článků je následně provedena extrakce lokalit pomocí zvoleného AI modelu. Výsledky jsou ukládány do seznamu `accepted_rows`, který slouží jako základ pro tvorbu výstupu.

Tvorba GeoJSON výstupu

Po dokončení zpracování skript vytváří výstup ve formátu GeoJSON. Geometrie má typ Point. Pokud se lokalitu nepodaří geokódovat, je geometrie nastavena na hodnotu null a atribut geocoded indikuje neúspěšné geokódování.

Atributová část obsahuje:

- titulek článku,
- URL adresu,
- kategorii,
- datum publikace,
- abstrakt,
- název lokality,
- typ lokality,
- případně další doplňující informace (např. stát nebo region).

Výsledná data jsou uložena jako FeatureCollection do souboru, jehož název odpovídá zvolenému AI poskytovateli, například *ct24_locations_chatgpt.geojson*.

Vizualizace výsledků pomocí mapové aplikace

Nad výše popsaným skriptem byla vytvořena jednoduchá webová aplikace. Ta umožňuje interaktivní zobrazení geokódovaných lokalit článků v mapě a představuje nadstavbu nad původním skriptem, který generuje data ve formátu GeoJSON.

Aplikace je implementována v jazyce Python s využitím frameworku Flask, který zajišťuje komunikaci mezi backendem a frontendem. Backendová část aplikace obsahuje funkci `scrape_range()`, která obsahuje celý proces popsaný výše. Oproti samostatnému skriptu zde není nutné zadávání vstupu přes příkazovou řádku, ale vstupní parametry jsou předávány prostřednictvím HTTP požadavku.

Frontend aplikace je realizován jako jednoduchá HTML stránka s využitím knihovny Leaflet. Mapa využívá podkladová data z OSM a umožňuje základní navigaci, jako je zoom a posun po mapě. Pro efektivní práci s větším množstvím bodů je použita knihovna Leaflet.markercluster, která seskupuje blízké body do shluků a tím zlepšuje přehlednost mapy.

Uživatel zadává vstupní parametry prostřednictvím formuláře umístěného v postranním panelu. Jedná se o stejné parametry jako ve skriptu. Po odeslání formuláře je na backend odeslán POST požadavek na endpoint `/api/scrape`, který spustí celý proces zpracování dat. Výsledný GeoJSON je následně vrácen zpět do frontendové části, kde je dynamicky vykreslen do mapy.

Každá lokalita je zobrazena jako bodový prvek, jehož barva odpovídá kategorii článku. Barevná škála je definována ve slovníku PALETTE a umožňuje rozlišení mezi tematickými skupinami článků. Při kliknutí na konkrétní bod se otevře popup okno, které obsahuje základní atributy článku, jako je titulek, datum publikace, kategorie, abstrakt a odkaz

na původní článek. Součástí aplikace je také interaktivní filtrování kategorií. Aplikace dále poskytuje základní zpětnou vazbu o průběhu zpracování dat. Uživatel může zjistit počet zpracovaných článků a počet zobrazených lokalit. Tato informace je zobrazena v dolní části sidebaru.

Pro spuštění aplikace je nutné mít nainstalovanou knihovnu Flask, Selenium všechny další knihovny potřebné pro chod skriptu uvedené v úvodu kapitoly.

Aplikace se spouští z příkazové řádky příkazem:

```
python ct24app.py
```

Po spuštění je lokálně vytvořen webový server, který běží na adrese:

```
http://127.0.0.1:8000
```

Tuto adresu je následně nutné otevřít v internetovém prohlížeči.

4.4 Deutsche Welle (RSS)

Třetí skript pro automatizované získávání článků využívá RSS kanál zpravodajského portálu Deutsche Welle. Na rozdíl od předchozího skriptu není k lokalizaci článků využíván jazykový model, ale metoda rozpoznávání pojmenovaných entit (Named Entity Recognition, NER) implementovaná v knihovně spaCy. Výstupem skriptu je GeoJSON, kde je každému článku přiřazena jedna nejpravděpodobnější geografická lokalita.

Softwarové a datové předpoklady

Skript je implementován opět v Pythonu a využívá kombinaci standardních a externích knihoven. Ze standardních knihoven jsou použity moduly argparse, json, time, dataclasses a typing.

Pro správnou funkčnost skriptu je nutné doinstalovat následující externí balíčky:

- **feedparser** – slouží ke zpracování RSS feedu a načítání jednotlivých položek,
- **requests** – zajišťuje komunikaci s geokódovací službou Nominatim,
- **spaCy** – poskytuje nástroje pro zpracování přirozeného jazyka a extrakci pojmenovaných entit.

Pro úlohu rozpoznávání geografických entit je využit předtrénovaný jazykový model knihovny spaCy, konkrétně model en_core_web_sm. Jedná se o statistický model založený na neuronových sítích, který je trénován na anglických textech a umožňuje provádět úlohy jako tokenizace, lemmatizace, part-of-speech tagging a zejména, pro tuto práci důležité, rozpoznávání pojmenovaných entit (NER). V rámci tohoto skriptu jsou využívány entity typu GPE (Geo-Political Entity) a LOC (Location), které reprezentují geograficky relevantní objekty.

Skript dále vyžaduje internetové připojení, protože komunikuje se dvěma externími zdroji:

- RSS kanálem Deutsche Welle, odkud získává seznam článků,
- geokódovací službou Nominatim.

Základní princip fungování skriptu

Zpracování dat probíhá ve čtyřech hlavních krocích:

- načtení položek z RSS feedu,
- extrakce geografických entit z titulku a popisu článku pomocí NER,
- geokódování kandidátních lokalit,
- vytvoření výstupu ve formátu GeoJSON.

