Získavanie textových dát zo slovenského internetu

Daniel Hládek, Ján Staš

Technická univerzita v Košiciach
E-mail: daniel.hladek@tuke.sk, jan.stas@tuke.sk

Abstrakt

V tomto článku predstavujeme systém na automatický zber a prípravu webových korpusov písaných textov vhodných na trénovanie jazykového modelu. Prvou súčasťou je agent na prechádzanie a sťahovanie webových stránok. Z HTML kódu je extrahovaný text, ktorý je heuristicky zbavený nepodstatných častí. Trénovací text je uložený v databáze, odkiaľ ho môžeme na požiadanie získať. Extrahovaný text je indexovaný v databáze pre potreby vyhľadávania dokumentov pomocou kľúčových slov, dátumu získania dokumentu, domény alebo podľa iných kritérií. Prínosom článku je metóda ohodnotenia domény vzhľadom na počet stránok obsahujúcich unikátny text. Z každého odseku je vytvorený digitálny odtlačok, pomocou ktorého môžeme ľahko určiť podobnosť časti dokumentu v indexe. Kvalita dokumentu je vyjadrená počtom unikátnych odsekov. Vďaka ohodnoteniu skupiny dokumentov pochádzajúcich z jedného zdroja sa vie agent zacieliť na časti s väčšou pravdepodobnosťou výskytu použiteľného obsahu.
Kľúčové slová: získavanie dokumentov; prehľadávanie dokumentov; jazykové modelovanie; trénovací korpus

1 Úvod

Jazykový model je dôležitou súčasťou viacerých úloh v oblasti spracovania prirodzeného jazyka a automatického rozpoznávania reči. Vyjadruje pravdepodobnosť postupnosti slov na základe predošlého kontextu. Jazykový model môže byť použitý v úlohách zameraných na rozpoznávanie plynulej reči, strojový preklad, opravu preklepov, generovanie prirodzeného jazyka či jeho porozumenie.

Natrénovanie kvalitného jazykového modelu si vyžaduje mať k dispozícii veľké množstvo textových dokumentov. Parametre jazykového modelu sa potom počítajú z početností výskytu *n*-tíc (najčastejšie dvojíc a trojíc) slov v trénovacom korpuse. Čím väčší je počet rôznych postupností slov, tým je jazykový model presnejší (Staš et al., 2015). Jazykový model by mal byť čo najviac tematicky naviazaný na oblasť, v ktorej prebieha samotné spracovanie. To sa dá docieliť najmä vhodným delením trénovacieho korpusu do menších tematicky zameraných celkov. Okrem množstva textových dát, ktoré máme k dispozícii je dôležité, aby sme mali k dispozícii tiež najčastejšie sa vyskytujúce slovné spojenia, ktoré sú charakteristické pre cieľovú oblasť použitia. Je nutné sa zamerať najmä na metódy, ktoré sú schopné automaticky spracovať a klasifikovať čo možno najväčší objem textových dát dostupných v elektronickej podobe.

Je zrejmé, že množstvo textu v trénovacom korpuse presahuje schopnosti jedného pisateľa. Najvhodnejším spôsobom je vytvoriť systém na automatické získavanie a spracovanie textových dát dostupných na sieti Internet, tzn. vytvoriť agenta, ktorý postupne prechádza webové stránky a ich obsah ukladá do relačnej databázy (Rychlý, 2007).

Pri prechádzaní webových stránok sa môžu vyskytnúť tieto problémy:

- zahltenie zoznamu nenavštívených stránok s automaticky generovanými odkazmi;
- veľa odkazov na nepodstatný obsah;
- niektoré stránky sa často menia, a preto je potrebná stratégia opätovnej návštevy.

