Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačního a znalostního inženýrství

Web Scraping s CasperJS

4IZ470 Dolování znalostí z webu

Student: Bc. Viet Bach Nguyen

Vyučující: doc. Ing. Vojtěch Svátek, Dr.

Datum: 12. května 2017

Abstrakt

Tato seminární práce pojednává o problematice extrahování dat z webu z praktického hlediska. Cílem této práce je implementace webového scraperu pro vybrané webové stránky. Práce je řešena v kontextu předmětu 4IZ470 Dolování znalostí z webu, mezi jehož zaměření patří WCM – web content mining, tj. dolování, extrakce a integrace užitečných dat, informací a znalostí z webového prostředí.

Problém je vyřešen pomocí skriptovacího nástroje CasperJS pro automatizaci zobrazování obsahu webových stránek. Skriptování je zhotoveno v programovacím jazyce JavaScript. Nejdříve je provedena analýza vybraných webů. Výběr těchto webů podléhá míře potenciálu obsažených dat vyhovujících byznysovému záměru. Analýza zahrnuje zkoumání obsahu těchto webů i jejich struktury. Na základě analýzy je vytvořen algoritmus pro automatizovanou extrakci dat z těch to webů.

Výstupem této práce je hotový web scraper a dokumentace pro jeho konfiguraci a spuštění. Tento scraper je schopný stahovat data z webu v požadovaném tabulkovém formátu, jehož sloupce lze nadefinovat v konfiguračním souboru. Pro ukázku výstupu je přiloženo několik datových souborů obsahující získané informace pomocí vytvořeného scraperu. Přínosem této práce je uskutečnění smysluplného experimentu nad studií o struktuře moderních webů a uvedení jednoho z mnoha možných, ne-li nejlepších, způsobů získávání dat z nich.

Klíčová slova

Web Content Mining, Web Scraping, strukturovaná data, headless prohlížeče, automatizace, skriptování, CasperJS, PhantomJS, JavaScript.

Obsah

1 Úvod

Hledání znalostí je v rámci každé vědecké disciplíny nesmírně důležitým východiskem pro nalezení nových objevů a invencí. Přítomnost celosvětové sítě WWW a internetu znamená pro svět významnou akceleraci rozvoje ve všech vědních oblastí za pomocí zpřístupnění enormního množství dat v otevřeném prostředí. Tato data v sobě skrývají mnoho byznysu relevantních znalostí, které se mohou velice dobře uplatnit v dnešním ekonomickém prostředí. Problém je ale jak tato data vůbec získat.

1.1 Představení problému

S uplynutím času je internet stále řízen čím dál tím masivnějším tokem informací, proto dnes již není tak jednoduché získávat data dle specifických potřeb. Dalším problémem je rychle rostoucí se trend rozvoje webových technologií. Tyto technologie jsou hlavním důvodem zralosti dnešního webového prostředí. To se projevuje zejména v dynamických webových aplikací, které postupně a jistě nahrazují tradiční způsob prezentace informací na webu.

Pro získávání dat z moderních webů je vyžadováno použití moderních nástrojů, které jsou na příchod nových technologií připraveny. Jedním z přístupů v řešení problému dynamického webu z pohledu získávání webových dat je použití speciálního internetového prohlížeče, který je srovnatelný s klasickými prohlížeči a který se navíc dá zautomatizovat. Příkladem takového internetového prohlížeče je PhantomJS[1] nebo HtmlUnit.

1.2 Přístup a cíle

Internet se dá považovat za distribuované úložiště potenciálních dat, která mají řadu možných způsobů využití. K tomu, aby data získaná z webu byla k něčemu užitečná, zaprvé musí být zajištěna a zadruhé musí být zpracovávána do akceptovatelné podoby ještě předtím, než budou použity v dalších fázích dolování znalostí jako např. předzpracování.

Nejideálnější případ extrakce je, když jsou všechna tato data dostupná na jednom místě v rámci určitého webu. Skutečnost je ale taková, že často jsou data zpřístupněna v rozpadlých dávkách podle toho, v jakém okamžiku jsou pro uživateli jak relevantní během jeho pobytu na daném webu. Zpravidla je žádoucí při extrakci mít všechna data sloučena do jednoho výsledného souboru, což usnadňuje jejich další zpracování. Proto

je obvykle potřeba tato data postupně posbírat z více místech na webu v různých okamžicích prohlížení. Z tohoto důvodu je potřeba použít sofistikovanější algoritmus pro zachytávání těch to dat.