Na rozdíl od skriptu pro ČT24, kde skript identifikoval více lokalit, tento vybírá pouze jednu reprezentativní lokalitu, která nejlépe vystihuje geografický kontext článku. Důvodem je využití přístupu založeného na rozpoznávání pojmenovaných entit. Na rozdíl od jazykových modelů použitých ve skriptu pro ČT24 není tento přístup schopen spolehlivě rozlišit význam jednotlivých lokalit v širším kontextu textu. Proto je zvolena strategie výběru jedné nejpravděpodobnější lokality na základě kombinace pravidel (určení váhy podle výskytu v titulku či popisu, preference víceslovných názvů apod.). Při zanechání všech lokalit obsahoval výsledek velké množství šumu. Model označil za lokality i např. jména osob či názvy firem.

Konfigurační část skriptu

Na začátku skriptu jsou definovány základní konstanty a parametry. Patří mezi ně zejména množina NER tříd LABELS, která určuje, jaké typy entit jsou považovány za geograficky relevantní (konkrétně GPE a LOC). Dále je definována adresa geokódovací služby Nominatim.

Načtení a zpracování RSS feedu

Skript přijímá vstupní parametry prostřednictvím příkazové řádky, které jsou zpracovány pomocí modulu argparse. Mezi tyto parametry patří:

- URL adresa RSS feedu,
- cesta k výstupnímu souboru,
- maximální počet zpracovaných položek,
- název jazykového modelu spaCy,
- kontaktní e-mail pro identifikaci při komunikaci s Nominatim,
- časová prodleva mezi geokódovacími dotazy,
- maximální počet kandidátních lokalit na článek.

Povinné jsou pouze URL adresa feedu, cesta k výstupnímu souboru a kontaktní email pro geokódování.

RSS feed je načítán pomocí knihovny feedparser, která umožňuje parsování XML struktury RSS kanálu. Výsledkem je sada položek (entries), kde každá položka reprezentuje jeden článek publikovaný na portálu Deutsche Welle. Pro omezení rozsahu zpracování je seznam položek redukován pomocí parametru limit, který určuje maximální počet článků zahrnutých do analýzy.

Z každé položky RSS feedu nejsou využívána všechna dostupná metadata, ale pouze vybraná pole, která jsou relevantní pro další zpracování. Tato extrakce je prováděna funkcí `extract_entry_fields()`, která z každé položky získává:

- titulek článku (title),
- URL adresu (link),
- textový popis (description),
- datum publikace (published, případně dc_date),
- id obsahu (dwsyn_contentid).

Důležité jsou především pole title a description, která slouží jako vstup pro následnou extrakci geografických entit. Jelikož RSS feed neobsahuje plný text článku, je nutné pracovat právě s těmito zkrácenými textovými reprezentacemi. Nicméně většina geograficky lokalizovaných článků už v těchto částech obsahuje název lokality, ke které se vztahuje.

```
<item rdf:about="https://www.dw.com/en/japan-raging-wildfires-cause-mass-evacuations/a-76930771?maca=en-rss-en-all-1573-rdf">
  <title>Japan: Raging wildfires cause mass evacuations</title>
  <link>https://www.dw.com/en/japan-raging-wildfires-cause-mass-evacuations/a-76930771?maca=en-rss-en-all-1573-rdf</link>
  <description>More than 1,000 firefighters are battling two blazes in northern Japan, with some 3,000 people urged to leave their homes.
</description>
  <dc:date>2026-04-25T07:50:00Z</dc:date>
  <dc:subject>News</dc:subject>
  <dc:language>en</dc:language>
  <dwsyn:contentID>76930771</dwsyn:contentID>
</item>
```

Obr. 7 Ukázka jednoho z článků v RSS

Extrakce geografických lokalit pomocí NER

Protože samotný NER model nerozlišuje význam jednotlivých lokalit v kontextu článku, bylo nutné vytvořit pravidla, podle kterých se vybere lokalita, která nejpravděpodobněji odpovídá kontextu článku.

Pravidla jsou následující:

- **Vyšší váha lokalit v titulku**
Lokality extrahované z titulku článku jsou ohodnoceny vyšší vahou než lokality z popisu. Titulek obvykle shrnuje hlavní téma článku a obsahuje klíčovou geografickou informaci.
- **Preference víceslovných názvů**
Víceslovné názvy (např. „New York“, „South Korea“) jsou mírně zvýhodněny, protože bývají podrobnější a reprezentativnější než jednoslovné názvy.
- **Penalizace krátkých řetězců**
Velmi krátké názvy (typicky o délce několika znaků) jsou penalizovány, protože často představují zkratky nebo nejednoznačné výrazy, které nejsou vhodné pro geokódování. Před jejich odstraněním mělo příliš mnoho článků přiřazenou lokalitu: USA či EU.
- **Odstranění duplicit**
Duplicitní výskyty lokalit jsou eliminovány na základě normalizované textové podoby (např. převedení na malá písmena a odstranění interpunkce). Tím je zajištěno, že každá lokalita je v seznamu zastoupena pouze jednou.

Na základě těchto pravidel je každé lokalitě přiřazeno skóre, které reprezentuje její význam v rámci článku. Následně jsou všechny kandidátní lokality seřazeny podle tohoto skóre sestupně. Výsledkem je uspořádaný seznam kandidátních lokalit, který slouží jako vstup pro geokódování. Vzhledem k tomu, že skript dále pracuje pouze s omezeným počtem kandidátů, dochází v této fázi k redukci dat a eliminaci méně podstatných lokalit.

Geokódování lokalit

Pro převod názvů lokalit na souřadnice je implementována třída `NominatimGeocoder`, která zajišťuje komunikaci se službou Nominatim. Geokódování probíhá opakovaně nad seznamem kandidátů. Pro každý kandidátní název je odeslán dotaz na API a z odpovědi je získán nejrelevantnější výsledek.

