Masarykova univerzita
Fakulta informatiky

Seskupování zpravodajských článků o stejné události

Diplomová práce

Jiří Vejvoda

Brno, podzim 2013

Prohlášení

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Jiří Vejvoda

Vedoucí práce: doc. RNDr. Petr Sojka, PhD.

Poděkování

Zde bude uvedeno „poděkování“ ...

Obdobně jako poděkování se mohou vysadit shrnutí a klíčová slova pomocí prostředí ThesisAbstract a ThesisKeyWords.

Obsah

1 Úvod

Text ...

2 Seskupování zpráv popisujících stejnou událost

Denně vychází velké množství zpravodajských článků, jejich autoři od sebe navzájem často opisují, ten stejný článek je mnohdy rozesílán několika různými RSS kanály. K jedné světové události mohou vycházet články několik dní až týdnů podle závažnosti nebo atraktivity události. Orientace v takovém množství je čím dál náročnější.

Projekt si dává za cíl vytvoření systému pro automatické rozdělení množiny zpravodajských článků na disjunktní podmnožiny podle událostí, které články popisují. Toto rozdělení má širokou škálu uplatnění, nejdůležitější je přetváření chaotických informací na strukturovanou formu, která je mnohem vhodnější k dalšímu automatickému zpracování. To mohou velmi ocenit například zpravodajské služby, velikosti podmnožin jsou také dobrým ukazatelem atraktivity události v rámci dané oblasti (specifikované výběrem zdrojů pro zpracovávané články). Při studování nějaké konkrétní události je to snadný přístup ke všem dostupným informacím zveřejněným danými zdroji.

Na toto rozdělení do podmnožin je možné pohlížet také jako na relaci mezi dvojicemi článků na této množině. Dva články jsou v relaci, jedině když oba popisují tutéž událost. Z toho triviálně vyplývá, že tato relace je reflexivní a symetrická. Transitivitu můžeme jednoduše získat metodikou, kdy článek patří do clusteru ve chvíli, kdy jeho skóre přesáhne definovanou hranici alespoň s jedním dalším článkem v daném clusteru. Tím získáme toto rozdělení ve formě relace ekvivalence.

Systém si klade za cíl zpracovávat i menší soubory vstupních dat, pro které by bylo (kvůli velikosti) nevhodné statistické zpracování.

2.1 Dosavadní přístupy

2.1.1 Konkrétní řešení

3 Využitelné nástroje a techniky

Systém navržený v rámci této práce bude využívat různé nástroje a techniky ke zpracování přirozeného jazyka (*NLP – Natural Language Processing*) a pomocí jejich výstupů dosahovat vlastních výsledků. Problém svým charakterem spadá do sémantické analýzy, uvedené nástroje a techniky se zabývají právě touto rovinou jazyka. Ostatní potřebné nástroje zpracovávající data na jiných úrovních budou uvedeny a popsány v kapitole o předzpracování vstupních dat.

3.1 Klasické NER

Rozpoznávání pojmenovaných entit (*NER – Named Entity Recognition*) je tradiční disciplína NLP zabývající se detekcí a kategorizací pojmenovaných entit jako jsou osobní jména, názvy míst a organizací a různé další. Sice se nedá říci, že by bylo kompletně vyřešené, ale dává v dnešní době použitelné výsledky minimálně pro angličtinu a několik dalších široce používaných jazyků.

Tato problematika je ve většině případů řešena statistickým přístupem, kdy se naučí modely na trénovacích datech (korpusech s vyznačenými pojmenovanými entitami) a tyto modely jsou dále využívány k detekci pojmenovaných entit na nových vstupních datech. Jako trénovací data se pro angličtinu často používají korpusy vytvořené při příležitosti zadávání sdíleného úkolu na konferencích CoNLL, hlavně z roku 2003, případně ještě z roku 2008, kde byl sice zadaný sdílený úkol jiný, poskytnutý korpus měl ovšem vyznačené i pojmenované entity. Rozpoznávané entity nejčastěji bývají osoby, organizace a místa, případně se ještě přidává zvláštní označení pro blíže neurčené entity. Toto rozdělení je především určeno způsobem označkování trénovacího korpusu a rozdělení je tedy možné udělat i jemnější. Samozřejmě platí, že čím více rozpoznávaných entit, tím stoupá náročnost úkolu a většinou se to negativně projeví na úspěšnosti.

