KA 3

Metodika postupu shlukové analýzy

Tento výstup byl vytvořen v rámci realizace projektu *Zefektivnění činnosti Technologické agentury ČR v oblasti podpory výzkumu, vývoje a inovací a podpora posilování odborných kapacit organizací veřejné správy v oblasti VaVaI* (reg. č. CZ.1.04/4.1.00/D4.00003), spolufinancovaného z Operačního programu lidské zdroje a zaměstnanost a státního rozpočtu

Autoři dokumentu: Jaroslava Hlaváčová a další

ISBN 978-80-88169-14-7

Obsah

Seznam grafů

Seznam tabulek

Manažerské shrnutí

Cílem materiálu bylo najít mezi zadanými dokumenty z různých zdrojů podobnosti, které by pomohly odhalit případné duplicity nebo značné vzájemně překryvy v projektových dokumentacích.

Zdroje, jejichž data byla zpracována, pocházejí z databází Informačního systému výzkumu, experimentálního vývoje a inovací, databáze CORDIS[1] (veřejný repozitář a portál pro informace o výzkumných projektech financovaných Evropskou unií; z angl. Community Research and Development Information Servis), z projektových dokumentů Technologické agentury České republiky (dále TA ČR) a z výběru z databáze údajů o přihláškách patentů a patentech PATSTAT[2].

V průběhu realizace byla snaha rozšířit zdroje dat i o dokumenty stažené přímo z webových stránek vědeckých a výzkumných organizací. Tento směr získávání dat se však ukázal jako neperspektivní z důvodu nemožnosti navržení nástroje, který by byl schopen v určeném časovém úseku a se stanovenou mírou spolehlivostí automaticky rozpoznat, které webové stránky jsou relevantní (tj. pojednávají o daném vědecko-výzkumném tématu) a které nekoliv.

Všechny dokumenty z výše uvedených zdrojů (cca 1 mil. textových dokumentů) byly podrobeny následujícímu zpracování:

1. čištění dat a jejich převod do jednotného formátu,
2. kontrola jazyka - zjištění, v jakém jazyce jsou textové položky,
3. překlad českých textů do angličtiny,
4. shluková analýza přeložených dokumentů.

Výsledkem je stromová struktura, tzv. dendrogram, který reprezentuje podobnosti mezi jednotlivými dokumenty. Vizualizace celého dendrogramu nemá smysl vzhledem k jeho velikosti. Z tohoto důvodu byl implementován jednoduchý nástroj, kterým lze strom procházet a sledovat jeho části lokálně. Nástroj umožňuje rovněž vyhledat určitý dokument a podívat se, jaké dokumenty jsou mu podobné, a tak vyhledávat případné duplicity k vybraným dokumentům. Popsané výsledky byly umístěny na stroji cluster v TA ČR a lze je i nadále využívat.

Do budoucnosti bude nezbytné data aktualizovat a zpracovat vždy znovu výše uvedeným způsobem (viz body 1 až 4). Všechny kroky byly automatizovány, ale vzhledem k velkému množství dat je nutné počítat, že jejich zpracování trvá dlouho. Nejdelší je výpočet dendrogramu (bod 4), který trvá cca 2 až 4 dny. Také automatický překlad velkého množství dokumentů trvá řádově dny. Není tedy možné provádět aktualizace každý den. Při rutinním použití se doporučuje aktualizovat data tehdy, když

[1] http://cordis.europa.eu/

[2] http://www.epo.org/searching-for-patents/business/patstat.html#tab1

přibude větší počet dokumentů přímo v TA ČR (např. při ukončení podávání návrhů do programu/výzvy TA ČR), a potom např. jednou za 2 měsíce kvůli průběžné aktualizaci dat v ostatních databázích, především RIV[3].

[3] Rejstřík informací o výsledcích

1. Úvod

Dokument byl vypracován jako výstup projektu *Zefektivnění činnosti TA ČR v oblasti podpory VaVaI a podpory posilování odborných kapacit organizací veřejné správy v oblasti VaVaI* (dále Projekt Zefektivnění činností TA ČR), resp. v rámci jeho klíčové aktivity 3 (dále KA 3) - Příprava nových analytických metodik hodnocení VaVaI. Projekt byl financován z Evropského sociálního fondu v rámci Operačního program Lidské zdroje a zaměstnanost.

Hlavním cílem projektu je zefektivnění poskytování podpory výzkumu, experimentálního vývoje a inovací (dále VaVaI) ze strany TA ČR a dalších organizací veřejné správy, posílení odborných kapacit organizací veřejné správy v oblasti VaVaI, posílení chápání významu aplikovaného VaVaI a jeho výsledků pro další rozvoj České republiky a do budoucna i sjednocení způsobů a podmínek poskytování podpory VaVaI.

Cíle Projektu Zefektivnění činností TA ČR jsou naplňovány sedmi klíčovými aktivitami. KA 3 - Příprava nových analytických metodik hodnocení VaVaI je zaměřena na podporu posilování odborných kapacit organizací veřejné správy v oblasti VaVaI. Cílem aktivity je zefektivnit a zkvalitnit činnosti TA ČR v oblasti analytických služeb jako základní metody zdůvodňování tvorby podkladů pro nastavení a hodnocení podpor. Toho má být dosaženo vytvořením nových analytických metodik pro podporu zajišťování hlavních činností a procesů realizovaných TA ČR.

Tento dokument byl vypracován jako doprovodný materiál ke Studii proveditelnosti pro implementaci modelu k hodnocení nepřímých dopadů výzkumu, experimentálního vývoje a inovací na základě principů modelu StarMetrics™, která byla hlavním výstupem projektu. Slouží jako jeden z podkladových materiálů pro implementaci kvalitnějších analytických služeb v TA ČR směrem k Evidence Based Policy, tj. principu činit informovaná rozhodnutí o tvorbě veřejných politik či programů na základě dostupných důkazů. Byl vypracován na základě analytických zjištění a zkušeností získaných při realizaci KA 3 a to v období říjen 2014 až listopad 2015. Shlukovou analýzu realizoval, pilotně ověřoval a metodicky zaznamenal pracovní tým vytvořený na Matematicko-fyzikální fakultě Univerzity Karlovy v Praze (dále MFF).

Výsledkem je zařazení dokumentů do shluků podle „podobnosti“ jejich textových částí. V ideálním případě by se do jednoho shluku měly dostat dokumenty týkající se jednoho projektu, případně více projektů, které spolu mají nějaký vztah. Souvislost, nebo podobnost, je zjišťována pouze z textů, nikoli z metadat. Znamená to, že se pro shlukování nevyužívají žádné informace, které dokumenty popisují. Tedy žádná jména, instituce, identifikační čísla apod. Snahou této aktivity bylo ověřit hypotézu, zda by podobnost nebylo možné zjistit i jinak, konkrétně pomocí jazykového rozboru textů.

Cílem je odhalit souvislosti, které nejsou zřejmé na první pohled, nebo které se dokonce někdo snaží záměrně zastřít. Jazyková analýza by měla pomoci např. odhalit žadatele s projekty, jejichž cílem je výzkum v oblasti již zkoumané nebo které se řeší v jiném projektu.

Při zpracování se vycházelo z předpokladu, že projekty zabývající se stejným tématem jsou popsány vždy obdobně (především použitím totožné terminologie). Odborné termíny nemívají často synonyma, jako je tomu v běžném jazyce a proto se konkrétní témata obtížně popisují „jinými slovy“. Také jazykový styl bývá u příbuzných témat podobný.

Podaří-li se sdružit jazykově podobné dokumenty do společných shluků, lze získat další informace pro zpracování analýzy souvislostí.

Současně je třeba upozornit, že veškeré metody, které byly použity, jsou založeny na statistickém přístupu. Jedná se o nástroje, které upozorní na možné neobvyklé vztahy mezi dokumenty, a vždy je potřeba tyto vztahy podrobit “ruční” analýze. V žádném případě by se dle výsledků těchto automatických statistických metod neměly dělat rychlé, natož automatické závěry.