Výběr nejvhodnější lokality

Funkce `best_geocode()` slouží k výběru jedné reprezentativní lokality z množiny kandidátů získaných pomocí NER. Proces výběru probíhá v několika navazujících krocích. Nejprve jsou kandidátní lokality zpracovávány v pořadí podle jejich relevance, která byla určena v předchozí fázi hodnocení.

Po získání geokódovaných výsledků je každý z nich klasifikován podle úrovně geografické přesnosti. Tato klasifikace vychází ze struktury adresních atributů vrácených službou Nominatim a rozlišuje několik základních úrovní:

- **sídelní jednotky** (settlement) – města, obce, vesnice,
- **administrativní celky** (admin) – kraje, regiony nebo jiné správní jednotky,
- **státy** (country),
- **případně neznámá nebo neurčitá úroveň** (unknown).

Na základě této klasifikace je definována hierarchie preferencí, která upřednostňuje lokality střední úrovně detailu. Nejvyšší prioritu mají sídelní jednotky, protože obvykle nejlépe reprezentují konkrétní místo, kde se děj článku odehrává. Administrativní celky představují obecnější kontext a jsou voleny v případě, že přesnější lokalita není k dispozici. Státy pak slouží jako nejhrubší úroveň reprezentace. Součástí procesu je také filtrace nevhodných výsledků. Z geokódovaných dat jsou odstraňovány příliš detailní objekty, jako jsou městské části, čtvrti nebo předměstí. Tyto jednotky bývají často příliš specifické a právě tyto objekty byly velmi často zaměňovány za jména jiná než geografická. Výběr výsledné lokality probíhá tak, že je postupně aktualizován nejlepší nalezený kandidát podle výše uvedeného pořadí. Jakmile je nalezena lokalita odpovídající sídelní jednotce, je proces ukončen, protože tato úroveň je považována za optimální reprezentaci.

Tvorba GeoJSON výstupu

Pro každý úspěšně zpracovaný článek je vytvořen prvek typu `Feature`, jehož geometrie má typ `Point`.

Atributová část obsahuje:

- titulek článku,
- URL odkaz,
- datum publikace,
- popis článku,
- identifikátor obsahu,
- informace o vybrané lokalitě (název, typ, přesnost, adresa).

▼ dw_en_all	
▼ place_display_name	日本
▶ (Odvozené)	
▶ (Akce)	
title	Japan: Raging wildfires cause mass evacuations
link	https://www.dw.com/en/japan-raging-wildfires-cause-mass-evacuations/a-76930771?maca=en-rss-en-all-1573-rdf
published	25.04.2026 07:50:00 (UTC)
description	More than 1,000 firefighters are battling two blazes in northern Japan, with some 3,000 people urged to leave their homes.
dw_content_id	76930771
place_query	Japan
place_display_name	日本
place_type	administrative
place_rank	4
place_precision	country
address	country: 日本, country_code: jp

Obr. 8 Ukázka atributové části vybraného článku

Výsledná data jsou uložena jako FeatureCollection do GeoJSON souboru. Součástí výstupu jsou také informace o průběhu zpracování, například počet zpracovaných položek nebo počet neúspěšných případů.

4.5 The Guardian (API)

Čtvrtý skript získává články z portálu The Guardian a k tomu využívá oficiální API přístup. Na rozdíl od skriptů pracujících s RSS kanálem nebo sitemapou zde není nutné parsovat webovou stránku ani načítat obsah pomocí prohlížeče. Nevýhodou jsou omezení vyplývající z podmínek používání, např. maximálně 500 dotazů denně. Data jsou extrahována přímo ve strukturované podobě ve formátu JSON. Skript následně z vybraných textových polí článku vytahuje geografickou lokalitu pomocí regulárních výrazů, tuto lokalitu geokóduje, stejně jako v předchozích skriptech, pomocí služby Nominatim a výsledek ukládá do GeoJSONu.

Softwarové a datové předpoklady

Skript je opět napsán Pythonu. Ze standardních knihoven jsou použity moduly os, re, time, json, argparse a typing. Tyto moduly zajišťují práci s regulárními výrazy, časovými prodlevami, serializací dat, argumenty příkazové řádky a typovými anotacemi.

Z externích knihoven je nutné doinstalovat balíček:

- requests – slouží k odesílání HTTP požadavků na Guardian API a geokódovací službu Nominatim.

Pro správné spuštění je dále nutné mít přístup k internetu a platný API klíč ke službě Guardian Content API. Tento klíč je předáván jako povinný parametr při spuštění skriptu. Kromě toho je vyžadován kontaktní e-mail, který se používá v hlavičce User-Agent při komunikaci se službou Nominatim, stejně jako v předchozích skriptech.

Skript komunikuje se dvěma externími službami:

- Guardian Content API, odkud získává články a jejich textová pole,
- Nominatim OpenStreetMap.

Základní princip fungování skriptu

Zpracování dat probíhá v několika hlavních krocích. Nejprve uživatel zadá vyhledávací dotaz, API klíč, kontaktní e-mail a případně další volitelné parametry, například rozsah dat nebo maximální počet stránek výsledků. Skript následně odešle požadavek na Guardian API a získá seznam článků odpovídajících zadanému dotazu.

Pro každý článek je následně načten detailní obsah včetně vybraných textových polí. Z těchto polí se skript pokouší extrahovat geografickou lokalitu. Pokud je lokalita nalezena, je předána geokódovací službě Nominatim. V případě úspěšného geokódování je vytvořen bodový prvek GeoJSON, který obsahuje souřadnice a základní atributy článku.