3.1.1 Dostupné nástroje

Samotných NER nástrojů je větší množství, co se zpracování angličitny týče, určitě za zmínku stojí Stanford Named Entity Recogni-

zer Finkel, Grenager a Manning 2005, dále pak Illinois Named Entity Tagger Ratinov a Roth 2009, případně ještě modul dostupný z NLTK knihovny Bird, Klein a Loper 2009 pro NLP v Pythonu. Důležitým aspektem je i licence, pod kterou jsou nástroje nabízené, stanfordský parser je vydán pod GPL, NLTK pod Apache licencí verze 2 a Illinois má vlastní licenci umožňující volné použití pro studijní a akademické účely a omezující komerční placené využívání.

Všechny jmenované nástroje mají tu výhodu, že je možné je nainstalovat a využívat ve specifickém programu relativně jednoduše, zajištění podpory pro další jazyk je většinou možné vytvořením dostatečně velkého označkovaného korpusu a naučením modelu pomocí tohoto korpusu. Nehrozí, že by nástroj během pár let příliš zestárl a celková efektivita programu, kde je využíván, by rapidně klesla. Tyto nástroje jsou ovšem určené především na vyznačení daných entit v textu. Samotný text a druh entity neříká nic o tom, co daná entita zastupuje, je potřeba dodatečné zpracování už jen k určení, které entity v jednom textu označují stejnou věc. Toto zpracování je netriviální už kvůli používání ruzných zkratek pro názvy organizací, míst (New York, NY, N. Y.), různé skloňování jmen, případně neuvádění křestních a prostředních jmen, opět jejich zkracování, používání přezdívek a pod.

3.2 NER se zjednoznačněním entit

Jedním ze způsobů, jakými se dají zmíněné problémy řešit, je využívání znalostních bází (anglicky *Knowledge Base*), kde jsou sesbírané různé faktické informace o reálném světě. Vytváření a využívání znalostních bází je směr, jakým se aktuálně orientuje hodně aplikací zabývajících se NLP[1], jedná se o další krok k analýze sémantické roviny jazyka.

Napojení na znalostní báze má hlavní výhodu v tom, že je možné pojmenované entity zjednoznačnit (s určitou pravděpodobností), vyhledáním pojmenované entity v databázích, kde jsou jednotlivé odkazy na tyto záznamy (URI) v rámci databáze jedinečné. Samozřejmě se může stát, že některé označené entity se nepodaří propojit se zá-

1. Tento trend je možné pozorovat i u napojení Googlem vytvořeného Knowledge Graph na jeho vyhledávač v roce 2012

znamy v databázích ať už proto, že pravděpodobnost vyšla příliš malá, záznam v databázi chybí, nebo byla entita propojena s chybným záznamem.

3.2.1 Znalostní báze

Rozsah znalostních bází neustále roste a samotné databáze se liší svým zaměřením, ať už tématickým nebo lokálním/globálním, a samozřejmě také rozsahem. Dalším specifikem je, jak rychle údaje v databázi stárnou, pro naše účely budou nejdůležitější databáze s geografickými názvy, známými osobnostmi, společnostmi a údálostmi. Geografická data netrpí tolik zmíněným stárnutím, kdežto mezi známé osobnosti a společnosti přibývají často zmiňované entity velmi rychle a často se tomu tak děje právě na základě událostí, které popisují aktuální zpravodajské články, které má tento projekt za cíl zpracovávat. Je tu tedy kladen velký důraz na aktuálnost dat ve znalostní bázi, přesněji je nutné aby existovaly záznamy o entitách, které vznikly nebo se staly známými až v poslední době. Znalostní báze je tedy nejlepší využívat přímo online s nejaktuálnějšími záznamy.