Rozsah dokumentů, které byly původně zamýšleny pro vstup do shlukové analýzy, bylo takové množství, že spočítání shluků by trvalo neúměrně dlouhou dobu (řádově až roky). Z uvedeného důvodu byla úloha rozdělena do dvou částí:

1. Vlastní shluková analýza

Shluková analýza byla provedena jen na části dokumentů, a to těch nejpodstatnějších. Jde konkrétně o dokumenty z následujících zdrojů (v závorce je uveden počet dokumentů, na kterých byla shluková analýza provedena před koncem projektu, tedy v listopadu 2015):

- Dokumenty o projektech TA ČR, tedy databáze Informačních systémů Patriot a BETA (celkem 3 883).
- Informační systém výzkumu, experimentálního vývoje a inovací (CEP[4]) – seznam projektů (celkem 41 581).
- Informační systém výzkumu, experimentálního vývoje a inovací (CEZ[5]) – seznam výzkumných záměrů (celkem 886).
- Informační systém výzkumu, experimentálního vývoje a inovací (RIV[6]) – seznam výsledků VaVaI od roku 1992 (celkem 772 890).
- Databáze CORDIS - výběr projektů, u kterých aspoň jeden partner byl z České republiky (celkem 3 428).
- Databáze PATSTAT - výběr patentů od roku 1992 (celkem 20 696).

Dokumenty z tohoto vzorku budou dále v textu nazývány jako podstatné. Podstatných textů bylo vybráno cca 1 mil. Jedná se tedy o množství, které je technicky realizovatelné

[4] CEP = Centrální evidence projektů VaVaI

[5] CEZ = Centrální evidence výzkumných záměrů

[6] RIV = Rejstřík informací o výsledcích VaVaI

v přiměřeném čase. Vypracování shlukové analýzy na serveru cluster trval s uvedenými objemy dat 2 až 4 dny.

Výsledkem shlukové analýzy bylo rozdělení podstatných dokumentů do shluků. Shluky se pomocí speciálního programu tvořily hierarchicky. Na počátku tvořil každý dokument jeden (triviální) shluk. Maximální počet shluků odpovídá celkovému počtu posuzovaných dokumentů. Následně se postupně sdružují dokumenty, které k sobě mají dle stanoveného kritéria blízko. Shlukování probíhá do té doby, dokud nejsou všechny dokumenty sdruženy do jednoho velkého shluku.

Výsledkem shlukové analýzy je tedy graf tvaru stromu (dendrogram), jehož jednotlivé větve tvoří shluky. Uživatel si může sám zvolit, na jaké hladině podrobnosti chce shluky zkoumat a jak mají být shluky velké, tj. jak moc se výsledné větvičky ještě budou dělit na menší a menší.

2. Začleňování dalších dokumentů, tzv. roubování

Tato část projektu řešila následující hypotézu:

Existuje předpoklad, že byla zpracována shluková analýza na podstatných textech. Jejím výsledkem je strom popsaný výše. V případě nového dokumentu je nutné zjistit, do jakého shluku by měl být zařazen v případě, kdyby byl součástí tvorby shlukové analýzy.

První variantou je zpracovat novou shlukovou analýzu s novým dokumentem a postavit nový strom, který ho bude obsahovat. Tato varianta je však časově velmi náročná (při větším počtu dokumentů zcela neproveditelná).

Druhou variantou je tzv. roubování, neboli kvalifikovaným odhadem. Jeho výsledkem je opět dendrogram, který se od původního liší pouze přičtením jednoho koncového uzlu zastupujícího nový dokument.

2. Vstupní data a jejich příprava

Jak již bylo uvedeno v kapitole 1, dostupná data byla rozdělena do dvou skupin. První skupina obsahovala tzv. **podstatná data**, která obsahovala popisy projektů a výsledků VaVaI v České republice. Tato data byla použita pro tvorbu vlastní shlukové analýzy. Především se jednalo o data TA ČR, která se týkají projektů jednotlivých programů.

Dalším důležitým zdrojem byly databáze Informačního systému výzkumu, experimentálního vývoje a inovací:

- RIV,
- CEP,
- CEZ.

Jako texty sloužily téměř výhradně abstrakty výsledků a anotace projektů. Jiná textová data buď nebyla k dispozici, nebo je lze nalézt jen v nevyhovujícím formátu, především formátu .pdf. Podstatné informace o výsledcích vědeckého výzkumu nebo projektu by však měly být obsaženy v abstraktu. Mezi textová data se počítá i název projektu či výsledku a klíčová slova, pokud se v dokumentu vyskytují. Vzhledem k tomu, že databáze RIV, zvláště její starší části, není v mnoha případech řádně vyplněna, jako zdroj dat byly použity pouze takové dokumenty, které byly dostatečně dlouhé. Z dalšího zpracování byly vyřazeny dokumenty, které byly příliš krátké, což se projevilo jedním z násl. způsobů:

- měly pouze title (titleEN), tzn. ostatní textové položky byly prázdné,
- obsahovaly pouze keywords (keywordsEN), tzn. ostatní textové položky byly prázdné,
- abstract (abstractEN) byl kratší než 10 slov,[7]
- annotation (annotationEN) byla kratší než 10 slov.

Rovněž byly vyloučeny dokumenty obsahující na místě textové položky výrazy typu “Not available”.

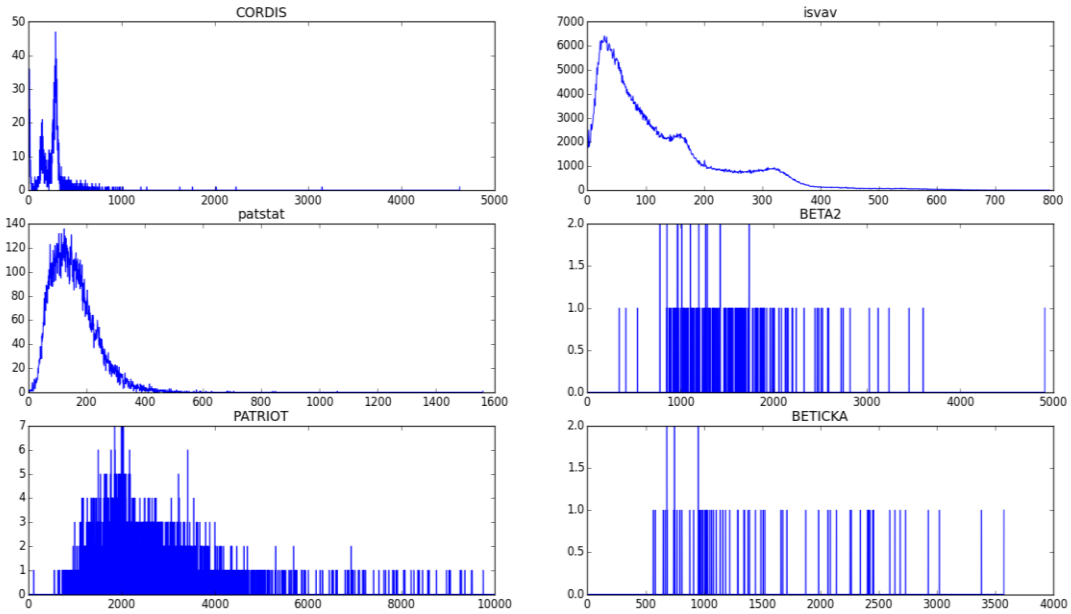
Třetím důležitým zdrojem podstatných textů byla databáze CORDIS. Z ní byly vybrány všechny projekty, kterých se účastnil jako partner zástupce z České republiky.

Posledním zdrojem podstatných textů byla databáze PATSTAT. Databáze obsahuje údaje o cca 70 mil. patentů. Objem dat je však příliš velký. Proto došlo k rozhodnutí zpracovat pouze její část, která byla v kapacitách výpočetně zvládnutelná. Z celé databáze byla proto extrahována data od roku 1992 dosud. Rok 1992 byl zvolen z důvodu vzniku RIV. Jedná se tedy o časovou paralelu s ostatními daty. Podobně jako data z RIV, i v případě databáze PATSTAT bylo nutné data vyčistit, naboť byly často textové položky prázdné.

[7] Při budoucím spouštění shlukové analýzy by se tento limit mohl zvýšit, např. na 20 slov.