V tomto príspevku predstavujeme nami navrhnutý systém na autonómny zber a spracovanie textových dát zo slovenskej časti internetu. Agent na zber textu udržuje v pamäti zoznam odkazov, ktoré sa chystá navštíviť, je schopný sa zacieliť na také webové stránky, u ktorých predpokladáme väčšiu pravdepodobnosť výskytu relevantného obsahu, vie identifikovať také časti, ktoré nenesú informačný obsah a bráni sa príliš častej návšteve tej istej domény. S pomocou agenta na zber textových dokumentov na sieti Internet sme schopní získať dostatočné množstvo textu na natrénovanie jazykového modelu, ktorý môže byť použitý napr. v systémoch na automatické rozpoznávanie reči v slovenčine.

Teoretickým prínosom tohto článku je navrhnutie vlastnej metódy na ohodnotenie domény, ktorá vychádza z pravdepodobnosti výskytu informačného obsahu na webovej stránke. Domény, u ktorých je zistený vysoký výskyt relevantného obsahu je možné uprednostniť pri prehľadávaní pred tými, ktoré obsahujú veľa duplicitného obsahu, častí používateľského rozhrania alebo veľké množstvo reklám v textovej podobe. Tak vieme predísť zbytočnému prechádzaniu a zahlteniu databázy obsahom, ktorý nie je použiteľný pre potreby jazykového modelovania.

2 Aktuálny stav problematiky

Bolo vytvorené veľké množstvo špecializovaných alebo všeobecne použiteľných systémov určených na automatický zber textových dokumentov na sieti Internet. Všeobecné nástroje na zber, ako je napr. wget alebo curl, sú použiteľné na získavanie textových dokumentov z obmedzenej množiny webových stránok, no v niektorých prípadoch sú náchylné na zacyklenie sa v nekonečných slučkách. Na extrakciu informačného obsahu z rôznych dokumentov typu DOC, HTML, PDF, RTF, alebo ODF je možné využiť napr. nástroj Apache Tika. Rozlíšenie relevantných častí textu HTML stránok sa dá docieliť pomocou nástroja jusText alebo BoilerPipe. Prehľad súčasného stavu v oblasti návrhu stratégie prehľadávania siete hypertextových dokumentov je zhrnutý v článkoch (Brin a Page, 2012; Wang a Lee, 2011).

Pre úlohu indexovania textových dokumentov je najznámejším nástrojom „Lucene“, ktorý obsahuje moduly pre predspracovanie, indexovanie a porozumenie otázke. Od neho sú odvodené rôzne produkty, služby alebo knižnice, ako je napr. ElasticSearch, Apache SOLR, Clucene, Lucy a iné.

Dokumentovo-orientovaná databáza ElasticSearch je vhodná na uloženie a spracovanie tzv. „veľkých dát“. Pre tento nástroj existuje podpora slovenského jazyka na úrovni slovníka HunSpell na identifikáciu morfologických tvarov slovenských slov a preklepov. Výskum v oblasti morfologickej analýzy slovenského jazyka bol prezentovaný v našej predošlej práci (Hládek, 2012). Problematike vytvárania reverzných indexov sa venuje práca (Cambazoglu, 2013).

V oblasti budovania rozsiahlych webových korpusov písaných textov podobný prístup využíva iniciatíva Aranea (Benko, 2016). Takto vytvorený webový korpus slovenského jazyka sa

používa najmä v oblasti korpusovej lingvistiky. Na získavanie textu využíva systém SpiderLing a texty sú spravované pomocou databázy Manatee-Bonito (Rychlý, 2007).