Základní stavební jednotkou dat uvnitř databází jsou RDF trojice (*Resource Description Framework*), ty reprezentují vždy vztah dvou entit a celou databázi je pak možné si nejjednodušeji představit jako orientovaný graf. Definice těchto trojic má spíše abstraktní formu, každá databáze používá svou vlastní implementaci, je však přesně nadefinována jejich serializace a díky tomu je možné propojovat RDF trojice mezi různými databázemi.

Mezi hlavní znalostní báze patří DBpedia Bizer, Lehmann et al. 2009 vyvíjená Lipskou univerzitou a Svobodnou univerzitou Berlín, první verze byla vydána na začátku roku 2007. Databáze je vytvářena ze stránek Wikipedia.org, používá se automatická extrakce strukturovaných dat především z tabulek se základními informacemi. Stejně jako klasická Wikipedia je i DBpedia vícejazyčná, anglická verze obsahuje 3,77 milionu záznamů, z toho 2,35 milionu je zařazeno do ontologie, celkově DBpedia sestává z 10,3 milionů záznamů ve 111 různých jazycích. DBpedia obsahuje 1,89 miliardy RDF trojic a je tak největší volně dostupnou znalostní bází Sahnwaldt 2013.

Další rozsáhlá volně dostupná znalostní báze je Freebase zalo-

žená roku 2007 společností Metaweb[2], ta je narozdíl od DBpedie nejen automaticky vyextrahovaná z Wikipedie, využívá automatickou extrakci i z dalších zdrojů a hlavně je část jejích dat přímo ručně vytvořena uživateli. Freebase obsahuje téměř 40 milionů záznamů[3] a přes 337 milionů RDF trojic.

3.2.2 Linked Data

Velké množství ruzných znalostních bází přineslo potřebu propojit je do jednoho celku, to je možné pokud jsou databáze stavěny na nějaké formě RDF trojic, nebo jsou do tohoto formátu alespoň převoditelné. Tak vznikl komunitní projekt Linking Open Data W3C SWEO Community Project 2013 a k němu dále Linked Data Bizer, Heath a Berners-Lee 2009, ty si dávají za cíl propojit entity z různých databází (tzv. *SameAs links* propojující různé URI v jednotlivých databázích zastupujících stejnou věc). Takto je popropojované opravdu velké množství různých databází, jak je vidět na grafu v Příloze, hlavním středem dat je DBpedia, současně ale vznikají souvislejší komponenty na pár místech v grafu podle zaměření různých databází. Jak je vidět, rozsah takto propojených znalostních bází je opravdu velký a pro účely tohoto projektu je velmi dobře využitelný.

3.2.3 Nástroje využívající znalostní báze

Mezi nástroje na detekci pojmenovaných entit s následným zjednoznačněním pomocí znalostních bází patří například AlchemyAPI AlchemyAPI, Inc. 2013 od AlchemyAPI, Inc. a OpenCalais Thomson Reuters Corporation 2013 od Thomson Reuters, detailnější přehled dostupných nástrojů je obsáhle zpracován v Olensky 2012.

AlchemyAPI

AlchemyAPI je webová služba (*SaaS API – Software as a Service Application Programming Interface*) stejnojmenné společnosti poskytující

2. V roce 2010 koupena společností Google, která Freebase použila jako jeden ze základních prvků při tvorbě svého Knowledge Graph.
3. vnitřně se jim říká *topics* a jsou k nim přidružené dodatečné informace včetně menší ontologie

širokou škálu nástrojů v oblasti zpracování přirozeného jazyka. Mezi hlavní nástroje patří rozpoznávání pojmenovaných entit a jejich zjednoznačnění pomocí URI z databází z Linked Data, rozpoznávání výroků (typu „Takové kroky rozhodně nemůžeme nechat bez odpovědi“ , řekl mluvčí společnosti), rozpoznávání klíčových slov, extrakce konceptů a další.