Následující obrázek obsahuje několik grafů, které ukazují rozložení dokumentů v závislosti na jejich délce. Na vodorovné ose je uvedena délka dokumentu ve slovech, na svislé ose jejich počet (vše pro jednotlivé podstatné texty). Z grafů je patrné, že vždy převažují spíše kratší dokumenty, ale v závislosti na zdroji se jejich délka liší. Nejvíce velmi krátkých, a z tohoto důvodu i nepoužitelných dokumentů, obsahují data z Informačního systému výzkumu, experimentálního vývoje a inovací.

Graf 1: Rozložení dokumentů v závislosti na jejich délce



Poznámka: Beticka - data získaná z 1. soutěže Programu veřejných zakázek ve výzkumu, experimentálním vývoji a inovacích pro potřeby státní správy Beta (usnesení vlády č. 54 ze dne 19. ledna 2011 změněný usnesením vlády č. 75 ze dne 31. ledna 2013); Beta 2 - data získaná z 2. soutěže programu Programu veřejných zakázek ve výzkumu, experimentálním vývoji a inovacích pro potřeby státní správy Beta (usnesení vlády č. 54 ze dne 19. ledna 2011 změněný usnesením vlády č. 75 ze dne 31. ledna 2013)

Druhou skupinu tvořily **ostatní zdroje dat**. Tato data lze použít pro zjišťování podobnosti s daty z první skupiny (roubování).

Jedná se o následující zdroje:

- články z Národní technické knihovny,
- veřejné zdroje z internetu.

Tato data nebyla použita přímo pro shlukovou analýzu, neboť jejich objem je tak rozsáhlý, že by dostupné algoritmy nebyly schopny vše spočítat v reálném čase. Proto byla pro zjišťování podobnosti zvolena metoda roubování. Roubovat lze rovněž jednotlivé dokumenty, u nichž se objeví nutnost najít podobnosti s ostatními dokumenty.

Národní technická knihovna

Objem článků v Národní technické knihovně je velký a Národní technická knihovna neklade žádná omezení na jejich využití. Problém však spočívá v jednotlivých nakladatelstvích, od kterých knihovna data přebírá, neboť používají různé formáty a tudíž je třeba pro každý balík článků napsat speciální program na extrakci pouze textových dat. Některé formáty nejsou jednoduše převeditelné do čistého textu, např. formát .pdf. V průběhu realizace projektu bylo vyzkoušeno několik nástrojů, ale žádný z nich nefungoval zcela bez problému. Nejvíce se osvědčil program pdf2txt.

Data z Národní technické knihovny proto nebyla použita.

2.1. Veřejné zdroje z internetu

Předpokládalo se, že veřejně dostupné zdroje z internetu poskytnou doplňková data, která nemusí být součástí oficiálních dokumentů o projektech a jejich výsledcích. Původní představa byla identifikovat webové stránky vědecko-výzkumných organizací, jejich složek a případně i jednotlivých vědecko-výzkumných pracovníků a vyfiltrovat z nich textové údaje o jejich odborných profilech. V průběhu realizace se však tento předpoklad nepotvrdil, neboť tento cíl byl nereálný.

Vědecko-výzkumných institucí je velké množství (např. počet institucí evidovaných v Informačním systému výzkumu, experimentálního vývoje a inovací je 5 745), ale hlavním problémem je jejich „košatost“ a rozmanitost struktur. Prakticky každá instituce má vlastní formu webových stránek a častokrát se na nich ani vědecké články nevyskytují. Existují situace, kdy jsou seznamy článků schované za formulářem „s ochranou proti robotům", takže s nimi nelze pracovat automaticky. Dalším problémem je provázanost dat a množství domén, které se váží k jedné instituci. Rozpoznání již navštívené webové stránky je snadné. Není však snadné rozpoznat stránky se (skoro) stejným obsahem. A rovněž není lehké identifikovat soubory, které jsou potřebné (viz dále). I přes uvedené negativní stránky bylo stahování veřejných zdrojů věnováno značné úsilí.

Byly staženy informace z webových stránek univerzit a jednotlivých ústavů Akademie věd České republiky. K tomu byl využit program httrack. Dále byl rovněž využit program justext k pročištění html kódu na prostý text.

Automatické stahování textových dat bylo časově náročné, a to nejen strojového, ale i „lidského“. Při stahování webů dochází ke stahování i všech subdomén, přičemž je třeba dávat pozor na to, že některé weby obsahují i informační systémy, vyhledávače a jiné služby, které program httrack nerozezná a dochází k stahování velkého množství "neužitečných" dat (v řádu jednotek GB). V tomto případě je nutný "ruční" zásah. Není však možné podat všeobecně platné doporučení, kdy má smysl ve stahování pokračovat, a kdy nikoli. Důsledkem je poté nemožnost automatizace.

Výsledky stahování a jeho dalšího zpracování jsou uvedeny v následujících odstavcích a tabulkách. Vzhledem k pracnosti získání těchto dat a jejich minimální využitelnosti se nedoporučuje získávat tato data automaticky jako zdroj.

Výsledkem stahování veřejných zdrojů je celkové množství 838 tis. pročištěných stránek. Jedná se však pouze o stránky, na kterých byl nalezen text. Stránky, na kterých

nezbyl po pročištění žádný text, byly mazány průběžně. Celkem bylo staženo přes 100 GB dat.

2.1.1. Klasifikace stažených webových stránek

S ohledem na množství stažených dat nelze ručně identifikovat ty stránky, které obsahují užitečné informace. Snaha řešitelského týmu byla vyřešit tuto úlohu automaticky pomocí standardních metod strojového učení.

Stažené informace z webových stránek nebyly pouze v českém jazyce, ale také v angličtině, popř. jiných jazycích. Byl proto zvolen obdobný postup jako v případě shlukové analýzy, tj. že u všech datbude nejprve identifikován jazyk, přičemž dokumenty v českém jazyce se přeložily do angličtiny.

Dokumenty byly roztříděny do celkem 7 kategorií:

1. zprávy, aktuality,
2. katedry, ústavy,
3. věda, výzkum,
4. všeobecné,
5. studium,
6. o škole,
7. lidé,
8. x = nezajímavé.

Dále byla využita metoda řízeného strojového učení, konkrétně logistické regrese implementované v nástroji Vowpal Wabbit[8]. Cílem implementace rysů (příznaků, z angl. features) bylo dosáhnout co nejpřesnější klasifikace.

Byly implementovány následující sestavy rysů:

- bag of words na kmenech slov,
- bigramové rysy na kmenech slov,
- bag of words vážený počtem výskytů,
- rysy z URL[9] stránky,
- rysy z titulku (nadpisu) stránky,
- indikátor prázdného dokumentu.

[8] https://github.com/JohnLangford/vowpal_wabbit/wiki

[9] jednotná adresa zdroje (z angl. Uniform Resource Locator)

Následující konfuzní matice (z angl. confusion matrix) ilustruje úspěšnost řazení dokumentů do jednotlivých kategorií. Byla vytvořena agregací výsledků získaných pomocí pětinásobné křížové validace na trénovacích datech klasifikátoru.

Tabulka 1: Konfuzní matice - úspěšnost řazení dokumentů do jednotlivých kategorií

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	59	5	1	0	0	0	2
3	2	3	67	7	0	2	1	5
4	7	0	12	146	25	15	1	16
5	0	0	3	19	114	8	1	8
6	0	0	1	2	1	11	0	0
7	0	4	0	3	0	0	113	0
8	1	3	7	27	18	10	0	166

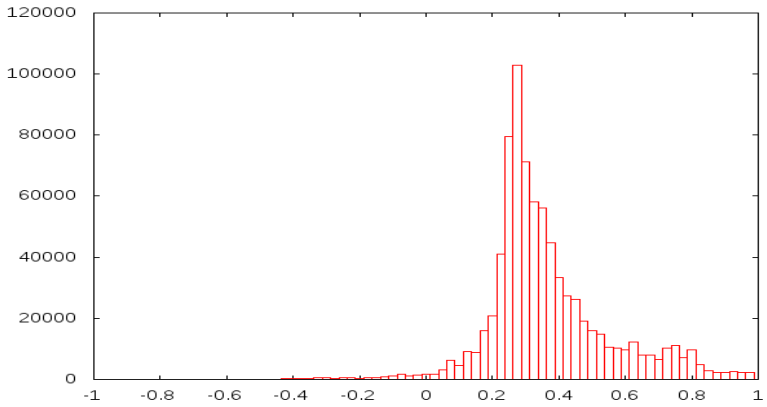
Jednotlivé řádky matice odpovídají třídám, které pro dokumenty určil automatický klasifikátor, sloupce odpovídají skutečným třídám (dle anotace). V ideálním případě (100% úspěšnost) by se nenulové hodnoty nacházely pouze na diagonále (automatická klasifikace odpovídá anotaci). Ve výsledcích se na diagonále nachází většina dokumentů: dosažená přesnost je cca 75 %, což se jeví jako dostatečné.