Konfigurační část skriptu

Na začátku skriptu je definována konstanta API_BASE, která obsahuje základní URL adresu Guardian Content API. Důležitou součástí konfigurace jsou také regulární výrazy určené pro extrakci lokalit. Skript využívá dva vzory, ukázka na Obr. 9. První, obecnější regulární výraz BYLINE_IN_RE, hledá fráze typu „in London“, „from Paris“ nebo „at Brussels“. Druhý, přísnější výraz INLINE_PLACE_RE, omezuje hledanou lokalitu na maximálně tři slova začínající velkým písmenem. Jde o snahu snížit riziko, že bude jako lokalita omylem zachycena příliš dlouhá nebo nevhodná část textu.

```
# Regulární výraz pro extrakci lokace z textu typu "in London", "from Paris"
BYLINE_IN_RE = re.compile(
    r"\b(?:in|at|from)\s+([A-Z][A-Za-z0-9\-\s'.,()]{1,120})"
)

# Přísnější regulární výraz - maximálně tři slova začínající velkým písmenem
INLINE_PLACE_RE = re.compile(
    r"\b(?:in|at|from)\s+([A-Z][A-Za-z\-' ]+(?:\s+[A-Z][A-Za-z\-' ]+){0,2})\b"
)
```

Obr. 9 Ukázka regulárních výrazů z kódu

Zpracování parametrů příkazové řádky

Skript je spustitelný z příkazové řádky. Povinnými parametry jsou:

- --q – vyhledávací dotaz pro Guardian API (“ “ se může použít pro všechny články),
- --api-key – API klíč ke službě Guardian,
- --contact – kontaktní e-mail použitý při geokódování pomocí Nominatim.

Volitelně lze nastavit:

- --max-pages – maximální počet stránek výsledků,
- --page-size – počet článků na jednu stránku výsledků,
- --geojson-out – název výstupního GeoJSON souboru,
- --from-date – počáteční datum vyhledávání,
- --to-date – koncové datum vyhledávání.

Spuštění může vypadat například takto:

```
python guardian_api.py --q " " --api-key API_KLIC --contact jmeno@email.xy --from-date 2024-01-01 --to-date 2024-12-31 --geojson-out guardian.geojson
```

Načítání článků z Guardian API

Hlavní část načítání probíhá ve funkci `main()`. Skript postupně prochází stránky výsledků podle parametru `page`, dokud není dosaženo maximálního počtu stránek nebo dokud API nevrátí prázdný seznam výsledků.

Při dotazu na endpoint `/search` jsou předávány parametry jako vyhledávací dotaz, číslo stránky, velikost stránky, řazení podle nejnovějších článků a seznam polí, která mají být vrácena. Skript požaduje pole `byline`, `trailText`, `bodyText` a `headline`, viz Obr. 10. Právě tato pole jsou následně využívána pro extrakci geografické lokality. Po získání seznamu článků je pro každý záznam odeslán ještě samostatný dotaz na detail článku. Díky tomu skript získá úplnější datovou reprezentaci článku, včetně všech potřebných textových polí.

```
"response": {
  "status": "ok",
  "userTier": "developer",
  "total": 1,
  "content": {
    "id": "environment/2026/apr/25/chris-bowen-iran-war-oil-shock-climate-negotiations",
    "type": "article",
    "sectionId": "environment",
    "sectionName": "Environment",
    "webPublicationDate": "2026-04-25T20:00:54Z",
    "webTitle": "Cop negotiations chief on how Iran war oil shock paves road to climate talks in Turkey",
    "webUrl": "https://www.theguardian.com/environment/2026/apr/25/chris-bowen-iran-war-oil-shock-climate-negotiations",
    "apiUrl": "https://content.guardianapis.com/environment/2026/apr/25/chris-bowen-iran-war-oil-shock-climate-negotiations",
    "fields": {
      "headline": "Cop negotiations chief on how Iran war oil shock paves road to climate talks in Turkey",
      "trailText": "<strong>Exclusive:</strong> As countries meet at key climate crisis meetings, Australia's Chris Bowen says",
      "byline": "Adam Morton Climate and environment editor",
      "bodyText": "The fallout from the Iran war is driving countries to boost homegrown energy reliability and opens an oppor",
    },
    "isHosted": false,
    "pillarId": "pillar/news",
    "pillarName": "News"
  }
}
```

Obr. 10 Ukázka vybraných polí jednoho článku

Extrakce geografické lokality

Extrakce lokalit je založena na pravidlovém přístupu pomocí regulárních výrazů. Pro extrakci jsou využity čtyři zdroje, které mají pevně stanovenou prioritu:

1. `byline`
2. `headline`
3. `trailText`
4. `bodyText`

Nejvyšší prioritu má pole `byline`, protože u zpravodajských článků často obsahuje informaci o místě, odkud reportér píše. Typickým příkladem může být text ve formátu „John Smith in London“. Pokud se lokalitu nepodaří najít v `byline`, skript pokračuje titulkem článku, krátkým úvodním textem a nakonec hlavním textem článku.

Samotná extrakce je rozdělena do několika pomocných funkcí. Funkce `extract_from_byline()` hledá lokalitu v poli autora článku. Funkce `extract_from_headline()`

pracuje s titulkem a funkce `extract_from_text()` analyzuje textová pole. U delších textů se skript omezuje pouze na první řádek a prvních 500 znaků. V drtivé většině případů se název hlavní geografické lokality u zpravodajských textů často nachází na začátku článku.

Hlavní sjednocující funkcí je `extract_location_from_article()`. Ta postupně projde všechna pole, uloží nalezené kandidáty a následně vybere první lokalitu podle stanovené priority. Výstupem této funkce je dictionary obsahující název lokality a informaci o tom, ze kterého pole byla získána.

