Univerzita Hradec Králové

Fakulta informatiky a managmentu

Katedra informatiky a kvantitativních metod

Sémantická analýza textů

Bakalářská práce

Autor: Matěj Fries

Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Jiří Haviger, Ph.D.

Hradec Králové duben 2017

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne Matěj Fries

Poděkování

Rád bych poděkoval Mgr. Jiřímu Havigerovi, Ph.D. za pomoc při vedení bakalářské práce. Mé poděkování patří též všem, kteří mě při psaní práce podporovali.

Anotace

Tato bakalářská se zabývá sémantickou analýzou textů. V teoretické části práce jsou popsány a vysvětleny základní způsoby sémantické analýzy textů, především česky psaných. V úvodu práce jsou popsány základní možnosti reprezentace textů ve vektorových prostorech, jejich využití, výhody a nevýhody. V další části práce je rozebrána problematika vysoké dimenzionality těchto vektorových reprezentací, způsoby snižování počtu dimenzí a jejich uplatnění. V hlavní části práce je popsána a vysvětlena metoda nazvaná latentní sémantická analýza, která řeší problémy vektorové reprezentace textů včetně vysokého počtu dimenzí. Popsán je i matematický základ této metody. I když tato metoda řeší nedostatky modelu vektorového prostoru, sama má svoje nevýhody a proto byla zavedena další metoda a to pravděpodobnostní sémantická analýza, která je založena na statistickém modelu a je řešena maximalizací pravděpodobností. Tato metoda byla velkým krokem dopředu, avšak neúplným, protože má velké nevýhody zabraňující jejímu praktickému využití. Proto je v závěru teoretické části popsána a vysvětlena metoda latentní Dirichletovy alokace, která řeší nevýhody předchozí metody a je z těchto metod nejvíce využívána.

Součástí práce je program v jazyku Java, ve kterém jsou popisované způsoby implementovány a v praktické části práce popsány na ukázkových datech. V praktické části je na dvou ukázkových souborech dat ukázána výkonost těchto postupů a jejich implementace. V závěru práce jsou rozebrány výsledky praktických využití práce.

Annotation

Title: Semantic Text Analysis

This Bachelor Thesis is about semantic text analysis. In theoretical part of this Thesis are described and explained basic ways of semantic text analysis, mainly Czech written text. In the beginning are described basic ways of representing text in vector spaces, their use, pros and cons. In next part is analysed problem of high dimension count of these representations, ways of lowering dimension count and their uses. In the main part of this Thesis is described and explained method called Latent Semantic Analysis. This method solves problems of vector space models including high dimension count. Mathematical principle used by this method is also described. Even though this method solves cons of vector space models, it has cons of its own. Therefore new method was introduced called Probabilistic Latent Semantic Analysis. This method is based on statistic model and it's solved by probability maximization. This method was big improvement, but incomplete because it has large drawback preventing practical use. Because of this is in the end of theoretical part explained method called Latent Dirichlet Allocation. This method solves these drawbacks and it's the most used method of these.

Implementation of these described ways of analysis is part of this Thesis written in Java and described use of demonstration data in practical part. In practical part is showed performance of these methods and their implementation on example sets of data. At the end of the Thesis are discussed results of practical uses.

Obsah

1 Úvod

Lidé používají počítače v dnešní době stále více a více, ale jejich celou sílu nevyužívají. Jedním z největších omezeních, zabraňujících jejich celou sílu využít, jejich malá schopnost rozumět smyslu lidského jazyka. Podle P. Turneye a P. Pantela (2010) je technologie webových vyhledávačů jen zavaděním o lidský jazyk, ale přesto je její dopad na společnost a ekonomiku nesmírný.

Sémantickou analýzou textů zde rozumíme převod lidmi psaného textu do datové podoby tak, aby mohl být počítačově zpracován a analyzován. Využití strojově zpracovatelného textu je mnoho, ať již zmíněné webové či jiné vyhledávání podobných dokumentů, cílení reklamy, doporučování, kontrola plagiátorství apod. Sémantická analýza slouží k identifikaci témat textových dokumentů a určení tematické podobnosti různých dokumentů. Díky identifikaci témat můžeme spolu porovnávat i dokumenty, které spolu žádná slova nesdílí.