Potenciální problém představovalo rozložení kategorií v reálných datech, které je velmi odlišné od trénovacích dat. V trénovacích datech byla většina stránek ručně klasifikovaná jako relevantní (v tabulce označena čísly 1 až 7), zatímco v automaticky získaných datech byla naprostá většina dat nerelevantní (kategorie 8). Vyvinutý klasifikátor dokázal sice úspěšně předpovědět data podobná trénovacím, ovšem v reálných datech považoval chybně většinu stránek za relevantní. Problém byl částečně řešen přechodem ke klasifikaci do dvou tříd, tj. každý dokument je buď:

- nerelevantní (dřívější kategorie “x” a “všeobecné”),
- nebo relevantní (ostatní kategorie).

Po natrénování klasifikátor stále nepředkládal dobré výsledky, protože apriorní četnost kategorií v testovacích datech byla výrazně jiná, než v trénovacích.

Graf 2: Histogram shrnující četnosti automatických skóre reálných dokumentů



Skóre 0 bylo zvoleno pro **relevantní** dokumenty, 1 označuje **nerelevantní** dokumenty. Z histogramu je patrné, že rozdělení dokumentů podle klasifikátoru neodpovídá skutečnosti: většina dokumentů je považována za relevantní. Přechod k binárnímu klasifikátoru umožnil empiricky zvolit hranici, od které lze dokumenty považovat za relevantní, a tak (byť za cenu ztráty určité části relevantních dat) odfiltrovat většinu dokumentů.

Následující konfuzní matice ukazuje výsledky binárního klasifikátoru na trénovacích datech (získané opět pomocí křížové validace).

Tabulka 2: Konfuzní matice - výsledky binárního klasifikátoru na trénovacích datech

	Relevantní	Nerelevantní
Relevantní	426	241
Nerelevantní	33	200

Jak vyplývá z konfuzní matice, binární klasifikátor poměrně úspěšně identifikuje jak relevantní, tak nerelevantní dokumenty. Nejčastější chybou je označení nerelevantního dokumentu za relevantní (241 případů).

Při následné evaluaci na náhodném vzorku ze současného (finálního) crawlu jsme získali následující výsledky.

Tabulka 3: Konfuzní matice - evaluace na náhodném vzorku ze současného crawlu

	Relevantní	Nerelevantní
Relevantní	24	120
Nerelevantní	98	125

Je zřejmé, že klasifikátor nedokáže správně rozlišit testovací dokumenty - řadu relevantních prohlásí za nerelevantní a naopak. Lze sice pracovat s prahem pro přijetí/zamítnutí dokumentu, ale to znamená pouze dosažení situace, kdy klasifikátor prakticky všechny dokumenty zahodí. Nelze jej tedy využít k identifikaci zajímavých dokumentů.

Důvodem je zřejmě různorodost dokumentů napříč institucemi. Trénovací data byla získána dříve, kdy crawl obsahoval pouze několik málo institucí. Model naučený na těchto datech ale zjevně nezobecňuje.

Pokud jsou do trénovacích dat zahrnuty i testovací dokumenty ze současné verze crawlu, klesne přesnost v křížové validaci na cca 65 %. K evaluaci tohoto klasifikátoru byly využity anotace dalších 200 náhodných dokumentů ze současné verze crawlu. Přesnost na těchto datech je opět velmi nízká (42 %), podrobněji výsledky ukazuje tabulka.

Tabulka 4: Konfuzní matice - evaluace klasifikátoru s využitím dalších náhodných dokumentů

	Relevantní	Nerelevantní
Relevantní	32	88
Nerelevantní	29	51

Obecné řešení této úlohy se tedy jeví jako velmi obtížné. Je možné, že není ani dobře definovaná: např. nemáme k dispozici měření anotátorské shody, čili nelze rozhodnout, zda se na této klasifikaci shodnou lidé mezi sebou (pokud ne, byla by tato úloha výpočetně jen obtížně řešitelná).

2.2. Formát vstupních dat

Veškerá data z obou částí projektu byla převedena do stejného formátu XML, aby se s nimi dalo pracovat. Každý dokument se skládal ze dvou hlavních částí:

1. metadata,
2. textová data.

Metadata obsahovala „vnější“ údaje o dokumentu, tedy jména autorů, řešitelů apod., instituce (jedna nebo více), a hlavně identifikátor. Z těchto údajů byly pro tyto účely používány pouze identifikátor, který však byl upraven tak, aby byl skutečně jednoznačným identifikátorem daného dokumentu, to znamená, aby jednoznačně určoval daný dokument, a navíc aby splňoval požadavky na další automatické použití. Bohužel často nebylo možné převzít žádný z identifikátorů od poskytovatelů, neboť tyto požadavky nebyly splněny. Identifikační čísla obsahovala znaky, které se běžně používají např. jako oddělovače (mezery), nebo mají nějaký speciální význam v nástrojích, které byly použity. Rovněž byly identifikovány nejednoznačné identifikátory.

Pro lepší orientaci v dokumentech byl každému dokumentu přiřazen prefix označující zkratku zdroje (v případě RIV navíc rok výsledku, neboť se ukázalo, že existují různé dokumenty se stejným identifikátorem a odlišuje je pouze rok).

Textové položky byly označeny tagem <text> s případným upřesněním typu, např. Key_words nebo Abstract.

3. Zjišťování jazyka, překlad

Porovnávané dokumenty musí být ve stejném jazyce. V opačném případě by shluková analýza nedávala smysl - výsledkem by byly shluky závislé na jazyce, což není žádoucí. Vědecko-výzkumné texty, se kterými se pracovalo, byly psané buď v českém jazyce, nebo anglickém jazyce, případně obsahovaly části v obou jazycích. Lze se setkat i s jinými jazyky, kterých je ale minimální množství.

Shluková analýzy byla následně provedena s texty v anglickém jazyce. Důvody byly:

1. anglických textů je většina,
2. automatický překlad z českého jazyka do anglického jazyka má lepší výsledky než směrem naopak,
3. vyhledávání podobností mezi texty na základě slovních tvarů je v anglickém jazyce díky omezené morfologii obtížnější.

Z uvedených důvodů bylo nutné všechny texty z českého jazyka automaticky přeložit do anglického jazyka. Práce s texty v jiných jazycích napokračovala z důvodu jejich zanedbatelného množství. V případě, že by se v budoucnu ukázala potřeba pracovat i s jinými jazyky, bylo by možné jazykový modul pro daný jazyk připojit.

3.1. Příprava překladového systému

Pro strojový překlad textových dat byl vytvořen jednoduchý nástroj, jenž získává překlady pomocí webové služby. Jedná se o nástroj wrapper využívající softwarové řešení MTMonkey[10] vyvinuté na Ústavu formální a aplikované lingvistiky Univerzity Karlovy. Uvedený nástroj je již nasazen v několika projektech, ve kterých poskytuje strojové překlady prakticky v reálném čase.

Jako samotný překladový systém byl využit česko-anglický soutěžní systém vyvinutý na rovněž na Ústavu formální a aplikované lingvistiky Univerzity Karlovy pro minulý rok s názvem Workshop on Statistical Machine Translation[11]. Pro rychlost použití byl upraven jak překladový, tak jazykový model. Taktéž byly díky využití úsporných binárních formátů pro uložení modelů výrazně sníženy nároky na diskový prostor i paměť RAM. Jazykový model byl prostřednictvím nástroje KenLM[12] převeden do formátu kvantizované trie a pro frázovou tabulku byla využita implementace Compact Phrase Table[13], která mj. využívá minimální perfektní hashování k uložení frází.