Geokódování lokalit

Geokódování probíhá stejně jako v předchozích skriptech pomocí Nominatim. Je vytvořena třída `NominatimGeocoder` a na službu Nominatim se posílá požadavek s parametry `q`, `format = jsonv2` a `limit = 1`. Pokud služba vrátí platnou odpověď, skript vybere první výsledek a získá z něj zeměpisnou šířku, zeměpisnou délku a textový popis místa (`display_name`). Pokud odpověď není úspěšná nebo neobsahuje žádný výsledek, funkce vrací hodnotu `None` a článek není zahrnut do výstupního GeoJSON souboru.

Hlavní zpracovatelská smyčka

Hlavní smyčka je součástí funkce `main()`. Skript nejprve inicializuje geocoder a prázdný seznam `features`, do kterého se ukládají výsledné GeoJSON prvky. Následně prochází jednotlivé stránky výsledků Guardian API.

Pro každý článek je proveden tento postup:

- načtení detailu článku z Guardian API,
- extrakce lokality z dostupných textových polí,
- přeskočení článku, pokud se lokalita nepodaří najít,
- geokódování nalezené lokality,
- přeskočení článku, pokud geokódování selže,
- vytvoření GeoJSON prvku a jeho přidání do výstupního seznamu.

Tvorba GeoJSON výstupu

Po dokončení zpracování všech stránek skript vytvoří výslednou strukturu typu `FeatureCollection`. Geometrie má typ `Point`.

Atributová část obsahuje:

- ID článku,
- titulek článku,
- URL adresu článku,
- text nalezené lokality,
- zdroj, ze kterého byla lokalita získána,
- popis místa vrácený geokódovací službou.

Výsledný GeoJSON je následně uložen do souboru s názvem zadaným v parametru --geojson-out. Pokud uživatel název výstupního souboru neuvede, použije se výchozí hodnota results.geojson.

5 VÝSLEDKY

Tato kapitola se zaměřuje na vyhodnocení výsledků získaných různými přístupy ke sběru a zpracování dat z webových zdrojů. Byly testovány čtyři odlišné metody sběru dat: web scraping (BeautifulSoup, Selenium), využití RSS feedu a extrakce dat pomocí API. Pro získání prostorové informace z textu byla využita kombinace AI modelů, NER či regulárních výrazů s geokódováním. Každý z těchto přístupů se liší jak náročností implementace, tak kvalitou a strukturou získaných dat, což se následně promítá do přesnosti prostorové lokalizace. Cílem této kapitoly je porovnat jednotlivé metody z hlediska jejich spolehlivosti, omezení a vhodnosti pro extrakci geografických informací z textových i strukturovaných dat.

5.1 Výsledky scrapování webu bezrealitky.cz

Scraping dat o nemovitostech z portálu bezrealitky.cz byl nejpohodlnější variantou. Data jsou na stránkách strukturovaně uložena ve formátu JSON a obsahují i informaci o poloze ve formě souřadnic. Díky tomu bylo možné vynechat krok geokódování, který je ve skriptu implementován pouze pro případ neúplnosti dat.

Data byla získávána pravidelně přibližně v týdenním intervalu během února, března a dubna 2026. Největší objem dat byl nasbírán pro pronájmy bytů, více než 7 000 záznamů za celou Českou republiku. Naopak nejméně dat bylo získáno pro prodej domů (796 záznamů).

Získaná data nejsou rovnoměrně prostorově rozložena. Nejvíce záznamů pochází z velkých měst, především z Prahy, odkud pochází více než 4 000 bytů k pronájmu z celkových 7 000. Tato nerovnoměrnost se výrazně projevuje zejména při agregaci na úroveň okresů. Mnoho méně obydlených okresů obsahuje pouze jednotky záznamů, což významně ovlivňuje interpretaci výsledků. Při interpretaci je také nutné zdůraznit, že data pocházejí pouze z jednoho realitního portálu, bezrealitky.cz, který navíc nepatří mezi největší hráče na českém trhu.

I přes uvedená omezení lze z dat identifikovat základní cenové vzorce typické pro Českou republiku. Nejvyšší ceny se soustřeďují do okolí velkých měst, zejména Prahy a Brna. Cena za m² při koupi bytu zde přesahuje 120 000 Kč, zatímco u pronájmů dosahuje přibližně 350 Kč za m². V případě cen rodinných domů se k těmto centrům přidává také Ostrava. Naopak nejnižší ceny se vyskytují převážně v pohraničních regionech, například na Jesenicku, Orlicku a v severních a západních Čechách. Analýza rovněž odhaluje regiony, které se svými hodnotami vymykají okolí. Například okres Louny vykazuje překvapivě vyšší cenu za m² u pronájmů bytů než sousední okresy.

5.2 Výsledky scrapování webu ČT24

Scraping z webu ČT24 byl ve srovnání s předchozím případem náročnější. Bylo nutné analyzovat HTML strukturu stránky a pracovat s CSS třídami. Pro efektivní filtrování článků podle data bylo využito sitemap stránky.

Samotnou extrakcí článků však proces nekončil, protože bylo nutné z textu určit lokalitu, ke které se článek vztahuje. K tomu byly využity velké jazykové modely v kombinaci

s následným geokódováním. Právě tato kombinace se ukázala jako kritická a zdaleka ne stoprocentně spolehlivá. Při opakované manuální validaci byla zjištěna přesnost správného určení lokality přibližně 84 %.

Chyby vznikaly například v případech, kdy došlo k nesprávnému geokódování kvůli shodným názvům různých míst. U liniových prvků i přes správnou identifikaci a geokódování byly v některých případech vzhledem ke kontextu článku lokalizovány nepřesně, kvůli vztažnému bodu linie (Např. lokalita dálnice D1 byla identifikována správně, nicméně jednalo se o úsek v Brně, ale vztažný souřadnice geometrie podle Nominatim jsou uprostřed trasy mezi Prahou a Brnem.)