K základní reprezentaci textů slouží model vektorového prostoru, který je popsán v druhé kapitole této práce. Každý dokument v tomto prostoru je reprezentován vektorem. Hodnoty tohoto vektoru se dají určovat vícero různými způsoby, každý se svými výhodami a s jiným využitím. Vektory v tomto vektorovém prostoru, které jsou si blíže, jsou si více sémanticky podobné a naopak. Největší nevýhodou reprezentace v tomto prostoru je jeho velikost, kde každá dimenze odpovídá jednomu slovu.

Ke snížení velikosti počtu dimenzí a zmenšení vektorového prostoru a tím i snížení výpočetní náročnosti slouží několik postupů, popsané ve třetí kapitole.

I přes tyto postupy počet dimenzí neklesne o příliš velký počet, a tak je nutné k tomuto problému přistoupit jiným způsobem. Prvním takovým způsobem je metoda nazvaná latentní sémantická analýza, která využívá lineární algebru k redukci počtu témat a strojovému vytvoření témat, pomocí kterých dokumenty reprezentuje. Této metodě a jejím rozšířením je věnována třetí kapitola.

Dalším způsobem je pravděpodobnostní latentní sémantická analýza, která je založena na statistickém modelu. Této metodě je věnována podkapitola 4.3, kde je popsána a vysvětlena. I když pravděpodobnostní latentní sémantická analýza dosahuje lepších výsledků než latentní sémantická analýza, má své nedostatky

a to zejména zvětšující se počet parametrů se zvětšující se kolekcí dokumentů a nemožnosti porovnávání dokumentů s novými dokumenty neobsaženými v trénovací kolekci.

Její nedostatky řeší latentní Dirichletova alokace, které se věnuje podkapitola 4.4. Tato metoda řeší oba hlavní nedostatky pravděpodobnostní latentní sémantické analýzy a proto je z těchto metod nejpoužívanější metodou sémantické analýzy prvků.

Součástí této práce je implementace těchto popsaných metod v jazyce Java. Na této implementaci a ukázkových datech jsou všechny popsané postupy prakticky předvedeny, nejprve na menší kolekci dokumentů, aby byly vidět všechny vlastnosti popisovaných způsobů a poté na větších datech s reálnější vypovídací hodnotou. Program není optimalizovaný pro rozsáhlá data, ale výsledky na opravdu velkých kolekcích jsou rozebírány z jiných zdrojů.

2 Vektorová reprezentace textu

K reprezentaci textu ve vektorech se využívá model vektorového prostoru (Vektor Space Model), jehož účelem je reprezentovat textové dokumenty jako body v prostoru, tedy vektory ve vektorovém prostoru. Vektory, které jsou v tomto prostoru blíže, jsou sémanticky podobné a body vzdálenější, jsou si i sémanticky vzdálenější (Turney, Pantel 2010). Dokumentem může být jakýkoliv text, obsah webové stránky, knihy, článku, ale i dotaz ve webovém vyhledávači obsahující jen několik slov. Z kolekce dokumentů je vytvořen vektorový prostor, kde jedna dimenze odpovídá jednomu slovu (může být i spojení slov, fráze nebo věta).

2.1 Váha prvků

Vektor ve vektorovém prostoru reprezentuje jeden dokument a nabývá hodnot podle výskytů slov v tomto dokumentu. Tyto hodnoty můžeme doplnit binárně, hodnota prvku ve vektoru bude nula, pokud se slovo v dokumentu nevyskytuje, nebo bude jedna, pokud se slovo v dokumentu vyskytuje jednou nebo vícekrát. „Tato binární reprezentace dokumentů je jednoduchá a v mnoha případech i dostačující," (Materna (1) 2011) ale ve větších kolekcích dokumentů k dosažení lepší přesnosti porovnávání tato reprezentace nedostačuje.