Hlavní obsah této práce se zabývá implementací automatizace extrahování dat pomocí skriptovacího nástroje CasperJS[2]. Implementace bude probíhat v jazyce JavaScript s použitím programovacího rozhraní poskytovaného nástrojem CasperJS. Předmětem extrakce budou weby s informacemi o podnicích různých oborů. Extrahovaná data pak budou sloužit jako podklad pro vývoj určité aplikace, která je hlavním projektem vymýšlené firmy. Cílem práce je tedy vytvoření soubor webových scraperů pro automatické získávání těchto dat.

1.3 Postup práce

Práce je rozčleněna do následujících částí, přičemž jejich pořadí vychází ze skutečného postupování a krokování v práci.

1. Rozbor problematiky moderního webu
2. Představení byznysového záměru
3. Výběr vhodných webových stránek pro extrakci
4. Obsahová analýza vybraných webů
5. Analýza struktury vybraných webů
6. Implementace scraperu
7. Zápis dokumentace vytvořeného programu
8. Zhodnocení výhod a nevýhod používaného nástroje
9. Diskuze o legálnosti scrapingu

1.4 Používané nástroje a knihovny

Vybraným nástrojem pro implementaci web scraper v rámci této práce je CasperJS. Tento open-source nástroj slouží ke skriptování prohlížeče a jeho automatizaci. Primárně je CasperJS navržen pro použití s headless WebKit prohlížečem PhantomJS a sekundárně s Gecko prohlížečem SlimerJS. Pro účely extrahování dat je dostačující použít jen PhantomJS.

V rámci CasperJS a PhantomJS lze psát běžné JavaScriptové kódy a proto je možné využít všech výhod a standardních funkcí tohoto programovacího jazyka. Pomocí těchto nástrojů je možné vytvořit plnohodnotný skript pro zachycení a simulaci případu užití běžného uživatele včetně jeho interakce s webovými stránkami včetně přesměrovávání a odesílání formulářů.

Pro účely extrakce dat přímo z dokumentového objektu webové stránky je navíc použita knihovna jQuery. Tato knihovna slouží k usnadnění procházení jednotlivých HTML tagů a parsování čistá data z nich.

2 Moderní web

S rychlým růstem webových technologií roste i složitost získávání dat. Dnešním trendem ve vývoji webových aplikací spočívá v použití a integrace různých revolučních technologií jako AJAX, WebSocket, ServiceWorker, WebStorage a další k vytváření lepšího prostředí pro interakce s uživateli. Dále tyto weby bývají navrženy jako single-page aplikace, která zlepšuje uživatelskou přívětivost a výkonnost systému. To však ale znamená, že weby se stávají dynamickými, což zkomplikuje proces extrakce dat z nich.

Všechny tyto technologie přináší na webové prostředí z hlediska systémového návrhu odlišný přístup v zobrazování informací uživatelům. Tento princip spočívá v dodatečném načítání informačních zdrojů až v okamžiku, kdy je to potřeba. Dále se jedná o dynamičnosti webové stránky, kdy struktura dokumentového objektu webové stránky se mění v závislosti na interakce uživatele. To vše se děje asynchronně v pozadí aplikace. Tyto změny však uživatelé nepoznají, ale pro účely extrakce dat to znamená složité simulace těchto událostí, bez které není šance získat určená data.

Jedním z možných řešení tohoto problému spočívá v skutečné simulaci uživatelských interakcí s webovou stránkou. K tomu je zřejmě potřeba webový prohlížeč, avšak takový, který se dá zautomatizovat.

3 Byznysový záměr

Tempor litora nisl in, litora ante quisque. Vitae conubia pede, mauris augue aliquam ultricies in nam, bibendum urna, eu nam magna commodo fringilla morbi. Nulla orci sed, tortor velit magna nec enim odio pellentesque, ultricies sodales platea vel varius lacus nullam, mi ante et, orci phasellus. Elementum cursus magna erat, lobortis adipiscing, ullamcorper malesuada, lorem wisi lorem vitae nunc, officia porta et pharetra

gravida mauris. Nulla phasellus ipsum vitae, id turpis maecenas et, turpis fusce. Quis accumsan, leo arcu ut, id aenean morbi amet porttitor. Aliqua sit gravida pellentesque nunc tellus. Praesent nulla metus pharetra magna, ante erat nullam, ut eius vel at turpis nulla vel. Hymenaeos vivamus et, ornare at mauris. Etiam a hendrerit suscipit.