Obecně lze říci, že geokódování na úrovni států funguje velmi dobře, avšak s klesající hierarchickou úrovní (regiony, města) roste pravděpodobnost chyby. Tento trend lze pozorovat i u extrakce lokalit pomocí modelu spaCy.

Při porovnání jednotlivých AI modelů bylo zjištěno, že jejich výkonnost je velmi podobná. Zajímavým zjištěním je, že modely dokážou článek prostorově zařadit i v případech, kdy konkrétní lokalita není explicitně zmíněna v textu, pravděpodobně na základě kontextu. Bylo zpozorováno, že model Anthropic Claude ve většině případů vracel výrazně menší počet lokalit na článek. Většina článků byla nicméně lokalizována správně.

Součástí práce je také webová mapa obsahující články z března 2026. Vzhledem k tomu, že se jedná o český zpravodajský portál ČT24, převažují články lokalizované na území Česka. Zároveň lze pozorovat shluky bodů v regionech, které byly v daném období významné jak v českém, tak i globálním kontextu. Výrazná koncentrace bodů se nachází například v oblasti Blízkého východu (Írán, Spojené arabské emiráty, Izrael) a na území Ukrajiny.

Při použití delšího časového období dochází k tomu, že některé oblasti (např. Česko, USA, Izrael, Írán či Praha) obsahují velmi vysoký počet bodů, což může zhoršovat přehlednost vizualizace. Při analýze kratších časových úseků se však tento problém výrazně neprojevuje.

5.3 Výsledky zpracování RSS feedu DW

Použití RSS feedu je obecně jednodušší než klasický scraping webových stránek. RSS poskytuje jednoznačnou a přehlednou strukturu ve formátu XML, ve které se uživatel orientuje výrazně snáze než v HTML, které primárně není určeno pro přímé čtení člověkem. Je však nutné zdůraznit, že RSS kanály využívá pouze omezené množství webových stránek.

Pro získání lokace z článků byl využit jazykový model spaCy. Nevýhodou tohoto přístupu je nutnost pracovat se zahraničním zpravodajským zdrojem, jelikož nebyl nalezen vhodný model natrénovaný pro český jazyk. Zvolený model spaCy obsahuje dvě třídy použitelné pro extrakci lokalit (GPE a LOC).

Úspěšnost modelu spaCy byla srovnatelná s výkonností velkých jazykových modelů a pohybovala se okolo 80 %. Na rozdíl od předchozího přístupu je zde však pro každý článek

ponechána pouze jedna lokalita. Při zachování většího počtu lokalit docházelo k zahrnutí i míst, která byla v článku zmíněna pouze okrajově. Důvodem je skutečnost, že model nedokáže spolehlivě rozlišit kontext a význam jednotlivých lokalit. Dalším problémem byla identifikace historických názvů, například Jugoslávie, které nebyly při následném geokódování správně zpracovány.

5.4 Výsledky extrakce lokalit z API (The Guardian)

Pro získání dat z článků na webu The Guardian byla využita další alternativa ke scrapingu, a to API. API představuje rozhraní předem připravené pro uživatele k extrakci dat, díky čemuž bylo nejjednodušší získat požadované atributy článků. Nevýhodou API jsou však jeho omezení. V případě Guardianu se jedná například o limit počtu dotazů, který je stanoven na maximálně 500 denně. Zároveň je uživatel omezen pouze na prvky, které jsou v rámci API oficiálně dostupné.

Pro získání informace o lokalitě byly využity regulární výrazy. Jejich výhodou je jednoduchost a skutečnost, že není nutné instalovat externí balíčky ani odesílat požadavky na externí služby. Na druhou stranu se jedná o nejméně přesnou metodu ze všech tří testovaných přístupů. Při validaci bylo zjištěno, že pouze přibližně 60 % geokódovaných lokalit odpovídalo skutečné geografické lokalizaci článku. Stejně jako v předchozím případě byla i zde pro každý článek ponechána pouze jedna lokalita, aby se minimalizoval šum (např. jména osob či firem). Nejlépe fungovalo získávání prostorové informace z části byline (např. „Katharine Gammon in Los Angeles“). Naopak extrakce lokality přímo z textu článku byla problematičtější. Specifický problém představovaly například názvy měsíců, které se v angličtině často pojí s předložkou „in“, používanou i pro označení lokality, a zároveň začínají velkým písmenem. To vedlo k jejich chybnému vyhodnocení jako geografických názvů.

6 DISKUZE

Při hodnocení výsledků práce je nutné na úvod zdůraznit legislativní a etický kontext web scrapingu. Jak bylo uvedeno v předchozí kapitole, web scraping se pohybuje v tzv. „šedé zóně“ práva a jeho přípustnost závisí na konkrétním způsobu použití. Tato práce byla realizována výhradně pro akademické účely, což je v souladu se zákonnými výjimkami pro vědecké a výukové využití dat. V případě komerčního využití by však takto navržené scrapingové postupy mohly být problematické nebo přímo nelegální. Většina webových stránek navíc ve svých podmínkách užití automatizované získávání dat explicitně zakazuje, což představuje další významné omezení praktického nasazení scrapingu.

Z technického hlediska se ukázalo, že web scraping je silně závislý na konkrétní struktuře webové stránky. Každý web má vlastní HTML strukturu, způsob načítání dat i logiku zobrazení obsahu, což znamená, že vytvořený skript nelze jednoduše aplikovat na jiný web bez úprav. Tento fakt omezuje obecnou přenositelnost navržených řešení. V práci byly záměrně použity dvě odlišné hlavní techniky scrapingu: Parsování HTML pomocí knihoven a simulace chování uživatele pomocí nástroje Selenium. Tyto přístupy reprezentují dva základní způsoby získávání dat (statický vs. dynamický obsah). Je však nutné dodat, že existuje celá řada dalších nástrojů a knihoven (např. Scrapy, Playwright), které by bylo možné využít a které mohou být v některých případech efektivnější.