3 Získavanie textu

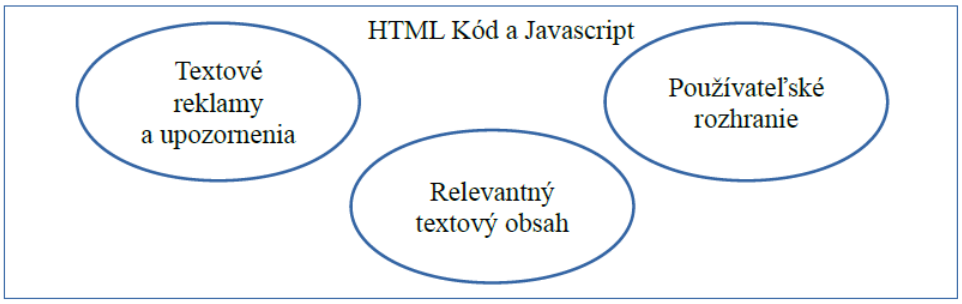
Na sieti Internet sa nachádza veľké množstvo textového obsahu v rôznych jazykoch a v rôznej kvalite. Navyše, informačný obsah v podobe textu je zastúpený nerovnomerne – na niektorých doménach sa nachádza viac textu ako na iných. Niektoré webové stránky, napr. databázy obrázkov, nemá zmysel prehľadávať, lebo sa tam nebudú nachádzať väčšie časti obsahujúce súvislý text, ale najmä časti používateľského rozhrania a reklám.

Úlohou agenta na zber textových dát je získanie čo najväčšieho množstva unikátneho textu pri minimálnom počte návštev Internetových domén. Doménou rozumieme skupinu dokumentov nachádzajúcich sa na jednom zdroji, venovaných jednej problematike a určených pre konkrétnu skupinu prijímateľov. Dokumenty obsiahnuté v jednej doméne majú spravidla pridelené to isté doménové meno v systéme DNS (domain name service), no nemusí to byť vždy tak.

Agent pri zbere postupne prehľadáva domény, analyzuje ich informačný obsah a relevantný text ukladá do databázy. Je potrebné, aby bol proces prehľadávania zacielený na miesta, kde je väčšia pravdepodobnosť výskytu „dobrého textu“ - unikátnych odsekov písaných v slovenskom jazyku. Náš prístup rieši všetky tieto problémy a umožňuje autonómny beh, počas ktorého agent prehľadáva webové stránky a získaný text postupne ukladá do databázy, spolu s informáciou o extrahovanom dokumente vo forme metadát.

Všeobecný algoritmus autonómneho získavania dokumentov by sa dal opísať v nasledujúcich bodoch:

- na začiatku máme k dispozícii počiatočný zoznam odkazov na webové stránky;
- získaj a spracuj dokumenty zo zoznamu;
- extrahuj odkazy zo získaných dokumentov;
- ulož nové odkazy do zoznamu získaných odkazov, aby nastala deduplikácia;
- vyber ešte nenavštívené odkazy zo zoznamu a pokračuj.



Obrázok 1: Obsah webovej stránky.

3.1 Analýza obsahu dokumentu

Pri získavaní textových dát je nutné prekonať nasledujúce technické prekážky:

- výskyt duplicitných, alebo takmer duplicitných častí textu;
- výskyt nepodstatných častí textu na webovej stránke, ako je používateľské rozhranie

alebo textové reklamy;
- výskyt častí textov písaných v inom jazyku.

Obsah webovej stránky je znázornený na Obr. 1. V kóde HTML sa zvyčajne spolu s relevantným obsahom nachádzajú časti používateľského rozhrania a textové reklamy. Značky HTML a JavaScript hovoria o tom, akým spôsobom sa text a obrázky na webovej stránke zobrazia. Tieto časti je potrebné odfiltrovať.

Základnou jednotkou objemu textu, ktorú uvažujeme pri analýze dokumentu je odsek, resp. paragraf. Získanie jedného dokumentu pozostáva zo stiahnutia kódu HTML zo zadaného odkazu a extrakcie odsekov, ktoré sa v ňom nachádzajú. Na to, aby sme vedeli odlíšiť podstatné a nepodstatné časti webovej stránky je nutné vykonať podrobnú analýzu obsahu. Čistý text, rozdelený na odseky, získame pomocou parsera HTML.

Odsek okrem použiteľného textu môže obsahovať tiež:

- textovú reklamu;
- časť textového rozhrania;
- duplicitný text.