Více vypovídající reprezentace dokumentů je četnost slov (Term Frequency, TF), kde hodnota prvku ve vektoru odpovídá počtu výskytů slova v dokumentu. Nevýhodou u této reprezentace je vysoká váha tzv. stop slov, což jsou slova s častým výskytem, ale malou vypovídací hodnotou jako předložky, spojky, sponová a modální slovesa apod. (Materna (1) 2011).

Nejpopulárnější reprezentací řešící tento problém je TF-IDF, tedy četnost slov násobené inverzní četností dokumentů (Term Frequency * Inverse Document Frequency) (Turney, Pantel 2010). Hodnoty se získávají součinem četnosti slov s převrácenou četností dokumentů, která se určí logaritmem podílu počtu dokumentů ku počtu dokumentů obsahujících dané slovo (Materna (1) 2011). Čímž se dosáhne efektu, kde slova, která se vyskytují v málo dokumentech, budou mít nejvyšší hodnoty, oproti tomu slova vyskytující se ve všech dokumentech, budou mít hodnotu nulovou. Aby nulovou hodnotu měla jen slova, která se v dokumentu nevyskytují, upraví

se výpočet IDF tak, že k podílu počtu dokumentů ku počtu dokumentů, ve kterých se slovo vyskytuje se přičte jednička v logaritmu (Husbands, Simon, Ding 2005).

1. TF-IDF = TF(*i*) * log(n / DF(*i*))
2. TF-IDF = TF(*i*) * log(1 + n / DF(*i*))

TF(i) je četnost slova *i*, *DF(i)* je počet dokumentů ve kterých se slovo *i* vyskytuje a *n* je počet dokumentů.

Dalším způsobem reprezentace, často spojovaný s TF-IDF je délková normalizace (lenght normalization), která se používá, protože vyhledávače mají tendence favorizovat delší dokumenty. Délková normalizace tyto tendence opravuje (Turney, Pantel 2010).

Další možností je log.entropy, která dosahuje podobných výsledků jako TF-IDF, ale ve větších kolekcích bývá méně přesná (Husbands, Simon, Ding 2005).

2.2 Porovnávání vektorů

K zjištění podobnosti dokumentů s vyhledávaným řetězcem je potřeba vektory porovnat. Nejpoužívanějším způsobem porovnávání vektorů je kosinová podobnost (Turney, Pantel 2010). Ta je dána skalárním součinem vektorů vyděleným součinem jejich velikostí, tedy vektory se porovnávají podle kosinu úhlu, který spolu svírají, a jejich délka nehraje žádnou roli.

Výsledná kosinová podobnost může nabývat hodnot od -1 do 1. Vektory s podobností 1 jsou identické, svírají úhel 0°, vektory s podobností 0 jsou kolmé, tedy svírají úhel 90° a vektory s podobností -1 jsou opačné s úhlem 180°. Pokud pracujeme s TF nebo TF-IDF vektory nemají záporné hodnoty a tedy podobnosti budou dosahovat hodnot jen od 0 do 1, ale například PMI, což je alternativa k TF-IDF může nabývat záporných hodnot (Turney, Pantel 2010).

Další možností porovnávání vektorů, která počítá s délkou vektorů je euklidovská vzdálenost. Možností porovnávání je podle Truneyho a Pantela (2012) více, ale nejpoužívanější je kosinová podobnost.

2.3 Vyhledávání

U menších kolekcí a při menším počtu dimenzí, není vyhledání nejpodobnějších dokumentů výpočetně náročné, ale při větším počtu dokumentů a slov je potřeba vyhledávání usnadnit.

U větších kolekcích dokumentů bude většina dokumentů reprezentována dlouhými vektory s většinou prvků o nulové hodnotě. Při součinu těchto vektorů záleží jen na hodnotách, které jsou alespoň u jednoho vektoru nebo u obou nenulové a můžeme vybrat jen ty (Turney, Pantel 2010).

J. Materna ((2) 2011) doporučuje použití struktur jako R-tree nebo M-tree, pro dosažení sublineární složitosti (není nutné porovnávat se všemi dokumenty z kolekce).