K nástroji se přistupuje přes web pomocí REST API (*Representational State Transfer*), kde veškerá komunikace probíhá pomocí HTTP dotazů. K dispozici je také několik knihoven (pro několik programovacích jazyků, mimo jiné také pro Python) zajišťující a zjednodušující tuto komunikaci. Vstupem pro aplikaci mohou být volně dostupná data na webové adrese, vlastní HTML kód, nebo přímo čistý text. Nástroj nabízí i extrakci a čištění textu z HTML stránky, takto získaný text je potom možné zařadit do výstupu nástroje společně se zbytkem výsledků. Výstup z aplikace je volitelně buď XML, JSON, nebo RDF.

3.3 Distributivní sémantika a Gensim

Mezi použitelné (a v některých aplikacích také používané) metody patří i distributivní sémantika, tou zde myslíme sémantickou analýzu zastoupenou především LSA (*Latentní Sémantická Analýza*) a LDA (*Latentní Dirichletova Alokace*). Tyto metody jsou určené k výpočtu podobnosti dokumentů a používají statistický přístup. Cílem této práce není detailní vysvětlení těchto technik, zaměřím se spíše na jejich základní principy a vlastnosti a omezení z nich vyplývající.

3.3.1 Vector Space Model

Distributivní sémantika nahlíží na dokumenty jako na množiny slov (anglicky *BoW – Bag of Words* a přeseji se jedná spíše o multimnožiny, protože nás zajímá počet výskytů daných slov, ztrácíme tak především informaci o pořadí slov v dokumentu). Základní princip je převedení dokumentu na vektor, skupina dokumentů je tedy zobrazena do vektorového prostoru (odtud *Vector Space Model – VSM*), kde je potom možné takto reprezentované dokumenty porovnávat. V tomto vektorovém prostoru odpovídá každá jeho dimenze jednomu ze slov

použitých v sadě dokumentů, případně jednomu slovu z použitého slovníku, hodnota vektoru v té dimenzi pak může být binarní (1 pokud se slovo vyskytuje v dokumentu, 0 jinak), častěji však četnost daného slova v dokumentu.

K zohledňování přidané informace každého slova je možné použít metodu TF-IDF (*Term Frequency – Inverse Document Frequency*), pak se místo četnosti uvádí hodnota vypočítaná jako frekvence slova v daném dokumentu vynásobená logaritmem podílu celkového počtu dokumentů ku počtu dokumentů obsahujících dané slovo. IDF část z TF-IDF váží frekvenci slova v dokumentech tím, jak často se v dokumentech vyskytuje, například u slov, které se vyskytují ve všech dokumentech, se hodnota úplně vynuluje.

Při výpočtu podobnosti je nutné vektory reprezentující dokumenty buď normalizovat, pak je možné použít euklidovskou vzdálenost, nebo je možné je rovnou porovnávat úhlově, k tomu se hodí cosinová míra, ta se také nejčastěji používá.

3.3.2 Latentní sémantická analýza

Vzhledem k velkému počtu dimenzí vektorového prostoru je vhodné tento počet co možná nejvíce snížit, k tomu se využívá několik technik, zde se nebudu zabývat těmi, které se dají řadit mezi předzpracování vstupních dat, ty budou uvedené v příslušné kapitole. Zde se zaměřím především na techniky nejen přispívající ke snížení dimenzionality, ale také k lepší abstrakci při porovnávání dat.