Odseky, ktoré neobsahujú relevantné informácie nie je potrebné ukladať do databázy. Navyše, ak vieme, ktorý odsek nesie informačný obsah a ktorý nie, tak vieme stanoviť koeficient užitočnosti pre dokument a celú doménu (skupiny dokumentov z jedného zdroja).

O užitočnosti odseku je takmer nemožné rozhodnúť s úplnou presnosťou. Namiesto toho používame heuristicky postup, ktorý by sa dal opísať nasledovne:

V prvom kroku rozhodneme o tom, či je text súčasťou textového rozhrania. K tomu používame knižnicu jusText (Pomikálek, 2011), ktorá obsahuje postupy pre rozlíšenie častí používateľského rozhrania pomocou analýzy kódu HTML.

V druhom kroku rozhodujeme o unikátnosti odseku. Ak sa taký istý alebo veľmi podobný odsek už v databáze vyskytuje, môže to znamenať, že nebude prínosom pre výsledný textový korpus. O unikátnosti odseku rozhodujeme jeho vyhľadaním v databáze.

3.2 Kontrolný súčet odseku dokumentu

V naivnom algoritme na overenie unikátnosti odseku je potrebné vyhľadať v databáze, či sa v nej náhodou nevyskytuje rovnaký paragraf. Tento prístup je časovo a priestorovo náročný. Ak chceme vyhľadávať odsek bez pomoci indexu, tak je potrebné postupne prechádzať všetky položky v databáze. Využitie indexu by prehľadávanie zrýchlilo, avšak za cenu veľkej spotreby pamäti. Tento prístup nie je tiež celkom vhodný v prípade, ak chceme skladovať veľké množstvo dát.

Ako kompromis medzi týmito dvoma prístupmi sme zvolili techniku, ktorá sa podobá vytváraniu kontrolného súčtu. Štandardný prístup na výpočet kontrolného súčtu znakov v reťazci (označovaný ako hashovacia funkcia) nie je ale celkom vhodný, pretože nevyjadruje vlastnosti podobnosti odsekov textu tak, ako ich vníma človek. Napríklad biele znaky, diakritika, či interpunkčné znamienka vytvárajú veľké rozdiely v kontrolnom súčte, no z pohľadu informačnej hodnoty nenesú žiaden význam.

Namiesto bežne používaného kontrolného súčtu vo forme hashovacej funkcie sme navrhli tzv. digitálny odtlačok paragrafu, ktorý v skrátenej podobe vyjadruje len také významové znaky,

ktoré sú v odseku viditeľné.

Kontrolný súčet paragrafu môžeme vyjadriť pomocou nasledovného pseudokódu:

```
VSTUP: odsek, ktorý sa skladá zo znakov
VÝSTUP: celé číslo súčet v rozsahu od 0 až po 2^32 - 1
súčet = 0
PRE KAŽDÝ znak V odseku:
    AK JE znak VIDITEĽNÝ:
        súčet += (súčet << 3) + znak
        súčet += (súčet >> 31)
```

Operácia >> a << je bitový posun doľava, resp. doprava a je ekvivalentný celočíselnému násobeniu alebo deleniu mocninou dvojky. Hlavný rozdiel od bežne použíanej hashovacej funkcie je, že algoritmus neberieme do úvahy neviditeľné znaky, ako je medzera alebo koniec riadku, pretože pridaním ďalšej medzery sa nemení obsah odseku.

Číslo s kontrolným súčtom paragrafu a jeho veľkosťou sa uloží do databázy. Po spočítaní kontrolného súčtu paragrafov je ľahko možné overiť mieru unikátnosti dokumentu, a tým aj celej domény.