3 Dimenzionalita

„Jedním z největších problémů reprezentace objektů pomocí vektorů jsou vysoké dimenze (každé dimenzi odpovídá právě jedno slovo, proto je počet dimenzí v řádu statisíců) je takzvané „prokletí dimenzionality““ (Materna (2) 2011). Vysoký počet dimenzí stěžuje porovnávání dokumentů a zvyšuje výpočetní náročnost. Ke snížení počtu dimenzí existují dva hlavní přístupy, výběr příznaků a extrakce příznaků (Materna (3) 2011).

Výběr příznaků označuje postupy, kdy se vybírají jen ty příznaky, které jsou nejvýznamnější a ty zůstanou nezměněny, oproti tomu extrakce příznaků nahrazuje původní příznaky novými, jejichž počet je menší, ale vypovídací hodnoty zůstanou téměř nezměněny.

3.1 Výběr příznaků

Princip výběru příznaků spočívá v tom, že se z množiny příznaků vyberou jen ty významné, které zůstanou nezměněné, a nevýznamné příznaky, v našem případě slova, se zanedbají. Jak již bylo zmíněno u četnosti slov, existují tzv. stop slova, která se vyskytují velice často, ale nemají skoro žádnou vypovídací hodnotu (neplnovýznamová slova). Tyto slova, jako spojky předložky, apod. můžeme při porovnávání dokumentů z našich dimenzí odebrat, tím však počet dimenzí příliš nesnížíme (Materna (3) 2011).

Další možností je odebrání slov, které se vyskytují jen ve velmi nízkých počtech, většinou se jedná o chyby a překlepy a můžeme snížit počet dimenzí o velké množství (Materna (3) 2011). Ale může se například jednat o odborné termíny nebo jména, která se vyskytují málo, ale významovou hodnotu mají vysokou, proto se tento postup uplatňuje podle určení aplikace.

3.2 Extrakce příznaků

Nejjednodušším příkladem extrakce příznaků je převedení slov na malé písmena, tím sjednotíme slova, která jsou na začátku vět s těmi ve větách. Ale také můžeme změnit význam jiných slov, například příjmení a názvů (Turney, Pantel 2010).

Nejčastějším případem extrakce příznaků je lemmatizace nebo stematizace.

Stematizace znamená převedení slova na jeho kmen, tento způsob je výhodný například pro angličtinu, ale pro český jazyk je lepší lemmatizace (Materna (3) 2011).

Lemmatizací se převede slovo na jeho základní tvar, což se v češtině většinou využívá.

Problémem lemmatizace a stematizace je víceznačnost, proto se většinou ke slovu ještě přidává tag, který zpřesňuje význam slova, nejčastěji slovní druh, tím se však počet dimenzí opět navyšuje, proto se tento postup používá u latentních sémantických analýz.

4 Latentní sémantická analýza

V případě velkých kolekcích dokumentů s velkým počtem slov ani po lemmatizaci nesnížíme počet dimenzí na přijatelnou úroveň. Snížení počtu dimenzí o vysoký počet je hlavní důvod používání latentních sémantických analýz.

Pokud vezmeme všechny vektory dokumentů z dané kolekce a vytvoříme z nich matici, dostaneme řídkou matici, která bude mít stejný počet řádků jako počet slov a počet sloupců jako počet dokumentů. Latentní sémantická analýza využívá rozklad na singulární hodnoty (Singular Value Decomposition, SVD), pomocí kterého převede tyto vektory v matici do prostoru s nižším počtem dimenzí (Husbands, Simon, Ding 2005). Dimenze v tomto novém prostoru budou tvořit koncepty, tedy tematické oblasti vybrané automaticky, které budou mít u každého slova váhu, kterou toto slovo odpovídá danému tématu (Materna (4) 2011).

4.1 Rozklad na singulární hodnoty

Rozklad na singulární hodnoty mimo jiné umožňuje aproximaci původních dat pomocí menšího počtu dimenzí, díky čemu lze použít k redukci dat bez toho, abychom přišli o významné informace, které nám dávají (Baker 2013). Pomocí tohoto rozkladu můžeme libovolnou obdélníkovou nebo čtvercovou matici rozdělit na součin tří jiných matic (Caid, Dumais, Gallant 1995).