4 Výběr webových stránek

Vehicula rhoncus erat curabitur praesent volutpat, bibendum nec scelerisque eros tempus tristique justo, euismod eu in, nunc ac libero facilisis. Sed orci faucibus dui. Arcu eleifend turpis, a rerum praesent vulputate non sollicitudin, aliquet rutrum porta quis. A urna fusce lacus. Mi nam erat sem, morbi amet amet, placerat aliquam sociosqu justo lorem mi, in nisl consequat leo aliquam. Sem elit augue dictum et nunc, eget consequat suscipit ridiculus in, dolor pretium nostra sed.

5 Obsahová analýza vybraných stránek

A urna fusce lacus. Mi nam erat sem, morbi amet amet, placerat aliquam sociosqu justo lorem mi, in nisl consequat leo aliquam. Sem elit augue dictum et nunc, eget consequat suscipit ridiculus in, dolor pretium nostra sed.

6 Analýza struktury vybraných stránek

Nulla orci sed, tortor velit magna nec enim odio pellentesque, ultricies sodales platea vel varius lacus nullam, mi ante et, orci phasellus. Elementum cursus magna erat, lobortis adipiscing, ullamcorper malesuada, lorem wisi lorem vitae nunc, officia porta et pharetra gravida mauris. Nulla phasellus ipsum vitae, id turpis maecenas et, turpis fusce. Quis accumsan, leo arcu ut, id aenean morbi amet porttitor. Aliqua sit gravida pellentesque nunc tellus. Praesent nulla metus pharetra magna, ante erat nullam, ut eius vel at turpis nulla vel.

7 Implementace scraperu

Arcu eleifend turpis, a rerum praesent vulputate non sollicitudin, aliquet rutrum porta quis. A urna fusce lacus. Mi nam erat sem, morbi amet amet, placerat aliquam sociosqu

justo lorem mi, in nisl consequat leo aliquam. Sem elit augue dictum et nunc, eget consequat suscipit ridiculus in, dolor pretium nostra sed.

8 Dokumentace programu

Elementum cursus magna erat, lobortis adipiscing, ullamcorper malesuada, lorem wisi lorem vitae nunc, officia porta et pharetra gravida mauris. Nulla phasellus ipsum vitae, id turpis maecenas et, turpis fusce. Quis accumsan, leo arcu ut, id aenean morbi amet porttitor. Aliqua sit gravida pellentesque nunc tellus. Praesent nulla metus pharetra magna, ante erat nullam, ut eius vel at turpis nulla vel. Hymenaeos vivamus et, ornare at mauris. Etiam a hendrerit suscipit.

9 Výhody a nevýhody nástroje

Quis accumsan, leo arcu ut, id aenean morbi amet porttitor. Aliqua sit gravida pellentesque nunc tellus. Praesent nulla metus pharetra magna, ante erat nullam, ut eius vel at turpis nulla vel. Hymenaeos vivamus et, ornare at mauris. Etiam a hendrerit suscipit.

10 Legálnost extrakce dat z webů

Erat elit vestibulum wisi sed ut, vel ac donec urna non, feugiat maecenas, quis metus condimentum, nunc porro felis wisi nam suspendisse. Vulputate sociosqu id fermentum elit, quam magna quaerat purus nec dolor, ipsum quam, lobortis mauris laudantium dolor bibendum, donec viverra dis sed.

Reference

[1] Hidayat, A. Phantomjs. [2017] http://phantomjs.org/. Cit. [12. 5. 2017].

[2] Perriault, N. Casperjs, a navigation scripting and testing utility for phantomjs and slimerjs. [2016] http://casperjs.org/. Cit. [12. 5. 2017].

Přílohy

- Projekt je publikován pod open-source licencí MIT na gitovém repozitáři https://github.com/nvbach91/4IZ470
- Zdrojový kód projektu se nachází ve složce `project`.
- Zdrojový kód textu práce se nachází ve složce `paper`.