3.3 Hodnotenie kvality domény

Doména je ohodnotená pomocou skóre, ktoré vyjadruje pravdepodobnosť získania dobrého textu. „Kvalitu domény“ vie agent zohľadniť pri stratégii návštev a uprednostniť domény s vyššou pravdepodobnosťou získania dobrého textu. Hodnotenie i-teho textového dokumentu sa vypočíta ako pomer počtu dobrých odsekov U_i ku všetkým odsekom O_i nachádzajúcim sa v dokumente. Spolu s kontrolnými súčtami odsekov v dokumente ukladáme aj počet znakov obsiahnutých v odseku, pre ktoré sa vypočítava kontrolný súčet. To nám umožňuje ľahko vypočítať koľko znakov v dokumente patrí unikátnym odsekom a koľko z nich patrí odsekom, ktoré sú duplicitné .Ohodnotenie celej domény F s počtom dokumentov N je ich aritmetickým priemerom, definovaným podľa vzťahu:

$$F = \frac{1}{N} \sum_{i}^{N} \frac{U_i}{O_i}$$

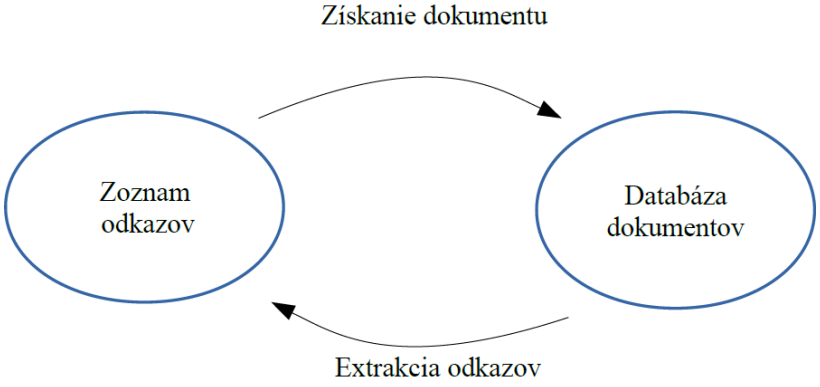
Výsledkom získania jednej domény URL je zoznam odsekov, hodnotenie dokumentu a zoznam extrahovaných odkazov pre ďalšie prehľadávanie. Doména, ktorá obsahuje veľké množstvo častí používateľského rozhrania alebo často opakujúce sa textové reklamy má pridelené nižšie skóre, ako doména obsahujúca väčší objem unikátneho textu.

Celý proces hodnotenia vieme vyjadriť pomocou pseudokódu:

```
VSTUP: URL, databáza odtlačkov
VÝSTUP: text a ohodnotenie unikátnosti
html obsah = STIAHNI(url)
odseky  a odkazy = SPRACUJ(obsah)
počet dobrých odsekov = 0
počet zlých odsekov = 0
PRE KAŽDÝ odsek V odsekoch:
    odtlačok = ODTLAČ(odsek)
```

```
        AK odtlačok SA NACHÁDZA :
            počet zlých odsekov += 1
        INAK:
            ULOŽ(odtlačok)
            počet dobrých odsekov += 1
    VRÁŤ odseky a odkazy, počet dobrých a zlých odsekov
```

4 Stratégia prehľadávania



Obrázok 2: Autonómne získavanie textu z Internetových stránok.

Základnú stratégiu prehľadávania podľa metódy výberu delíme na:

- náhodný výber s uniformnou distribúciou;
- náhodný výber podľa kvality domény;
- cielený výber na vybrané domény.

Najpodstatnejšou časťou je stratégia výberu ďalších odkazov na prehľadávanie. Stratégia, pri ktorej sa odkazy na návštevu vyberajú úplne náhodne (slepým výberom) sa neukázala ako vhodná. Pri nevhodne zvolenom začiatočnom odkaze dochádzalo k zahlteniu databázy veľkým množstvom odkazov na nepodstatný alebo duplicitný obsah. Redakčné systémy často odkazujú náhodnými odkazmi na dynamicky generované stránky, ktoré obsahujú menej podstatné informácie. Pri prehľadávaní je preto vhodne sa sústrediť len na miesta, kde je vysoká pravdepodobnosť výskytu informačného obsahu. Na druhej stane, prílišné zameranie sa iba na overené zdroje môže byť kontraproduktívne, pretože sa zdroje môžu ľahko vyčerpať (všetky ich texty budú po čase indexované v databáze). Stratégia výberu odkazov na prehľadávanie by mala uprednostňovať kvalitné domény pred nekvalitnými, avšak nemala by sa vyhýbať neznámym alebo menej kvalitným zdrojom.