- $X = U\,S\,V^T$

X značí rozkládanou matici o *m* řádcích a *n* sloupcích. Matice *S* je diagonální matice a matice *U* a *V* jsou ortogonální matice. Matice *U* má *m* řádků a sloupců, matice *S* má *m* řádků a *n* sloupců a transponovaná matice *V* má *n* řádků a sloupců. Součinem těchto matic dostaneme opět původní matici *X*.

Pro výpočet matice *U* musíme nejprve původní matici *X* násobit její transponovanou maticí, tedy XX^T. Z výsledné matice pomocí rovnosti $A\vec{v} = \lambda\vec{v}$ získáme vlastní čísla matice λ, pomocí kterých získáme vlastní vektory matice $\vec{v}$. Vlastní vektory seřadíme podle velikosti příslušných vlastních čísel a vytvoříme z nich matici, kde tyto vektory budou tvořit její sloupce. Pro získání matice *U* je potřeba převést tuto matici na ortogonální matici.

Výpočet matice V je podobný, transponovanou původní matici vynásobíme původní maticí, tedy X^TX. Z této matice opět získáme vlastní čísla a příslušné vlastní vektory, které opět seřadíme podle hodnoty vlastních čísel do matice, kde vlastní vektory budou tvořit její sloupce. Pro získání matice V stačí z výsledné matice vytvořit matici ortogonální a jejím transponováním získáme hledanou matici V^T.

Hodnoty matice diagonální matice S budou odmocniny nenulových vlastních čísel, která jsou u obou matic stejná. Tyto seřazené odmocniny vytvoří diagonálu matice S.

Hodnoty v matici S se nazývají singulární hodnoty (singular values), sloupce v matici U se nazývají levé singulární vektory a sloupce v matici V nebo řádky v matici V^T se nazývají pravé singulární vektory.

Tento postup byl napsán s pomocí práce K. Bakera (2013), kde je vysvětlen podrobněji i s příklady.

4.2 Využití rozkladu

Pokud vezmeme matici četnosti slov v dokumentech, ať již s hodnotami četnosti slov, TF-IDF nebo jiné, a rozložíme ji pomocí rozkladu na singulární hodnoty, můžeme snížit počet dimenzí na menší počet tak, že vezmeme největší singulární hodnoty diagonální matice S a zbylé a s příslušnými levými a pravými singulárními vektory smažeme (Caid, Dumais, Gallant 1995).

- $X' = U' S' V'^T$

Po této úpravě bude diagonální matice S' mít jen k řádků a sloupců, kde k je počet singulárních hodnot, které necháme v matici a také počet témat či dimenzí, které budou tvořit sémantický prostor. Matice U' bude mít stále m řádků, ale k sloupců a její řádkové vektory budou reprezentovat slova a jejich váhu pro dané téma. Matice V'^T bude mít k řádků a n sloupců a její sloupcové vektory budou reprezentovat dokumenty a jejich váhu pro dané téma. „Lze matematicky dokázat, že chyba, ke které dojde při zanedbání konceptu s minimální vahou, je nejmenší možná.“ (Materna (4) 2011)

Pro porovnávání dokumentů je potřeba jejich vektory vynásobit singulárními hodnotami z matice S' a poté je možné je porovnávat jako v kapitole 2.2, tedy nejčastěji kosinovou podobností.

Vzhledem ke snížení počtu dimenzí přestávají být slova nezávislá, pokud jsou dvě slova používaná v podobných dokumentech, budou mít i slova podobné vektory. Takže dokumenty si mohou být podobné i když neobsahují stejná slova (Caid, Dumais, Gallant 1995). Podobnost slov můžeme porovnávat, podobně jako u dokumentů, porovnáváním vektorů slov vynásobených singulárními hodnotami.

Práce se synonymy je výhodou latentní sémantické analýzy, nevýhodou je problém s víceznačností, protože latentní sémantická analýza bere všechny stejně zapsaná slova za stejná. Další výhodou je, že díky redukci dimenzí vznikají menší nepřesnosti, které eliminují šum a chyby v datech (Materna (4) 2011).