Pro překlad vzorových dat byla využita instance systému spuštěná na serverech Ústavu formální a aplikované lingvistiky Univerzity Karlovy.

---

[10] z angl. Machine translation web service infrastructure (http://ufal.cz/mtmonkey)

[11] http://www.statmt.org/wmt14/translation-task.html

[12] https://kheafield.com/code/kenlm/

[13] http://ufal.mff.cuni.cz/pbml/98/art-junczys-dowmunt.pdf

3.1.1. Instalace překladu v TA ČR

Následně se uskutečnilo zprovoznění uvedeného řešení pro strojový překlad v TA ČR (stroj s 24 GB RAM, CPU se 4 jádry s HT a operačním systémem Debian Linux, 64-bit).

Na tento stroj (s názvem *cluster*) byly nainstalovány všechny potřebné nástroje, jmenovitě:

- Moses[14] - dekodér, tj. software provádějící samotné hledání vhodného překladu.
- MTMonkey - nástroj pro poskytování strojového překladu jako webové služby.
- Morphodita[15] - nástroj pro morfologickou analýzu, lemmatizaci a tagging vyvíjený na Ústavu formální a aplikované lingvistiky Univerzity Karlovy.

Rovněž byly na *cluster* zkopírovány modely pro česko-anglický překlad.

Nástroj Moses byl spuštěn v režimu server. MTMonkey byl konfigurován a spuštěn, aby na vstupu mohl být libovolný český text bez nutnosti předchozích úprav (z angl. pre-processing). V rámci webové služby následně proběhlo automatické rozdělení na věty (tzv. segmentace), jednotlivá slova (tzv. tokenizace) a lemmatizace (poslední dva kroky byly zajištěny pomocí mástroje Morphodita). Tato služba na stroji *cluster* běží nepřetržitě a poskytuje překlady na lokálním URL:

- 192.168.200.63:10000/mtmonkey

Formát dotazů na webovou službu (JSON API) je uveden v následujícím odkazu:

https://github.com/ufal/mtmonkey/blob/master/API.md.

3.2. Identifikace jazyka

Před samotným překladem bylo nezbytné provést identifikaci jazyka. Přestože je většina dokumentů z množiny podstatných dat opatřena identifikátorem jazyka, není tento údaj spolehlivý. Velmi často se např. v kolonkách pro anglický abstrakt objevují abstrakty české. Z tohoto důvodu je nutné před vlastním překladem testovat, v jakém jazyce se text nachází.

Bylo zvažováno a testováno několik nástrojů pro identifikaci jazyka. Mezi nimi YALI[16], vyvinutý na Ústavu formální a aplikované lingvistiky Univerzity Karlovy, nebo několik modulů pro programovací jazyk Perl volně dostupných v repozitáři CPAN (z angl. Comprehensive Perl Archive Network)[17].

[14] http://www.statmt.org/moses

[15] http://ufal.cz/morphodita

[16] http://ufal.mff.cuni.cz/tools/yali

[17] http://www.cpan.org/

Aktuálně je využíván nástroj TextCat[18], který po úpravě kódování (texty jsou kódovány v UTF-8 (z angl. UCS Transformation Format)[19] bylo nutno konvertovat soubory a spolehlivě rozlišuje anglický, český a slovenský jazyk.

Nástroj pro identifikaci jazyka přidá všem elementům <text> XML atribut guessed_lang s hodnotami "cs", "en" nebo "other". Do překladu vstupují texty v českém jazyce a části v anglickém jazyce jsou ponechány v původní podobě. Rovněž části označené jako "other", které jsou následně ze shlukové analýzy vyloučeny.

3.3. Strojový překlad dostupných dat

Texty v čekém jazyce byly přeloženy připraveným překladovým systémem (pro veřejné dokumenty, které převažovaly, byla pro urychlení využita výpočetní infrastruktura Ústavu formální a aplikované lingvistiky Univerzity Karlovy) a označeny atributem translated=true.

Výstupem této fáze byly nové verze XML dokumentů, v nichž byl veškerý text jednotně v anglickém jazyce, nebo je označeno, že se nemá dál zpracovávat (texty s atributem guessed_lang="other"). Soubory s texty v anglickém jazyce jsou vstupními daty pro shlukovou analýzu.

[18] http://odur.let.rug.nl/~vannoord/TextCat/

[19] UTF-8 používá proměnnou délku znaku, a to od 1 bajtu až do 6 bajtů

4. Shluková analýza

4.1. Porovnání vyzkoušených postupů

V tomto oddíle je nejprve popsána historie testování a výsledky těchto testů.

a) Vlastní nástroj vyvinutý na řešení jiné úlohy

Testování se uskutečnilo na abstraktech publikací z Mendelovy univerzity v Brně, která byla použita jako modelový příklad. Data sice nebylo možno zcela vyčistit, protože byla ve formátu .pdf (viz výše). Už první experimenty však ukázaly, že tento nástroj bude nepoužitelný kvůli výrazně většímu počtu vstupních dokumentů, které je třeba zpracovat.

b) Lingo3G[20]

Program na clustering dokumentů podle klíčových slov. Algoritmy na clustering zde mají vysoké množství nastavitelných parametrů. Výstupem z algoritmu je hierarchická struktura clusterů, ve které je každému clusteru přiřazeno charakteristické slovo (případně skupina slov). Dokáže zpracovat tisíce kratších dokumentů v řádu sekund.

Nevýhody:

- Nezvládá větší množství dokumentů.
- Jedná se o proprietární software. Bylo by tedy nutné k němu zakoupit licenci.

c) Scikit-learn a implementace shlukování v Python

Scikit-learn je knihovna pro jazyk Python, kterou lze použít i pro clustering dokumentů. Tento postup by zřejmě byl vyoužitelný. Algoritmus na clustering by se však musel naimplementovat ručně.

d) Apache Lucene

Nástroj vytváří databázi textových dokumentů a nad ní implementuje vysoce efektivní vyhledávání. Apache Lucene je ke stažení zdarma a s licencí přívětivou k použití pro komerční účely.

4.2. Shlukování s Lucene

Lucene mimo jiné poskytuje metody analýzy slov a tvorby tzv. term vektorů. Term vektor pro daný dokument obsahuje počet výskytů každého slova z dat v tom daném dokumentu. Například, máme-li slovník (*a, b, c, d, e*) a dokument „*a a b b e*“, term vektor pro tento dokument bude (*2, 2, 0, 0, 1*). Přirozeně, s počtem dokumentů výrazně narůstá velikost použitého slovníku, a tím i délka term vektorů reprezentujících jednotlivé dokumenty. Lucene umožňuje tento slovník do jisté míry zmenšovat, například

[20] http://carrotsearch.com/

filtrováním tzv. s*topwords,* čili slov nenesoucích význam, který by charakterizoval text (spojky, předložky, členy, pomocná slovesa atd.).

Další metody na zkrácení term vektorů jsou lemmatizace, stemming, nebo tzv. distribuované reprezentace.

Než bylo přistoupeno ke zpracování shlukové analýzy na celém souboru dokumentů, byla realizována řada experimentů s menším množstvím dat z důvodu nutnosti odhadu času potřebného ke zpracování. K tomu byly využity texty z databáze CORDIS.

a) Časová analýza shlukování

V následující tabulce jsou uvedeny informace o dobách běhu shlukování s daným nastavením na různém počtu dokumentů. Tabulka rovněž obsahuje údaje o tom, kolik termů (slov) bylo vyřazeno ze slovníku na základě heuristik popsaných ve sloupci Setting. Sloupec Vector length pak ukazuje délku term vektoru (= velikost slovníku).