Na začiatku procesu získavania dokumentov je výber ešte nenavštívených odkazov. Skupina odkazov sa zoradí podľa zdroja, z ktorého pochádza a ak je to možné, tak sa každému zdroju priradí hodnotenie. Ak je zdroj dostatočne dobrý alebo ešte nemá hodnotenie, prebehne stiahnutie webových stránok z každého zdroja a vypočíta sa ich unikátnosť. Na konci sa upraví hodnotenie celého zdroja podľa novozískaných webových stránok. Výsledkom je to, že agent sa po čase vie vyhnúť doménam s málo unikátnym textom.

Algoritmus na autonómny beh agenta vieme vyjadriť pomocou nasledovného pseudokódu:

```
VSTUP: zoznam url
domény = ZORAĎ PODĽA DOMÉNY (zoznam url)
PRE KAŽDÚ url V KAŽDEJ doméne:
    hodnotenie = PREČITAJ HODNOTENIE(doména)
    AK hodnotenie NIE JE ZLÉ:
        ZÍSKAJ všetky url a ich unikátnosť
        ULOŽ HODNOTENIE (doména)
        ULOŽ ZOZNAM (nové url z domény)
    INAK POKRAČUJ
```

5 Databáza textu

Dokumenty sú v databáze reprezentované vo vektorovom priestore. Algoritmy nekontrolovaného učenia sú schopné odhaliť sémanticky podobné skupiny dokumentov. Identifikácia sémanticky podobných dokumentov v databáze umožňuje presnejšie vyhľadávanie vo veľkých dátach na základe významovej podobnosti dokumentu s dopytom. Množina významovo podobných dokumentov potom reprezentuje znalosť o určitej doméne a môže byť využitá napríklad pri tvorbe robustných jazykových modelov adaptovaných do zvolenej oblasti používania.

V oblasti automatického rozpoznávania reči takýto spôsob modelovania jazyka prináša výrazné zlepšenie presnosti rozpoznávania reči. Jedná z nami navrhnutých metód zhlukovania dokumentov v korpuse písaných textov obsahujúceho veľké množstvo tém je opísaná v práci (Staš, 2014). Výsledky štatistického modelovania slovenského jazyka založeného na textových dátach získaných z webových stránok sú zhrnuté tiež v práci (Staš, 2015).

5.1 Metainformácie o dokumente

Text je v databáze uložený spolu s informáciami o dokumente. Texty sú následne v databáze indexované, vďaka čomu je možné v nich ľahko vyhľadávať pomocou kľúčových slov. V databáze sú uchovávané okrem extrahovaného textu aj nasledovné informácie:

- dátum návštevy;
- URL;
- informácie z hlavičky dokumentu;
- kľúčové slovíčka;
- autor;
- najskorší dátum získaný z textu;
- pôvodné HTML.

5.2 Veľkosť databázy

Veľkosť databázy textu získanej v období rokov 2011 až 2018 je zhrnutá v Tabuľke 1.

celkový počet domén	564 598
celkový počet dokumentov	110 649 081
celkový počet tokenov	3 795 840 362
celkové množstvo v bajtoch	27 191 594 154

Tabuľka 1: Objem dát získaných zo slovenského Internetu.

5.3 Úprava textov

Získané texty musia následne prejsť procesom prvotného spracovania do podoby vhodnej na trénovanie jazykových modelov. Procesu spracovania textových dokumentov sa podrobnejšie venujeme v publikácii (Hládek a Staš, 2010) a zahŕňa najmä tokenizáciu textu, určovanie hraníc viet, prepis čísloviek, symbolov a skratiek do vyslovovanej podoby, morfologickú anotáciu a pod. Čo sa týka morfologickej anotácie, pre tento účel sme vyvinuli tiež vlastný morfologický analyzátor Dagger (Hládek a Staš, 2012).