Z hlediska získávání dat se jednoznačně ukázalo, že pokud jsou dostupné strukturované zdroje, jako jsou API nebo RSS feedy, je jejich využití výrazně jednodušší než klasický web scraping. Tyto přístupy eliminují potřebu parsování HTML a snižují riziko chyb způsobených změnou struktury webu. Jejich nevýhodou je však omezený rozsah dostupných dat.

Hlavním tématem práce byla extrakce prostorové informace. Výsledky ukazují, že žádná z použitých metod extrakce lokality z textu není plně spolehlivá. Použití velkých jazykových modelů dosahuje relativně dobrých výsledků, jejich výstup však není vždy konzistentní a nelze očekávat stoprocentní přesnost. Regulární výrazy se ukázaly jako nejméně vhodný přístup, protože nejsou schopny zachytit kontext a vedou k vysoké míře chybovosti. Metoda založená na knihovně spaCy poskytuje podobnou úspěšnost výsledků jako velké jazykové modely, avšak jejich nevýhodou je absence modelu pro český jazyk.

Dalším problémem byla fáze geokódování. Přesnost výsledků závisí nejen na kvalitě extrahované lokality, ale i na vlastnostech geokódovací služby. V některých případech docházelo k chybám způsobeným nejednoznačností názvů nebo nevhodně určeným vztažným bodem liniových či plošných prvků vzhledem ke kontextu článku.

Z hlediska celkového přínosu práce lze konstatovat, že web scraping představuje využitelný nástroj pro získávání prostorově lokalizovatelných dat, zejména v případech, kdy nejsou dostupná jinými způsoby. Zároveň je však nutné brát v úvahu jeho omezení, a to jak technická, tak legislativní a metodologická. Do budoucna by bylo vhodné zaměřit se na kombinaci různých přístupů k extrakci lokalit, která by mohla vést ke zvýšení přesnosti výsledků.

7 ZÁVĚR

Cílem této diplomové práce bylo analyzovat potenciál web scrapingu při získávání prostorově lokalizovatelných dat z webového prostředí a posoudit jejich využitelnost pro následné prostorové zpracování, analýzu či vizualizaci. Práce se zaměřila na různé typy webových zdrojů, od klasických webových stránek přes RSS kanály až po aplikační rozhraní API, a ověřila, jakým způsobem lze z těchto zdrojů získávat data s přímou nebo nepřímou prostorovou informací.

V teoretické části byla nejprve vymezena problematika web scrapingu a jeho porovnání s dalšími způsoby získávání dat z webu, jako jsou otevřená data, API nebo RSS formáty. Dále byly popsány základní přístupy ke scrapingu podle způsobu získávání dat, struktury zdrojových dat a metod extrakce. Zvláštní pozornost byla věnována získávání prostorové informace z textových webových dat, protože právě tato část představuje jednu z hlavních výzev při práci s nestrukturovanými zdroji. Součástí práce bylo také shrnutí technických, právních a etických omezení, která mohou využití web scrapingu ovlivňovat.

Praktická část práce byla založena na návrhu několika odlišných postupů získávání dat. Byly zvoleny čtyři zdroje reprezentující různé typy webových dat: realitní portál Bezrealitky.cz, zpravodajský web ČT24, RSS kanál Deutsche Welle a API deníku The Guardian. U portálu Bezrealitky.cz byl realizován scraping realitních inzerátů, u nichž byla prostorová informace získávána především z adres nebo dostupných souřadnic. Data byla ukládána do prostorové databáze PostgreSQL/PostGIS a dále využita pro agregaci a vizualizaci. V případě ČT24 byl použit postup založený na získávání článků ze sitemap a následném zpracování jejich obsahu pomocí velkých jazykových modelů. U Deutsche Welle byla testována extrakce lokalit z RSS kanálu pomocí metody Named Entity Recognition a knihovny spaCy. U The Guardian byl využit přístup založený na API a regulárních výrazech, které identifikovaly jazykové vzory naznačující lokalitu události.

Výsledky práce ukázaly, že web scraping může být použitelným nástrojem pro získávání aktuálních prostorových dat z webového prostředí, zejména v případech, kdy nejsou data dostupná prostřednictvím otevřených dat nebo běžných geoinformačních služeb. Zároveň se potvrdilo, že vhodnost konkrétního postupu závisí především na charakteru zdrojového webu a na podobě prostorové informace. Pokud jsou souřadnice nebo adresa dostupné přímo ve strukturované podobě, je následné zpracování relativně spolehlivé. Naopak u textových dat je nutné nejprve lokalitu rozpoznat, interpretovat v kontextu a následně geokódovat, což zvyšuje riziko chyb i nejistotu výsledků.

Bylo zjištěno, že neexistuje univerzální metoda vhodná pro všechny typy webových zdrojů. Pro strukturovaná data je nejvhodnější přímá extrakce ze strukturovaných částí webu, například JSON. Pro polostrukturovaná HTML data lze efektivně využít knihovny jako BeautifulSoup nebo Selenium, přičemž Selenium je vhodné zejména u dynamicky načítaného obsahu. Pro nestrukturovaná textová data jsou naopak důležité metody extrakce geografických entit, například regulární výrazy, NER nebo velké jazykové modely. Tyto metody se liší přesností, náročností implementace i schopností zohlednit kontext.

Práce dále ukázala, že samotné získání dat představuje pouze první krok celého procesu. Aby bylo možné webová data využít v GIS, je nutné je vyčistit, sjednotit, geokódovat, uložit do vhodného formátu a následně vizualizovat. V rámci práce byla výsledná data ukládána do formátu GeoJSON nebo do prostorové databáze a následně zobrazována pomocí webových map. Tím bylo ověřeno, že web scraping může sloužit nejen jako metoda sběru dat, ale také jako součást širšího workflow vedoucího od získání dat až po jejich publikaci a kartografickou prezentaci.