Tabulka 5: Časová analýza shlukování

	Documents	Setting	Skipped terms	Vector length (non-skipped terms)	Duration [s]
Baseline	100		0	4888	5,54
	200		0	7096	28,88
	400		0	10534	166,41
Re-use of norm calculation results	100		0	4 888	2,89
	200		0	7 096	12,46
	400		0	10 534	67,25
	800		0	14 414	360,10
Skip low-frequency terms	100	total tf >= 10	4 472	416	0,68
	200	total tf >= 10	6 220	876	1,61
	400	total tf >= 10	8 925	1 609	7,12
	800	total tf >= 10	11 951	2 463	37,47
DF threshold	800	df >= 2	7 703	6 711	92,78
	800	df >= 2 & total tf >= 10	11 967	2 447	36,98
Analyzers	800	df >= 2 & total tf >= 10 & En Analyzer	7 882	1 925	29,18
	1600	df >= 2 & total tf >= 10 & En Analyzer	11 164	2 722	152,84
SLINK	100	df >= 2 & total tf >= 10 & En Analyzer	2 921	481	0,74

	200	df >= 2 & total tf >= 10 & En Analyzer	3 946	844	1,71
	400	df >= 2 & total tf >= 10 & En Analyzer	5 672	1 383	6,70
	800	df >= 2 & total tf >= 10 & En Analyzer	7 882	1 925	30,26
	1 600	df >= 2 & total tf >= 10 & En Analyzer	11 164	2 722	161,26
	3 200	df >= 2 & total tf >= 10 & En Analyzer	16 076	4 009	945,33

Poznámka:

Baseline - výchozí implementace hierarchického shlukování. Její časová složitost je $O(n^3)$, kde n označuje počet dokumentů. Jak je z uveedného patrné, už na několika stovkách dokumentů je tato metoda velice pomalá a pro počet dokumentů v řádech tisíců a výše zcela nepoužitelná.

Re-use of norm calculation results - výsledkem aplikace dynamického programování na původní řešení bylo zjištěno relativně výrazné zlepšení. Stále je však, co se týče délky výpočtu, toto řešení nepoužitelné pro řádově tisíce dokumentů.

Skip low-frequency terms - metoda odstraní ze slovníku termy s nízkou frekvencí. Je k diskusi, zda nemohou být právě málo frekventovaná slova, podle nichž se dokumenty shluknou. Je-li ale v tak velkých datech jedno slovo spatřeno pouze desetkrát (výchozí hodnota pro další pokusy), lze předpokládat, že se jedná buď o překlep, nebo o velice vzácný výskyt, který nemá velkou vypovídací hodnotu. Tato metoda přinesla další řádové zrychlení. Stále však platí, že pro např. 100 tis. dokumentů by probíhal výpočet cca 1,5 roku.

DF threshold - metoda filtruje slova, která se vyskytla v méně než 2 dokumentech. Podle nich shlukovat nelze, neboť se vyskytla jen v jednom dokumentu a nelze zjistit podobnost s jinými dokumenty. Samotná metoda nepřinesla žádné zlepšení. V kombinaci s předchozí metodou došlo ke zlepšení avšak zanedbatelnému.

Analyzers - metoda přidává filtrování anglických *stopwords.* Samotná metoda opět nepřinesla žádné zlepšení. V kombinaci s předchozí metodou došlo ke zlepšení avšak zanedbatelnému.

4.3. Algoritmus SLINK

Poslední vyzkoušenou metodou je algoritmus SLINK[21]. Uvedený algoritmus na hierarchické clusterování funguje v čase $O(n^2)$. Z tohoto důvodu by měl být řádově lépe škálovatelný na velké objemy dat než uvedené předchozí metody. Další výhodou algoritmu je, že ho lze použít online, to znamená, že do už vytvořené hierarchické

[21] An optimally efficient algorithm for the single-link cluster method

struktury lze přidávat nové dokumenty (tzv. roubování, viz výše). To platí pouze za předpokladu, že se nezmění slovník, který musí od začátku zůstat neměnný. Slova z nových dokumentů, která nejsou ve slovníku, se tedy musí ignorovat. Jesliže bude akceptováno toto omezení, přidávání dokumentů bude lineární s počtem dokumentů, čili O(n). Budou-li extrapolovány časy z tabulky kvadratickým polynomem, lze získat pro 100 tis. dokumentů ve verzi uvedeného algoritmu (nakombinovaná s předchozími metodami na zmenšení slovníku) čas zpracování cca 2 týdny.

Další zrychlování algoritmu

Další možností zrychlení algoritmu je použití metody, která zkracuje délku vektoru reprezentujícího jednotlivé dokumenty. Zmíněné zlepšení nepřinese asymptoticky lepší časovou složitost algoritmu, nicméně dokáže čas běhu několikanásobně zmenšit. Hlavním nedostatkem předchozích metod je však délka vektoru reprezentujícího dokument, který roste s počtem dokumentů, resp. s množstvím různých slov, které se v nich vyskytují. Pokud by se podařilo, aby byl vektor pořád stejně dlouhý, lze dosáhnout výrazně lepších časů pro velký objem dat. V následujících odstavcích jsou popsány dvě metody, které zmíněné umožňují.

4.4. Latent Semantic Analysis

Latent Semantic Analysis[22] (dále LSA) je metoda založená na technice rozkladu na singulární hodnoty, používané v lineární algebře. V praxi metoda funguje tak, že odstraní z vektoru slova, která nesou nejméně informace o podobnosti dvou dokumentů. Pokud má být metoda aplikována, je spuštěno vytvoření dlouhých vektorů (viz předchozí popis) a následně metoda LSA. Tím dojde k vytvoření sady krátkých vektorů, jejichž délku lze nastavit počátečním parametrem. Vektory mohou být následny využity pro shlukování.

4.5. Doc2vec

Další způsob, jak získat kratší vektory reprezentující dokumenty, je tzv. distribuovaná reprezentace popsaná v materiálu Distributed Representations of Sentences and Documents[23]. Dokumenty jsou namapovány do n-dimenzionálního prostoru reálných čísel tak, aby podobnost dokumentů byla reprezentována vzdáleností vektorů v tomto prostoru. Pro informaci lze uvést, že počet dimenzí takového prostoru se pohybuje v řádu desítek až stovek, tj. v mnohem menších číslech než jsou čísla ve sloupci Vector length z tabulky časové analýzy.

[22] https://en.wikipedia.org/wiki/Latent_semantic_analysis

[23] https://cs.stanford.edu/~quocle/paragraph_vector.pdf

4.6. Implementace shlukování algoritmem SLINK s použitím LSA a Lucene

Pro implementaci metody LSA byl vybrán projekt SemanticVectors, jež umí mimo jiné pracovat i s dokumenty uloženými v Lucene. Jednotlivé kroky pro vytvoření shluků jsou:

- Vytvoření Lucene databáze z připravených XML dokumentů.
- Konstrukce sémantických vektorů reprezentujících dokumenty.
- Spuštění shlukovacího algoritmu SLINK na vektorových reprezentacích.

Tento konkrétní přístup byl postupně aplikován nejprve na data z Informačního systému výzkumu, experimentálního vývoje a inovací - Centrální eveidence projektů (celkem 42 224 dokumentů).

Tabulka 6: Časové běhy metody LSA pro různé velikosti vstupních dat

	Počet dokumentů	Čas běhu [s]	Počet dokumentů	Čas běhu [s]
LSA (dim200)	100	0,09	3 200	3,57
	200	0,10	6 400	13,98
	400	0,15	12 800	53,31
	800	0,34	25 600	255,28
	1 600	1,20	42 224	707,56

Z tabulky vyplývá, že algoritmus je schopný pracovat i s velkými počty dokumentů v relativně krátké době. Pokud jsou uvedená čísla proložena kvadratickým polynomem (algoritmus shlukování má kvadratickou časovou složitost), lze odhadnout čas běhu na větších datech. Pro 100 tis. dokumentů byl odhad běhu 68 min., pro 1 mil. dokumentů, což je očekávaný počet dokumentů, na kterém se shluková analýza bude provádět, se odhad pohybuje kolem 4,8 dne.

Při skutečných bězích byly naměřeny časy o trochu kratší, nicméně je třeba počítat s takto dlouhými dobami zpracování.

Výsledkem hierarchické shlukové analýzy je tzv. dendrogram (strom), který vznikl odspoda postupným sléváním shluků od jednotlivých dokumentů až po jediný shluk. Výška ve stromě, ve které došlo ke spojení dvou shluků, odpovídá tomu, jak moc se od sebe dva shluky liší – čím dříve se tedy dokumenty potkají ve stejném shluku, tím by si měly být podobnější.