Jednou z těchto technik je LSA, zde už vektor není reprezentován jednotlivými slovy, ale spíše tematickými koncepty, občas se jim také říká jednoduše témata. Každé takové téma je reprezentováno několika slovy, které ho charakterizují a to s různými vahami. Jedno slovo je typicky součástí několika témat (v každém s jinou vahou, podle toho, nakolik se váže k ostatním slovům v tématu). Vektorový prostor má potom počet dimenzí odpovídající počtu vytvořených témat a dokumenty jsou následně porovnávány právě pomocí zastoupení těchto témat. V LSA je potřeba předem rozhodnout, kolik témat se pro zpracování vytvoří. Čím větší počet, tím jemnější rozdělení by se mělo vytvořit, naopak příliš malý počet nebude schopný dobře zachytit rozdíly u příbuznějších témat. LSA pomáhá abstrakci především tím, že například dva dokumenty popisující podobnou věc, ale

používající jiné termíny, zde mohou mít netriviální podobnost i když v klasickém TF-IDF by byla nulová, protože by nebyla použita stejná slova.

Jednotlivá témata jsou získána pomocí metody *Singular Value Decomposition* z matice dokumentů (například ve formě TF-IDF) a počtu požadovaných témat.

3.3.3 Latentní Dirichletova alokace

Z klasické LSA je potom možné vyvodit pLSA (pravděpodobnostní Latentní Sémantická Analýza) založenou více na teorii pravděpodobnosti než pouze na statistice a lineární algebře. Ta přináší stabilnější matematický základ a lepší výsledky, současně má ovšem problémy s přeučováním a není jasné jak určit pravděpodobnost dokumentu mimo trénovací množinu Blei, Ng a Jordan 2003.

Tyto problémy řeší LDA, podobně jako pLSA je založena na pravděpodobnosti, je potřeba si dopředu určit jaký počet témat chceme vytvořit. LDA využívá dvě statistická rozdělení, Dirichletovo a multinomické. Dirichletovo rozdělení s parametrem α, což je vektor jehož počet dimenzí odpovídá zvolenému počtu témat, modeluje výběr odpovídajících témat pro daný dokument (multinomické rozdělení pravděpodobností θ témat pro každý dokument). Hodnoty α na jednotlivých pozicích musí být menší než jedna (a většinou jsou stejné), abychom dostali efekt, kdy dané téma dokumentu buď odpovídá nebo neodpovídá (pravděpodobnost 1 nebo 0) a nic mezi tím. Podobně je to s Dirichletovým rozdělením modelujícím výběr slov pro jednotlivá témata, kde je parametr β, který tentokrát má počet dimenzí odpovídající velikosti slovníku a jehož hodnoty jsou vetšinou stejné a menší než jedna (chceme aby témata byla zastoupena spíše méně než více slovy). Zde se vytváří multinomické rozdělení pravděpodobností ϕ pro každé téma. Dále pro každou pozici slova i, j kde i označuje dokument a j pozici slova v tomto dokumentu se vybere na základě multinomického rozdělení θ_i pravděpodobnost tématu $z_{i,j}$, pro stejnou pozici se naopak z $\phi_{z_{i,j}}$ vybere příslušné slovo na této pozici $w_{i,j}$. Tato slova jsou jediné známé veličiny na začátku zpracování a na základě nich, se optimalizují ostatní parametry, aby se dosáhlo co nejvyšší pravděpodobnosti modelu. Samotná optimalizace je možná například Gibbsovým vzorkováním.

Následné zjišťování podobnosti mezi dokumenty už koresponduje s LSA, dokument je reprezentován zastoupením témat (vektorem témat), o kterých pojednává a parametr α zajišťuje, že těchto témat nebude příliš velké množství a tím pádem bude lépe možné od sebe dokumenty tematicky rozlišit (při zastoupení většího počtu témat u každého článku by mohlo hrozit, že si všechny dokumenty budou příliš podobné).

3.3.4 Gensim

Gensim Řehůřek a Sojka 2010 je pythonovská knihovna implementující několik technik distributivní sémantiky. Součástí jsou kromě pLSA všechny zde zmíněné metody a několik dalších. Jedna z hlavních výhod je kvalitní API s dobrou dokumentací a jednoduchost použití, další z výhod je dobrá škálovatelnost vstupních dat (myšleno především na čím dále větší data). Program zvládne běžet s konstantní náročností na operační pamět, velká data jsou průběžně ukládána na disk a zpracovávána po dávkách. Gensim obsahuje i metody potřebné pro základní předzpracování dat, práci se vstupním korpusem a tvorbu slovníku. Poskytuje tak kompletní funkcionalitu na zpracování dokumentů ve formátu čistého textu.