6 Záver

V príspevku prezentujeme komplexný systém na získavanie, spracovanie a ukladanie textových dát zo slovenského Internetu. Agent na zber textu je autonómny, je schopný dlhodobo získavať veľké množstvo dát bez akéhokoľvek dozoru. Samotný proces získavania textových dát je časovo náročný proces. V priebehu siedmych rokov behu agenta sa nám podarilo vytvoriť jednu z najväčších databáz písaných textov získaných z Internetových stránok. Táto databáza je využívaná nielen v oblasti trénovania jazykových modelov pre systémy na rozpoznávanie plynulej reči (Rusko et. al., 2016), ale aj v rôznych iných úlohách zameraných na spracovanie prirodzeného jazyka. Časť nami získaných textov je aj súčasťou korpusu Aranea (Benko, 2016) a korpusu (Hládek, 2016).

7 Bibliografia

Benko, V. (2016). Two years of Aranea: Increasing counts and tuning the pipeline. In *Proc. of LREC 2016,* Portorož, Slovenia, pp. 4245-4248.

Brin, S., and Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 56(18): 3825-3833.

Cambazoglu, B.B., et al. (2013). A term-based inverted index partitioning model for efficient distributed query processing. *ACM Transactions on the Web*, 7(3): 15.

Hládek, D. a Staš, J. (2010). Text mining and processing for corpora creation in Slovak language. *Journal of Computer Science and Control Systems*, 3(1): 65-68.

Hládek, D., Staš, J. and Juhár, J. (2012). Dagger: The Slovak morphological classifier. In *Proc. of ELMAR 2012*, Zadar, Croatia, pp. 195-198.

Hládek, D., Staš, J. and Juhár, J. (2016). Evaluation set for Slovak news information retrieval. In *Proc. of LREC 2016*, Reykjavik, Island, pp. 1913-1916.

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., and Suchomel, V. (2013). The TenTen corpus family. In *Proc. of the 7th International Corpus Linguistics Conference, CL 2013,* Lancaster, UK, pp. 125-127.

Pomikálek, J. (2011). Removing boilerplate and duplicate content from web corpora. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno, Czech Republic.

Rychlý, P. (2007). Manatee/Bonito – A modular corpus manager. In *Proc. of 1st Workshop on Recent Advances in Slavonic Natural Language Processing*, Masaryk University, Brno, pp. 65–70.

Rusko M. et al. (2016). Advances in the Slovak judicial domain dictation system. In Vetulani, Z., Uszkoreit, H., Kubis, M. (eds): *Human Language Technology. Challenges for Computer Science and Linguistics, LTC 2013*, LNCS 9561, Springer, Cham, pp. 55-67.

Staš, J., Juhár, J., and Hládek, D. (2014). Classification of heterogeneous text data for robust domain-specific language modeling. *EURASIP Journal on Audio, Speech, and Music Processing*, 2014(14): 1-14.

Staš J., Hládek, D., and Juhár, J. (2015). Language model speaker adaptation for transcription of Slovak parliament proceedings. In Ronzhin, A., Potapova, R., Fakotakis, N. (eds): *Speech and Computer. SPECOM 2015*, LNCS 9319, Springer, Cham, pp. 259-267.

Wang, Y.-T. and Lee, A.J.T. (2011). Mining web navigation patterns with a path traversal graph. *Expert Systems with Applications*, 38(6): 7112-7122.

Poďakovanie

Táto práca vznikla vďaka podpore Vedeckej grantovej agentúry realizáciou výskumného projektu VEGA 1/0511/17 a vďaka podpore Agentúry na podporu výskumu a vývoja realizáciou projektu aplikovaného výskumu APVV-15-0517, financovaných z prostriedkov Ministerstva školstva, vedy, výskumu a športu SR.