4.7. Roubování nových dokumentů na hotový dendrogram

Roubováním se rozumí zařazení nového dokumentu do již vytvořeného dendrogramu, který vznikl shlukovou analýzou.

K naroubování nového dokumentu je nutné mít:

1. zpracovaný dendrogram,

2. nový dokument, který chceme naroubovat,

3. všechny dokumenty z původního dendrogramu (staré dokumenty).

Nový dokument se přidá do databáze ke všem starým dokumentům a pomocí algoritmu na reprezentaci dokumentů se ke každému znovu vytvoří vektor termů. Tyto vektory se ovšem nyní nepoužijí k nové shlukové analýze, jen se jimi nahradí původní vektory reprezentující zmíněné staré dokumenty, které jsou napevno umístěné ve struktuře dendrogramu.

Nový dokument se, resp. jeho reprezentace vektorem termů, porovná se všemi novými vektory starých dokumentů, a zařadí se = naroubuje = na dendrogram. Výsledkem je nový dendrogram, který se od toho starého liší pouze přidáním jednoho uzlu.

4.8. Prohlížení výsledků shlukové analýzy

Aby byl vyřešen problém s velikostí stromu, byl vytvořen nástroj, kterým lze strom procházet a prohlížet ho lokálně. Pomocí tohoto nástroje lze načíst výsledný strom, zvolit si konkrétní výšku ve stromě, zobrazit statistiku o velikostech shluků v této výšce, zobrazovat konkrétní shluky v konkrétní výšce, apod. Součástí je i nápověda k jeho použití. Nástroj je interaktivní a volá se z příkazové řádky.

4.9. Metody evaluace výsledků

U výsledků shlukové analýzy je nutné ověřit, zda jsou smysluplné, tj. zda se skutečně shlukují napřed dokumenty, které jsou si velmi podobné, případně zcela shodné, a následně dokumenty méně a méně související. K tomu je třeba ruční evaluace. Je nezbytné, aby se na výsledky podíval anotátor a zhodnotil, zda výsledné shluky skutečně obsahují dokumenty se stejnou nebo podobnou tematikou. Nejvhodnější se jeví, aby stejnou anotaci provedlo anotátorů více z důvodu vyloučení vlivu subjektivního hodnocení. V uvedeném případě byli využiti celkem dvou anotátoři.

V této souvislosti existuje několik metod na provádění a následné vyhodnocení ruční evaluace.

a) Ruční kontrola struktury stromu, jež je základním evaluačním postupem, který je schopen odhalit fundamentální chyby v návrhu shlukovacího algoritmu. Pozorování založená na aplikaci této metody jsou popsána v následující sekci.

b) Pro ověření předpokladu, že podobné dokumenty budou v dendrogramu ve stejném shluku v nižší výšce než méně podobné dokumenty, byl navržen následující postup: Anotátor obdrží náhodný shluk a přiřadí mu skóre od 1 do 5, podle podobnosti dokumentů uvnitř shluku. (5 = velice podobné až identické texty napříč celým shlukem, 1 = shluk obsahuje dokumenty s naprosto rozdílnými texty.) Podrobné pokyny k anotaci jsou obsaženy v příloze 1 tohoto materiálu. Po ruční anotaci menší porce výsledných dat lze změřit korelaci mezi hodnotou skóre přidělenou anotátory a výškou ve stromě, na které shluk vznikl. Pokud bude tato korelace vysoká, existuje předpokla, že podobné dokumenty budou v dendrogramu ve stejném shluku.

c) Metoda evaluace shlukování zmíněná v úvodu předkládá anotátorovi shluk s jedním náhodným přidaným dokumentem. Anotátor má za úkol odhalovat tyto přidané dokumenty. Čím vyšší je jeho přesnost, tím věrohodnější je výsledek shlukování.

Nedostatky současného řešení:

Použitím algoritmu SLINK byl vytvořen strom, jehož strukturu tvoří jeden nebo dva větší shluky, které na sebe postupně nabalují ostatní, menší shluky. Tyto malé shluky zřídkakdy dosáhnou velikosti více než deseti dokumentů, než se připojí do velkého shluku.

Pro naše testování jsou zajímavé právě menší shluky, které obsahují jednotky dokumentů. V nich lze nalézt zajímavosti (např. duplicity, podobná témata, apod.), které naznačují, že shlukování funguje v pořádku. Nicméně, například pro evaluaci podle (3) z minulé sekce by bylo nutné znát jednu konkrétní výšku ve stromě, ze by byly vzaty shluky, aby podobnost mezi původními dokumenty shluku byla stále na stejné úrovni. To provést po první shlukové analýze, avšak došlo by ke ztrátě informací ze shluků, které vznikly buď níže nebo výše ve stromě.

Alternativa - algoritmus CLINK**:**

Algoritmus CLINK (z angl. Complete-Linkage Clustering) je obdobou algoritmu SLINK (viz kapitola 4.3). Na rozdíl od SLINK zmíněná metoda porovnává shluky na základě podobností všech dokumentů. To by mohlo zabránit problému, který vznikl s metodou SLINK kvůli tomu, že porovnává vždy jen nejpodobnější dva dokumenty z každého shluku.

Po realizaci experimentů s algoritmem CLINK bylo zjištěno, že problém s nevyvážeností stromu byl vyřešen.

4.10. Výsledky shlukové analýzy

První dendrogram, vytvořený na základě první shlukové analýzy pomocí algoritmu SLINK, byl testován dle následujícího stanoveného postupu:

Nejprve bylo vygenerováno celkem 103 náhodných shluků, obsahujících 3-10 dokumentů. Dvě anotátorky měly každý shluk ohodnotit navrženým skóre (viz příloha). Obě anotátorky přiřadily stejné skóre v 55 případech. V ostatních případech se výsledné skóre lišilo maximálně o 1 bod, až na dva případy, z nichž v jednom případě se jednalo o chybnou záměnu nejlepšího a nejhoršího skóre (1 vs. 5), ve druhém se skóre lišilo o 2 body. Pro jednodušší vyhodnocení evaluace obou anotátorek byla pětistupňová škála převedena do binární, a to následovně:

- 1 nebo 2 (málo podobné) byly převedeny na 0,
- 3 až 5 bylo převedeno na 1.

Dle uvedeného binárního hodnocení byla shoda obou anotátorek na 75 %.[24] Z těchto statistik lze dojít k závěru, že na výsledky obou anotátorek je možné se spolehnout.

Technické podrobnosti o zvoleném řešení shlukové analýzy jsou uvedeny v příloze 2.

[24] Pro exaktní vyhodnocení tzv. mezianotátorské shody se navíc počítá tzv. koeficient kappa, který bere v úvahu i případnou náhodnou shodu. Zjednodušeně řečeno odečítá od absolutní shody (v uvedeném případě 75 %) shodu, která by nastala pravděpodobně tehdy, kdyby obě anotátorky přiřazovaly 1 nebo 0 zcela náhodně. Pro dané hodnocení vyšel koeficient kappa 0,41. Pro tento případ však není hodnocení zcela relevantní, neboť nečiní rozdíl mezi blízkým a vzdáleným hodnocením obou anotátorek. Jinými slovy, nepočítá s tím, že hodnocení tvoří škálu, a považuje za stejný rozdíl hodnoceni 1 a 5, který nastal pouze jednou (viz výše) i rozdíl 1 a 2.

