Čištění zpravodajských webových stránek

Jaromír Novotný[1]

1 Úvod

Nejvíce používaná možnost publikování informací jsou webové stránky, kde je obsaženo ohromné množství snadno dostupného gramaticky správného textu. Hlavní důvod, proč čistit zpravodajské webové stránky, je vytvořit jazykové modely. Samozřejmě aby se jazykové modely nemuseli vytvářet manuálně, je vhodné tento proces zautomatizovat. Tedy zde bude uvedena jedna možnost (algoritmus) pro automatické čištění. Dále také dvě možnosti ohodnocení výsledků tohoto algoritmu.

2 Automatický čistící algoritmus

Algoritmus je psán v Pythonu. Hlavní inspirace je z článku (Marek , 2008, Victor: the Web-Page Cleaning Tool). Vstupem algoritmu jsou html (webové) stránky. Algoritmus provádí následující kroky: příprava webových stránek ke zpracování (zahrnuje i načtení stránek), trénování (pouze v případě, že není natrénován nebo že se chce vyzkoušet jiná trénovací sada dat), přiřazování (čištění) a ohodnocení výsledků. Důležitá část algoritmu obsahuje algoritmus podmíněného náhodného pole (CRF - Conditional Random field). Byl použit již existující algoritmus (Okazaki , 2013, CRFsuite) CRF (psaný v c++) a knihovna do Pythonu (Korobov , 2014, Python-crfsuite) dovolující používat algoritmus (Okazaki , 2013, CRFsuite).

2.1 Příprava webové (html) stránky ke zpracování

K načtení stránek je využívána knihovna do Pythonu (Richard , 2013, Beautiful Soup). Po načtení stránek algoritmus vybere části (opět využití BeautifulSoup), které jsou ohraničeny tagy (včetně): $< h1 >< \backslash h1 >$, $< li >< \backslash li >$, $< p >< \backslash p >$. Inspirace k tomuto postupu pochází z (Mingsheng , 2012, An Approach for Text Extraction From Web News Page). Dále se jednoduchým parserem odstraní tagy v datech a tedy zůstane čistý text. Předpokládá se, že mezi začínajícím a konečným tagem je odstavec a tedy jsou data ve formě odstavců textu (samozřejmě některé odstavce jsou např. odkazy nebo reklamy a proto je důležité provést čištění). Takto připravená data jsou vstupem buď trénování nebo přiřazování (čištění).

2.2 Trénování

V případě trénování, se použijí připravená data. Trénování probíhá manuálně a to tak, že program vypíše odstavec ze vstupní stránky na monitor a uživatel se rozhodne zda mu přiřadí HEADER, TEXT a nebo OTHER. Po dokončení přiřazení všech odstavců ze vstupu (ze všech stránek ve složce Vstup) je spuštěno trénování CRF.

[1] student doktorského studijního programu Aplikované vědy a informatika, obor Kybernetika a řídící technika, specializace Umělá inteligence, e-mail: fallout7@kky.zcu.cz

2.3 Přiřazování (čištění)

Opět vstupem jsou připravená data (ovšem testovací nikoli trénovací jako v případě použití trénování). Zde je každému odstavci přiřazena (díky CRF) hodnota HEADER, TEXT nebo OTHER. Odstavce dané stránky s označením HEADER a TEXT (tedy vyčištěná data) se následně zapíší do souboru se stejným názvem jako má daná vstupní webová stránka.

2.4 Ohodnocení výsledků

Pro tento účel byla vybrána Levenshteinova vzdálenost, což je nejčastěji používaná míra pro porovnání dvou řetězců. Dále jsou počítány hodnoty Přesnost, Úplnost a F-skóre. Aby vůbec výsledky získané z čistícího algoritmu mohli být ohodnoceny je potřeba mít manuálně vyčištěné texty z webových stránek, které chceme ohodnotit.

3 Výsledky

Vstupní data získány ze čtyř zdrojů: denik.cz (DEN), lidovky.cz (LID), ihned.cz (IHN) a idnes.cz (IDS). Z každého zdroje je vytvořena sada jak trénovacích tak testovacích dat. K testovacím datům je vytvořena také sada dat (manuálně vyčištěných) potřebná k ohodnocení. Natrénování a testování je provedeno jak pro každý zdroj zvlášť tak pro všechny dohromady. Výsledky porovnány v Tabulce 1, kde L. vz. značí Levenshteinovu vzdálenost, P je přesnost, R je úplnost a F je f-skóre. Výsledné hodnoty představují průměr přes všechna data.

	DEN	LID	IHN	IDS	All
L. vz.	748.3	1023.45	616.5	921.9	668.293
P	0.918	0.789	0.837	0.828	0.853
R	0.734	0.820	0.621	0.694	0.751
F	0.874	0.795	0.782	0.797	0.830

Tabulka 1: Tabulka ohodnocení automatického čistícího algoritmu pro dané vstupy

4 Závěr

Tento čistící algoritmus je stále ve fázi vývoje, ale z výsledků v Tabulce 1 je vidět, že tento algoritmus by podle mého názoru mohl po dalším vyladění být vcelku užitečný. Dalším cílem do budoucna je tento algoritmus porovnat s jinými a tím lépe získat přehled o jeho kvalitě.

Literatura

Marek M., Spousta M., Pecina P., 2008., *Victor: the Web-Page Cleaning Tool* , In Proceedings of the 4th Web as Corpus Workshop, LREC (2008)

Mingsheng Hu, Zhijuan J., 2012. *An Approach for Text Extraction From Web News Page* , 2012 IEEE Symposium on Robotics and Applications(ISRA)

Richard L. 2013, *Beautiful Soup*, http://www.crummy.com/software/BeautifulSoup/

Korobov M., Peng T., 2014, *Python-crfsuite*, http://python-crfsuite.readthedocs.org/en/latest/

Okazaki N., 2013, *CRFsuite*, http://www.chokkan.org/software/crfsuite/