Optimální hodnota *k*, určující počet témat, se liší podle obsahu kolekce dokumentů a její velikosti a většinou se určuje empiricky. Kvůli vyšší výpočetní náročnosti latentní sémantické analýzy se často upřednostňují menší hodnoty *k* (Kontostathsis, Pottenger 2006).

Výsledky latentní sémantické analýzy jsou většinou lepší než u původního vektorového prostoru, nebo alespoň stejně dobré. V některých případech jsou výsledky dosažené s latentní sémantickou analýzou až o 30% lepší (Caid, Dumais, Gallant 1995). Podle práce Husbandse, Simona a Dinga (2005) jsou výsledky latentní sémantické analýzy u větších kolekcí nedostatečné a často dochází ke snižování váhy méně častým slovům, čímž dochází k negaci efektů TF-IDF, pro řešení tohoto problému doporučují svou normalizovanou latentní sémantickou analýzu.

4.3 Pravděpodobnostní latentní sémantická analýza

Na rozdíl od latentní sémantické analýzy využívá pravděpodobností latentní sémantická analýza, jak již její název napovídá, pravděpodobností k porovnávání dokumentů. Jejím základem je statistický model (Hofmann 1999). Pravděpodobnostní latentní sémantická analýza vypočítává relevantní pravděpodobnosti vybíráním hodnot, které maximalizují pravděpodobnost kolekce dokumentů. Tedy určuje maximální pravděpodobnost, kterou odhaduje pomocí algoritmu předpokladu a maximalizace (Expectation Maximization, EM) (Farahat, Chen 2006).

4.3.1 Model latentních hodnot

Model latentních hodnot v pravděpodobnostní latentní sémantické analýze počítá s třemi soubory proměnných, a to s dokumenty *d*, jejichž počet označíme jako N_d, se slovy *s*, jejichž počet označíme jako N_s a s tématy nebo dimenzemi *t*, jejichž počet podobně označíme jako N_t (Oneata 2016).

Podmíněná pravděpodobnost, že je slovo *s* obsaženo v dokumentu *d* je:

$$P(s|d) = \sum_t^{Nt} P(s|t) * P(t|d)$$

Pravděpodobnost dokumentu *d* a slova *s* je:

$$P(d,s) = P(d) * P(s|d)$$

$$= P(d) * \sum_t^{Nt} P(s|t) * P(t|d)$$

A pomocí Bayesova pravidla může být zapsána jako (Farahat, Chen 2006):

$$P(d,s) = \sum_t^{Nt} P(s|t) * P(d|t) * P(t)$$

Neznámé jsou v tomto případě pravděpodobnosti *P(s|t)*, *P(d|t)* a *P(t)*, tedy pravděpodobnost slova *s* v tématu *t*, pravděpodobnost dokumentu *d* v tématu *t* a pravděpodobnost samotného tématu *t*. Hodnoty těchto proměnných budou dosazeny v algoritmu předpokladu a maximalizace s pravděpodobnostní funkcí pravděpodobnosti kolekce dokumentů *C* (Materna (5) 2011):

$$\log[P(C)] = \sum_d^{Nd} \sum_s^{Ns} n(s,d) * \log[P(s|d)]$$

P(C) značí pravděpodobnost celé kolekce dokumentů, *n(s,d)* je hodnota z matice četnosti slov nebo TF-IDF kolekce dokumentů.

4.3.2 Algoritmus předpokladu a maximalizace

Algoritmus předpokladu a maximalizace (Expectation Maximization, EM algoritmus) se dělí na dva kroky, první krok předpokladu (E) vypočítá z pravděpodobností latentní proměnou, pomocí které v kroku maximalizace (M) upraví pravděpodobnosti (Hofmann 1999).