5. Seznam použitých zkratek

KA 3	Klíčová aktivita 3
LSA	Latent Semantic Analysis
MFF	Matematicko-fyzikální fakulta Univerzity Karlovy v Praze
Projekt Zefektivnění činností TA ČR	Zefektivnění činnosti TA ČR v oblasti podpory VaVaI a podpory posilování odborných kapacit organizací veřejné správy v oblasti VaVaI
RIV	Rejstřík informací o výsledcích
TA ČR	Technologická agentura České republiky
VaVaI	Výzkum, experimentální vývoj a inovace

6. Seznam použitých zdrojů

Internetové zdroje

- http://cordis.europa.eu/
- http://www.epo.org/searching-for-patents/business/patstat.html#tab1
- https://github.com/JohnLangford/vowpal_wabbit/wiki
- http://ufal.cz/mtmonkey
- http://www.statmt.org/wmt14/translation-task.html
- https://kheafield.com/code/kenlm/
- http://ufal.mff.cuni.cz/pbml/98/art-junczys-dowmunt.pdf
- https://github.com/ufal/mtmonkey/blob/master/API.md
- http://ufal.mff.cuni.cz/tools/yali
- http://www.statmt.org/moses
- http://ufal.cz/morphodita
- http://www.cpan.org/
- http://odur.let.rug.nl/~vannoord/TextCat/
- http://carrotsearch.com/
- https://en.wikipedia.org/wiki/Latent_semantic_analysis

Ostatní

- Sibson, R.: SLINK: an optimally efficient algorithm for the single-link cluster method. The Computer Journal. Ročník 16, č. 1. 1973. str. 30–34.
- Le, Q., Mikolov, T. Distributed Representations of Sentences and Documents [online]. In: arXiv preprint arXiv:1405.4053. 2014 [https://cs.stanford.edu/~quocle/paragraph_vector.pdf]
- Program veřejných zakázek ve výzkumu, experimentálním vývoji a inovacích pro potřeby státní správy BETA schválený usnesením vlády č. 54 ze dne 19. ledna 2011 [online]. In: Portál TA ČR [https://www.tacr.cz/index.php/cz/programy/program-beta.html]

7. Přílohy

Příloha 1: Anotace výsledků shlukové analýzy

Hodnoty 1 – 5, přičemž 1 znamená nejhorší hodnocení, 5 znamená nejlepší hodnocení.

1	Žádná shoda, minimální shoda ve výsledcích, výsledky nemají žádný společný jmenovatel (např. obsah, téma apod.), naprosto odlišné téma výsledků a vědní obor (např. jeden výsledek je o pozorování koček, druhý výsledek o zpracování kovů).
2	Větší než minimální shoda, shodný jmenovatel u všech zobrazených výsledků (např. totožný nástroj na měření, objekt výzkumu, apod.). Výsledky jsou si málo podobné, ale určitá shoda v nich lze nalézt. Pokud jsou některé výsledky naprosto stejné + zároveň jeden nebo více výsledků naprosto odlišných (naprosto jiné téma).
3	Shoda ve výsledcích, minimální požadavek je stejné téma výsledků, výsledky jsou si hodně podobné, výsledky mají více společných věcí (např. téma + objekt výzkumu + nástroj měření + ..., atd.) Pokud jsou některé výsledky naprosto stejné + zároveň je jeden nebo více odlišných, ale všechny výsledky mají společné téma apod. Pokud mají výsledky např. naprosto stejný název, ale liší se v anotaci nebo naopak, pokud mají výsledky naprosto stejnou anotaci, ale liší se v názvech.
4	Většina výsledků je totožná, jsou mezi nimi pouze malé nesrovnalosti (např. všechny výsledky mají naprosto stejnou anotaci, ale názvy výsledků se drobně odlišují nebo naopak všechny výsledky mají stejné názvy, ale drobně se liší v anotaci). Výsledky na pokračování - je patrné, že všechny výsledky se týkají jednoho výzkumu (např. mohou mít naprosto stejnou anotaci, ale název se liší), ale výzkum je rozdělen na více částí (často je u takových výsledků napsáno: 1. část, 2. část, I., II., ...).
5	Naprostá shoda ve výsledcích, mohou se lišit jen minimálně (např. maximálně jedním slovem v názvu nebo minimálně pozměněnou anotací).

Příloha 2: Technická dokumentace ke shlukové analýze

Text popisuje nástroje vyvinuté pro provádění shlukové analýzy na velkých objemech dokumentů.

Komponenty řešení

- Index builder – psaný v Java, převádí XML data do Apache Lucene indexu (databáze).
- Semantic vectors – knihovna třetí strany, na základě dat a analýzy Lucene indexu přiřazuje dokumentům sémantické vektory.
- Clustering – psáno v Javě, provádí samotnou shlukovou analýzu v sémantickém prostoru.
- Grafting – přidává nové dokumenty do již vytvořeného shlukovacího stromu (dendrogramu).
- Analyzér – command-line nástroj psaný v Perlu, umožňuje jednoduché a přehledné procházení velkých dendrogramů.

Jednotlivé nástroje jsou popsány v individuálních odstavcích níže.

Shluková analýza všech dokumentů

Pro vytvoření shluků ze všech dokumentů (od začátku) je třeba použít následující sekvenci nástrojů:

1. Index Builder – vstupem pro tuto aplikaci je adresář, který obsahuje (přeložené) XML soubory odpovídající jednotlivým dokumentům, a adresář, do kterého se zapíše výsledná databáze. Výstupem je Lucene index (databáze), který všechny dokumenty obsahuje.
2. Semantic Vectors – vstupem je adresář s Lucene indexem a dimenze sémantického prostoru, do kterého se dokumenty zobrazí. Výchozí volba je 200, což znamená, že každý dokument bude zobrazen na 200prvkový vektor. Samotné zobrazení zprostředkovává knihovna SemanticVectors. Zobrazení na tyto vektory do jisté míry zachovává sémantickou podobnost mezi dokumenty založenou na výskytech důležitých slov.
3. Clustering – vstupem jsou sémantické vektory pro každý dokument (sestavené v předchozím kroku). Výstupem je tzv. dendrogram, tj. reprezentace hierarchického shlukování konkrétní množiny dokumentů. Dendrogram je popsán výstupním souborem v následujícím formátu:

   id_dokumentu <mezera> label <mezera> lambda <mezera> pi

   kde ID dokumentu je pořadové číslo konkrétního dokumentu v rámci dendrogramu (dokument, který byl vybrán shlukovacím algoritmem jako první bude mít ID = 1, další bude mít ID = 2 atd.), label je text, který jednoznačně identifikuje konkrétní dokument (ve zmíněním případě buď název souboru nebo hodnota uvnitř XML

tagu internalID a lambda a pi jsou hodnoty, které (jednoznačně) určují pozici tohoto dokumentu uvnitř dendrogramu.

Přidávání nových dokumentů do existujícího dendrogramu (tzv. roubování)

Pro přidání nových dokumentů do daného stromu je zapotřebí mít připravený soubor obsahující daný dendrogram, adresář s Lucene indexem, ze kterého je pomocí sémantických vektorů daný dendrogram vytvořen, a seznam souborů, k přidání do dendrogramu.

1. Index Builder – vstupem pro tento nástroj je adresář s přidávanými dokumenty a cesta k existujícímu Lucene indexu, ze kterého byl vytvořen „starý“ dendrogram. Tento nástroj přidá nové soubory do stávajícího indexu.
2. Semantic Vectors – z obohacené databáze se znovu vytvoří vektory reprezentující jednotlivé dokumenty (včetně nových, přidaných dokumentů). Tyto vektory nemusí odpovídat starým sémantickým vektorům vytvořeným v průběhu předchozího shlukování. Nové vektory budou použity k odhadu podobnosti mezi dvojicemi nově přidávaných dokumentů a podobnosti mezi stávajícími a nově přidávanými dokumenty.
3. Grafting neboli roubování - vstupem je starý dendrogram (ze kterého se použijí implicitně zakódované podobnosti mezi dvojicemi starých dokumentů), seznam přidaných dokumentů a sémantické vektory všech dokumentů z databáze (tj. stávajících i přidávaných), vzniklé v předchozím kroku. Pomocí nových informací doplní starý dendrogram o nové dokumenty a vytvoří nový strom, který je výstupem tohoto nástroje. Nový dendrogram je ve stejném formátu jako starý, obsahuje však o tolik řádků navíc, kolik bylo do dendrogramu přidáváno dokumentů.

Jaroslava Hlaváčová a další
KA 3
Metodika postupu shlukové analýzy
Vydala: Technologická agentura ČR, Evropská 1692/37, 160 00 Praha 6
http://www.tacr.cz
Praha 2016
1. vydání

ISBN 978-80-88169-14-7