Web scraping se ukázal jako užitečný, technicky náročnější nástroj, který může vhodně doplňovat tradiční zdroje prostorových dat. Jeho největší přínos spočívá zejména v možnosti získávat data, která jiným způsobem často nejsou dostupná. Pro jeho další využití je však nezbytné respektovat technické limity zdrojů, kvalitu získaných dat a právní i etické podmínky jejich zpracování.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

ATTARD, Judie, Leonardo ORLANDO, Simon SCERRI a Sören AUER. A systematic review of open government data initiatives. *Government Information Quarterly*. 2015, 32(4), s. 399–418. ISSN 0740-624X.

BRENNING, Alexander a Andreas HENN. Web scraping for spatial data science. *International Journal of Geographical Information Science*. 2023.

CAVALLO, Alberto. Scraped data and sticky prices. *American Economic Review*. 2018, 108(2), s. 519–522. ISSN 0002-8282.

GEORSS. GeoRSS specification [online]. 2006 [cit. 2026-05-01]. Dostupné z: <http://www.georss.org>

GOLDBERG, Daniel W., John P. WILSON a Craig A. KNOBLOCK. From text to geographic coordinates: The current state of geocoding. *URISA Journal*. 2007, 19(1), s. 33–46.

HU, Xuke, Zhiyong ZHOU, Hao LI, Yingjie HU, Fuqiang GU, Jens KERSTEN, Hongchao FAN a Friederike KLAN. Location reference recognition from texts: A survey and comparison [online]. 2022 [cit. 2026-05-01]. Dostupné z: <https://arxiv.org/pdf/2207.01683>

KAHLON, Navroz a Williamjeet SINGH. A Systematic Review of Web Scraping: Techniques, LLM-Enhanced Approaches, Performance Metrics, and Legal-Ethical Issues. SSRN, 2025. Dostupné z: <https://ssrn.com/abstract=5429131>

KOBAYASHI, Kazuhisa a Koichi TAKEDA. Information retrieval on the Web. 2000.

KROTOV, Vladislav a Leiser SILVA. Legality and ethics of web scraping. *Communications of the ACM* [online]. 2018[cit. 2026-05-01]. Dostupné z: https://www.researchgate.net/publication/324907302_Legality_and_Ethics_of_Web_Scraping

MARTÍNEZ LLAMAS, Javier, Koen VRANCKAERT, Davy PREUVENEERS a Wouter JOOSEN. Balancing Security and Privacy: Web Bot Detection, Privacy Challenges and Regulatory Compliance under the GDPR and AI Act. *Open Research Europe*. 2025, roč. 5, čl. 76. DOI: 10.12688/openreseurope.19347.1. Dostupné z: <https://doi.org/10.12688/openreseurope.19347.1>

MIDDLETON, Stuart E., Giorgos KORDOPATIS-ZILOS, Symeon PAPADOPOULOS a Yiannis KOMPATSIARIS. Location extraction from social media: geoparsing, location disambiguation and geotagging [online]. *ACM Transactions on Information Systems*. 2018, 36(4), s. 1–27 [cit. 2026-05-01]. Dostupné z: https://eprints.soton.ac.uk/419443/1/location_extraction_from_social_media.pdf

MITCHELL, Ryan. Web Scraping with Python. 2nd ed. Sebastopol: O'Reilly Media, 2018. ISBN 978-1-4919-8273-3.

MUTLU, Muhammed Ali. A web scraping app for smart literature search of the keywords. Online. Č. 2024. [cit. 2026-05-01]. Dostupné z: <https://doi.org/10.7717/peerj-cs.2384>

OGC – OPEN GEOSPATIAL CONSORTIUM. OGC GeoRSS Encoding Standard [online]. 2017 [cit. 2026-05-01]. Dostupné z: <https://www.ogc.org>

OGC – OPEN GEOSPATIAL CONSORTIUM. OGC Web Feature Service (WFS) Standard. 2019.

OPEN KNOWLEDGE FOUNDATION. Open Definition 2.1 [online]. 2015 [cit. 2026-05-01]. Dostupné z: <https://opendefinition.org>

ROSATI, Eleonora. The DSM Directive Five Years On [online]. SSRN. 2024 [cit. 2026-05-05]. DOI: 10.2139/ssrn.4992748. Dostupné z: <https://ssrn.com/abstract=4992748>

SAURKAR, Anand V., Kedar G. PATHARE a Shweta A. GODE. An Overview On Web Scraping Techniques And Tools [online]. International Journal on Future Revolution in Computer Science & Communication Engineering. 2018, 4(4), s. 363–367 [cit. 2026-05-05]. Dostupné z: <https://www.ijfrcsce.org/index.php/ijfrcsce/article/view/1529/1529>

SCRAPY. Selecting dynamically-loaded content. Scrapy 2.15 documentation [online]. 2026 [cit. 2026-05-01]. Dostupné z: <https://scrapy.readthedocs.io/en/latest/topics/dynamic-content.html>

TELEC, Ivo a Pavel TŮMA. Autorský zákon: komentář. Praha: C. H. Beck, 2019.

VOIGT, Paul a Axel VON DEM BUSSCHE. The EU General Data Protection Regulation (GDPR): A Practical Guide. Cham: Springer, 2017.

PŘÍLOHY

SEZNAM PŘÍLOH

Volné přílohy

Příloha 1	Web
Příloha 2	Kompletní skripty a aplikace vytvořené v rámci diplomové práce
Příloha 3	Poster
Příloha 4	Vstupní data
Příloha 5	Výstupní data