Latentní pravděpodobnostní proměnná *P(t|d,s)* nám označuje pravděpodobnost tématu *t* za předpokladu vybrání dokumentu *d* a slova *s*. Tuto proměnou vypočítáme v prvním kroku algoritmu pomocí rovnice (Hofmann 1999):

$$P(t|d,s) = \frac{P(t) * P(d|t) * P(s|t)}{\sum_t^{Nt} P(t) * P(d|t) * P(s|t)}$$

Pomocí této latentní proměnné a hodnot z matice četnosti slov v dokumentech vypočítáme nové hodnoty pravděpodobností (Hofmann 1999):

$$P\ (s|t) = \frac{\sum_{d}^{Nd} n(s,d) * P(t|d,s)}{\sum_{d,s}^{Nd,Ns} n(s,d) * P(t|d,s)}$$

$$P(d|t) = \frac{\sum_{s}^{Ns} n(s,d) * P(t|d,s)}{\sum_{d,s}^{Nd,Ns} n(s,d) * P(t|d,s)}$$

$$P(t) = \frac{\sum_{d,s}^{Nd,Ns} n(s,d) * P(t|d,s)}{\sum_{d,s}^{Nd,Ns} n(s,d)}$$

Původní hodnoty těchto pravděpodobností se mohou nastavit náhodně a poté je algoritmus předpokladu a maximalizace upraví při každém svém průběhu. Algoritmus se zastavuje při dosažení maximální pravděpodobnosti *P(C)*, případně *log[P(C)]*, toto maximum však nemusí být globální, ale může se jednat pouze o lokální maximum (Hofmann 1999).

Podobně, jako u latentní sémantické analýzy, můžeme pak výsledné pravděpodobnosti zobrazit jako tři matice, z toho jednu diagonální (Oneata 2016).

- A = L U R

Matice *L* o bude obsahovat hodnoty *P(d|t)* a bude mít N_d řádků a N_t sloupců. Diagonální matice *U* bude mít na diagonále hodnoty *P(t)* a N_t řádků a sloupců. A matice *R* bude obsahovat hodnoty *P(s|t)* s N_t řádky a N_s sloupci. Vektory dokumentů budou v tomto případě řádkové vektoru matice *L*. Výsledná matice *A* je vzhledem k jinému postupu jiná než matice, která by se získala stejným postupem v latentní sémantické analýze (Oneata 2016).

Počáteční hodnoty pravděpodobností silně ovlivňují výsledky pravděpodobnostní sémantické analýzy. Vzhledem k podobnosti výsledků s výslednými maticemi v latentní sémantické analýze se můžou počáteční hodnoty získat její pomocí, ale s určitými úpravami (Farahat, Chen 2006).

Hofmann (1999) také uvádí postup pro optimalizaci EM algoritmu pomocí temperace, kde se výpočty v EM algoritmu mocní parametrem β, který je na začátku algoritmu nastaven na jedna, tedy výsledky budou stejné, jako bez jeho použití, ale po průběhu EM algoritmu se parametr β změní o malou hodnotu a zkusí se, jestli výsledky se změněným parametrem jsou lepší.

Výsledné vektory dokumentů můžeme opět porovnávat pomocí kosinové podobnosti, ale J. Materna ((5) 2011) doporučuje Hellingerovu vzdálenost nebo KL-divergenci.

Výsledky pravděpodobnostní latentní sémantické analýzy jsou oproti latentní sémantické analýze výrazně lepší, ale jejími nevýhodami je nemožnost predikce vektorů témat u nových dokumentů, které nebyly v původní kolekci dokumentů a tendence k přeučování, neboť s rostoucím počtem dokumentů v kolekci roste i počet parametrů, které je třeba odhadnout (Materna (5) 2011).

4.4 Latentní Dirichletova alokace

Všechny předchozí modely byly založeny na modelu balík slov (Bag-Of-Words), ve kterém se předpokládá, že pozice a pořadí slov nemá význam a slova můžou být vyměněna (Blei, Ng, Jordan 2003). Ale pokud chceme brát slova a dokumenty za vyměnitelné, měl by se tomu přizpůsobit i model, který by toto reprezentoval. Základní myšlenkou latentní Dirichletovy alokace (Latent Dirichlet allocation, LDA) je, že dokumenty jsou reprezentovány jako náhodné směsice latentních témat, kde každé téma je reprezentováno přidělenými slovy (Blei, Ng, Jordan 2003).