4 Vstupní data a jejich předzpracování

4.1 Charakteristika vstupních dat

Systém zpracovává anglické zpravodajské články stažené pomocí rss kanálů, k dispozici je i skript napojený na existující nástroje určené pro získání vlastní množiny článků na zpracování (uživatel si zvolí vlastní preferované rss kanály) včetně nástrojů na předzpracování takto získaných článků (především extraktor textu článku z webové stránky). Uživatel samozřejmě může dodat i vlastní soubor článků, který by chtěl zpracovat tímto způsobem. Jako základní formát dat na zpracování je plaintext v kódování UTF-8, je také nutné jasně vymezit nadpis článku.

4.2 Získávání dat

4.3 Předzpracování dat

4.4 Dodatečná zpracování potřebná pro využití distributivní sémantiky

5 Pokusy s distributivní sémantikou

6 Vlastní přístup k řešení problému

6.1 Zvolené nástroje

6.2 Způsob využití výstupů těchto nástrojů k získání výsledků

6.3 Vlastnosti systému

7 Vyhodnocení systému

8 Závěr

Bibliografie

[1] AlchemyAPI, Inc. *AlchemyAPI*. 2013. URL: www.alchemyapi.com.

[2] Steven Bird, Ewan Klein a Edward Loper. *Natural Language Processing with Python*. 1st. O'Reilly Media, Inc., 2009. ISBN: 0596516495, 9780596516499.

[3] Christian Bizer, Tom Heath a Tim Berners-Lee. „Linked Data - The Story So Far". In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 5.3 (břez. 2009). Ed. T. Heath, M. Hepp a C. Bizer, s. 1–22. ISSN: 1552-6283. DOI: 10.4018/jswis.2009081901. URL: http://dx.doi.org/10.4018/jswis.2009081901.

[4] Christian Bizer, Jens Lehmann et al. „DBpedia – A crystallization point for the Web of Data". In: *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 7 (zář. 2009), s. 154–165. ISSN: 1570-8268. URL: http://portal.acm.org/citation.cfm?id=1640848.

[5] David M. Blei, Andrew Y. Ng a Michael I. Jordan. „Latent dirichlet allocation". In: *Journal of Machine Learning Research* 3 (břez. 2003), s. 993–1022. ISSN: 1532-4435. URL: http://portal.acm.org/citation.cfm?id=944919.944937.

[6] Christopher Sahnwaldt. *DBpedia – About*. Zář. 2013. URL: http://dbpedia.org/About.

[7] Jenny Rose Finkel, Trond Grenager a Christopher Manning. „Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, červ. 2005, s. 363–370. URL: http://www.aclweb.org/anthology/P/P05/P05-1045.

[8] W3C SWEO Community Project. *Linking Open Data*. 2013. URL: http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.

[9] Marlies Olensky. „Market study on technical options for semantic feature extraction“. In: (dub. 2012). http://ec.europa.eu/information_society/apps/projects/logos//2/270902/080/deliverables/001_DeliverableD74MarketStudyTools.pdf.

[10] Thomson Reuters Corporation. *OpenCalais*. 2013. URL: http://www.opencalais.com.

[11] Lev Ratinov a Dan Roth. „Design challenges and misconceptions in named entity recognition“. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. CoNLL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, s. 147–155. ISBN: 978-1-932432-29-9. URL: http://dl.acm.org/citation.cfm?id=1596374.1596399.

[12] Radim Řehůřek a Petr Sojka. „Software Framework for Topic Modelling with Large Corpora“. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, 22. květ. 2010, s. 45–50.