Obecné čištění webových stránek

Jaromír Novotný[1]

1 Úvod

Webové stránky představují velmi bohatý zdroj dat v podobě textu, jenž může být dále využit na různé úlohy NLP (přirozeného zpracování jazyka) a to hlavně na jazykové modelování. Největším problémem tohoto zdroje je, že webové stránky obsahují kromě pro nás užitečného textu i velké množství šumu (např. odkazy, obrázky). Bylo by velmi vhodné, kdyby se daly tyto data získat automaticky, protože pro člověka je to sice triviální úkol, ale s množstvím webových stránek zároveň i velmi časově náročný (spíše nemožný). Pro tento úkol byl navržen obecný čistící algoritmus jenž bude pracovat automaticky. U toho algoritmu se probere též pokrok oproti minulému roku a bude porovnán s již hotovým algoritmem.

2 Algoritmus obecného čištění webových stránek

Původní algoritmus uváděn v minulém roce Novotný (2015) byl razantně upraven a vylepšen. Klasifikační část algoritmu již nepracuje s CRF (podmíněné náhodné pole - Conditional Random field), které se v tomto typu úloh často používají. Tato část byla nahrazena lineárním SVM (Support Vector Machines) klasifikátorem, jenž využívá k popisu bloků textu ze stránek vektory příznaků. Tato změna byla provedena jednak kvůli velké náročnosti CRF algoritmu (hardwarové i časové), jenž bylo způsobeno velkým nárůstem vstupních dat a také kvůli možnosti porovnání s již zaběhnutým čistícím algoritmem BoilerPipe uvedeným prvně v Kohlsch C. (2010) (dále jen BP), jenž též používá Lineární SVM. Jako vstupní data bylo připraveno celkem 412 665 webových stránek získaných z webových serverů: ihned.cz, denik.cz, lidovky.cz, idnes.cz. Z toho trénovací množina obsahuje 330 127 stránek a testovací 82 538 stránek. Kvůli obrovskému množství testovacích dat nebyla referenční data (potřebná k ohodnocení) k těmto stránkám vytvořena manuálně ale za pomoci algoritmu na bázi pravidel (pro každý server jiná specifická pravidla). Algoritmus aktuálně pracuje v následujících krocích: příprava vstupních dat, trénování, testování (klasifikace) a ohodnocení.

2.1 Příprava vstupních dat

Jednotlivě se načítají webové stránky a za pomoci balíčku Beautiful Soup jsou pak následně vybrány bloky textu ohraničeny tagy $< p >$ a $< /p >$. Každý získaný blok je následně reprezentován vektorem příznaků a v případě trénovacích dat je též uvedeno zda se jedná o blok hledaného textu nebo o blok šumu (odkazy, atd.). Použité příznaky jsou např.: počet čísel v bloku, počet slov v bloku, pozice bloku na webové stránce a další.

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

2.2 Trénování a testování

Trénování i testování bylo provedeno za pomoci Lineárního SVM (scikit-learn balíčku), jenž využívá vektory příznaků. Testovací data byla vyčištěna jak zde navrženým algoritmem tak BP algoritmem pro možnost porovnání.

2.3 Ohodnocení

Ohodnocuje se jak výstup z algoritmu navrženého zde, tak výstup BP algoritmu. Díky tomu bude moci být provedeno porovnání kvality obou algoritmů. Ohodnocení je prováděno dvěma způsoby a to výpočtem F_1 míry a Levenshteinovy vzdálenosti. Oba způsoby porovnávají výsledné texty získané z výstupů algoritmů s referenčními texty.

2.4 Výsledky

V tabulce 1 jsou uvedené konečné výsledky. Hodnoty pro Levenshteinovu vzdálenost vyjadřují kolik slov muselo být průměrně změněno na jednu webovou stránku aby výstup algoritmů odpovídal referenčním datům. Zjednodušeně lze říci, že F_1 míra uvádí hodnotu úspěšnosti algoritmu v procentech.

	Levenshteinova vzdálenost		F_1 míra [%]	
	navržený algoritmus	BP algoritmus	navržený algoritmus	BP algoritmus
Výsledné hodnoty	58.62	78.97	87.2	85.4

Tabulka 1: Ohodnocení testovacích dat pro oba algoritmy

3 Závěr

Aktuální výsledné hodnoty viz. Tabulka 1 v porovnání s minulým rokem Novotný (2015) jsou o něco lepší a to nejen díky vylepšení algoritmu ale také kvůli množství použitých dat jak pro trénování tak pro testování (tedy lze říci, že výsledné hodnoty jsou přesnější). Je patrné, že oba algoritmy dosahují velmi dobrých výsledků. To že navržený algoritmus dosahuje lepších výsledků může být tím, že byl trénován na datech z uvedených webových serverů a testován též na datech z těchto serverů, zatímco BP algoritmus je trénován na neznámých datech a testován na jiných. Lze říci, že byl vytvořen obecný čistící algoritmus s dobrou úspěšností a tedy, že lze použít k vytvoření jazykových korpusů pro další použití v NLP. Samozřejmě je vždy co vylepšovat a po úpravách např. vektoru příznaků nebo po použití jiného klasifikátoru by se mohlo dosáhnout ještě lepších výsledků.

Poděkování

Příspěvek byl podpořen grantovým projektem SVK1-2016-023

Literatura

Novotný J., 2015., *Čištění zpravodajských webových stránek*, Sborník rozšířených abstraktů (str. 97-98), SVK (2015)

Christian Kohlsch, Peter Fankhauser, 2010, *Boilerplate Detection using Shallow Text Features*, L3S Research Center, Leibniz University Hannover, Germany, Wolfgang Nejdl, WSDN (2010)