EventParser

Extrakce obsahu HTML stránky popisující událost v daném místě a čase

1. Účel

Cílem je ze zadané URL adresy získat co nejformativnější a zároveň nejkratší část textu popisující určitou událost. Tato událost je zadaná datem konání a místem (městem). Proto je potřeba osekat nepotřebné části HTML struktury (zápatí, menu, reklamy), zjistit relevanci odkazu a osekat nepotřebný obsah (zajímá nás název/typ události, místo, čas a příp. cena).

Projekt je řešen v programovacím jazyce Python 2.7 za použití knihoven Beautiful Soup, Datetime, RE, System, URL Lib a Date Util. Odkazy na dokumentaci těchto knihoven naleznete v seznamu zdrojů.

1.1. Související projekty

Projekt, jenž by řešil stejný problém a byl volně přístupný, doposud neexistuje, či nebyl nalezen. Za související projekty považujeme programy, které vybírají a ořezávají obsah (převážně text) HTML stránky. Takovými projekty jsou například:

- **Boilerpipe**: „...provides algorithms to detect and remove the surplus "clutter" around the main textual content of a web page.“ [1]
- **HTML::Content::Extractor**: „Receiving a main text of publication from HTML page and main media content that is bound to the text.” [2]
- **CiteSeer**: „Extracting Content from Accessible Web Pages” [3]
- **Readability:** „…turns any web page into a clean view for reading...“ [4]

Dále existují projekty zabývající se získáváním informací z textu. Ty jsou založeny převážně na principech strojového učení:

- Komerční projekty:
  - **ClearForest** [5]
  - **MUC**: Information Extraction [6]
- Bakalářské a diplomové práce:
  - Extrakce informací z hypertextu [7]
  - Extrakce informací z textu [8]
  - Extrakce strukturních informací z běžného textu na základě syntaktického analyzátoru [9]
  - Extrakce textových dat z internetových stránek [10]

2. Implementace

Sekvence úkolů po obdržení vstupu

1. Načtení obsahu HTML stránky
2. Osekání nepotřebných částí těla HTML struktury (haeder, script, style)
3. Extrakce titulu a nadpisů h1
4. Hledání nejkratší vhodné části textu
5. Detekce multiudálosti
6. Zobrazení výsledku výsledku

2.1. Načtení obsahu HTML stránky

Je použita knihovna Beautiful Soup, která umožňuje jednoduché procházení HTML struktury a obsahuje velmi užitečnou funkci get_text().

2.2. Osekání nepotřebných částí těla HTML struktury

Informace o události jsou hledány pouze v těle stránky (tag body), proto je nutné vyloučit struktury typu script a style. Bylo testováno i použití existujících knihoven, to se ale ukázalo být zbytečně složité.

2.3. Extrakce titulu a nadpisů h1

Titul stránky a hlavní nadpis často obsahuje klíčové informace o zkoumané informace. Stává se však, že jedna nebo druhá struktura na stránce chybí, proto jsou zkoumány obě.

2.4. Hledání nejkratší vhodné části textu

Zde se jedná přímo o vyhledání co nejkratšího úseku HTML struktury, který obsahuje jak místo, tak datum. Struktura se prohledává do hloubky, rekurzivně se hledá první výskyt. Toto je nejklíčovější část celého procesu a díky hledání datumu také nejkomplikovanější. Datum může mít rozličnou strukturu, používat různé oddělovače, některé údaje mohou být vypuštěny či neznámé a měsíc může být vyjádřen slovně – což je dále komplikováno českým skloňováním.

Za účelem vyhledání dat v textu právě vzniká nástroj Czech Date Parser inspirovaná Parserem v balíku Date Util.

2.5. Zobrazení výsledku výsledku

Před zobrazením nalezeného výsledku se ještě odstraní přebytečné bílé znaky. Na standardní výstup se také tisknou dodatečné informace o délce zpracovaného textu.

3. Testování

Data se získala pomocí jednoduchého opakovaného automatizovaného vyhledávání pomocí Google Search a příslušné Python knihovny Google. Nashromážděno bylo zhruba 160 stránek, z čehož 60 stránek byly chybné vstupy (včetně chybných multiudálostí).

Data byla dále ručně značkována (zda se jedná o chybný/správný vstup, jaké informace jsou k nalezení na stránce) a zpracovávána. Vyhodnocení probíhalo dvojí: automaticky a ručně. Automaticky se vyhodnocovalo, zda program správně urči, že se jedná o událost, a ručně se poté potvrzovalo, jak velká část zveřejněných informací byla ve výstupním textu vrácena.

Vzhledem k neúplnosti českého Date Parseru, probíhalo testování dvojí: za pomoci striktního a tolerantního vyhledávače dat. Jak ukazují následující výsledky, striktní vyhledávač má tendence k falešné negativitě, nedokáže najít událost v korektním vstupu, a tolerantní vyhledávač je zase naopak často falešně pozitivní.

Tolerantní hledání dat

Event	All data	78**%**
	Correct	**90%**
	Incorrect	57%
Info	All data	38%
	Marked as event	43%

Striktní hledání dat

Event	All data	70**%**
	Correct	54%
	Incorrect	**97%**
Info	All data	44%
	Marked as event	**85%**

První část tabulky „Event“ popisuje chování programu při jednoduchém určování, zda se na stránce vyskytuje událost v daném místě a čase nebo ne. Druhá část „Info“ ukazuje, jak je program úspěšný ve vracení co nejformativnější části textu – kolik dostupných informací bylo ve výsledném textu obsaženo. Ač je v této části Striktní vyhledávání úspěšnější, uživateli zůstává možnost vybrat si, jaký vyhledávač datumu chce použít.

4. Spuštění

Pro spuštění je také nezbytné mít nainstalovaný Python 2.7, který obsahuje většinu použitých knihoven, a nainstalovat knihovny, které v něm obsaženy nejsou: BeautifuSoup a DateUtil. Více informací o těchto knihovnách naleznete na konci tohoto souboru ve zdrojích.

EventParser.py musí obdržet tři vstupní parametry: URL adresu, na které se má událost hledat, datum, kdy proběhne, a město, ve kterém se koná. Čtvrtý parametr (pravdivostní hodnota) udává, zda se má použít tolerantní vyhledávání nebo ne.

4.1. Příklady

Ukázka na stránkách Tomáše Kluse.

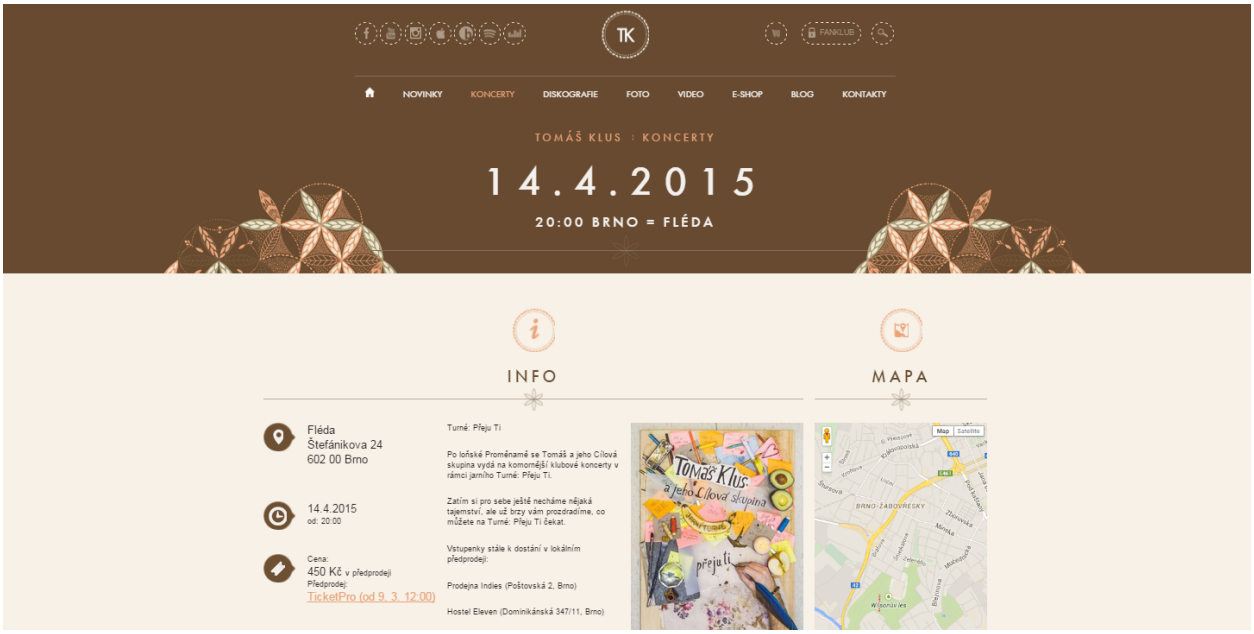
Příklady vstupů:
> python Event_Parser_final.py http://www.tomasklus.cz/koncerty/brno-3/ 14.4.2015 Brno
> python Event_Parser_final.py http://www.katapult.cz/koncerty/pisek-0 18.4.2015 Písek
> python Event_Parser_final.py http://semilasso.cz/akce/detail/kabat-open-air-turne-2015-brno-bvv 16.06.2015 Liberec

Chybný vstup:
> python Event_Parser_final.py http://semilasso.cz/akce/detail/kabat-open-air-turne-2015-brno-bvv 19.06.2015 Liberec

4.2. Příklad Tomáš Klus

Vstup:
> python Event_Parser_final.py http://www.tomasklus.cz/koncerty/brno-3/ 14.4.2015 Brno

Stránka:



Výstup:

```
Whole body text lenght: 2320
Selected text lenght: 166

Start of final text:
---------
Title: 13.4.2015 20:00 Brno = Fléda | Tomáš Klus
Headline: 13.4.201520:00 Brno = Fléda

Text:
FlédaŠtefánikova 24602 00 Brno
13.4.2015od: 20:00

Cena:
450 Kč v předprodeji
Předprodej:
TicketPro (od 5. 3. 12:00)
---------
End of final text.
```

5. Zdroje

5.1. Související projekty:

[1] https://code.google.com/p/boilerpipe/
[2] https://metacpan.org/release/HTML-Content-Extractor
[3]http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.357
[4] https://readability.com/
[5] http://www.clearforest.com/index.html
[6] http://www-nlpir.nist.gov/related_projects/muc/
[7] http://is.muni.cz/th/60743/fi_m/
[8] http://is.muni.cz/th/51819/fi_m/
[9] http://is.muni.cz/th/172962/fi_b/
[10]https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/20916/Zdenek_Mazal.pdf?sequence=1&isAllowed=y

5.2. Python knihovny:

Beautiful Soup
- documantace: https://dateutil.readthedocs.org/en/latest/
- download: https://pypi.python.org/pypi/beautifulsoup4

Date Util
- dokumantace: http://www.crummy.com/software/BeautifulSoup/bs4/doc/
- download: https://pypi.python.org/pypi/python-dateutil/

Datetime - dokumantace: https://docs.python.org/2/library/datetime.html

RE - dokumantace: https://docs.python.org/2/library/re.html

Sys – dokumantace: https://docs.python.org/2/library/sys.html

URL Lib – dokumantace: https://docs.python.org/2/library/urllib.html