Latentní Dirichletova alokace je založena na pravděpodobnostní latentní sémantické analýze, na rozdíl od ní má však fixní počet parametrů, jejichž počet neroste s velikostí dokumentů ani kolekce (Materna (6) 2011). Latentní Dirichletova alokace využívá generativní model a podobně jako pravděpodobnostní latentní sémantická analýza maximalizuje pravděpodobnosti kolekce, ale na rozdíl od ni využívá Gibbsova vzorkování.

4.4.1 Generativní model a Gibbsovo vzorkování

Generativní model pomocí latentních proměnných počítá společnou pravděpodobnost. Tuto společnou pravděpodobnost upravuje tak, aby maximalizovala pravděpodobnost kolekce dokumentů. Generativní model u latentní Dirichletovy alokace se skládá ze tří kroků (Materna (6) 2011):

1. Pro každý dokument z kolekce dokumentů vyber parametry multinomického rozdělení $\theta(i)$ z Dirichletova rozdělení s parametry α.

2. Pro každou pozici slova v dokumentu vyber z Dirichletova rozdělení s parametry *θ(i)* téma *t(i,j)*.
3. Pro každou pozici *(i,j)* vyber slovo *s(i,j)* z multinomického rozdělení *φ[t(i,j)]* s parametry *β*.

Obě rozdělení, multinomické i Dirichletovo, mají *k* dimenzí, kde *k* je námi určený počet témat. Parametr α a β jsou tzv. hyperparametry modelu latentní Dirichletovy alokace.

V prvním kroku tedy vybereme pro každý dokument pravděpodobnosti pro všechna témata. S hodnotami hyperparametrů menšími než jedna zajistíme, že s největší pravděpodobností bude mít dokument jen několik málo témat nezanedbatelnou pravděpodobnost. Ve druhém kroku přiřadíme každé slovní pozici téma a ve třetím kroku vybereme slovo s největší pravděpodobností (Materna (6) 2011).

Pravděpodobnost celé kolekce, kterou chceme maximalizovat, získáme podle vzorce (Blei, Ng, Jordan 2003):

$$P(C|\alpha,\beta) = \prod_d^{Nd} \int P(\theta_d|\alpha) \left[\prod_n^{Nd} \sum_{t_{dn}} P(t_{dn}|\theta_d) P(s_{dn}|t_{dn},\beta)\right] d\theta_d$$

C označuje celou kolekci dokumentů, parametry α a β se určují při generování kolekce. Hodnoty θ se přepočítávají pro každý dokument a hodnoty *z* a *s* se určují pro každé slovo v každém dokumentu (Blei, Ng, Jordan 2003).

Vstupem není jako v předchozích případech matice četnosti slov, ale vektory dokumentů, který má každý dimenzí stejný počet, jako je počet slov v dokumentech a jedna hodnota vektoru reprezentuje pořadové číslo slova z množiny všech slov v kolekci dokumentů. Kvůli tomu není možné použít TF-IDF nebo jiné metody pro lepší vážení slov a je lepší stop slova z dokumentů odstranit.

Vektory dokumentů se získají z hodnot θ, které jsou již uspořádány v matici vektorů dokumentů a jejich hodnot, které určují váhy daných témat. Porovnání vektorů dokumentů je možné opět kosinovou mírou nebo podobně jako u pravděpodobnostní latentní sémantické analýzy.

Díky generativnímu přístupu latentní Dirichletovi alokace je možné, na rozdíl od pravděpodobnostní latentní sémantické analýzy, odhadovat témata i pro nové dokumenty, neobsažené v původní kolekci (Materna (6) 2011).

Díky těmto výhodám, menší výpočetní náročnosti a lepším výsledkům než u předchozích možností se v praktických aplikacích používá převážně latentní Dirichletova alokace (Materna (6) 2011).

5 Ukázka na datech

Pro názornou ukázku popisovaných postupů sémantické analýzy jsem implementoval popisované postupy v jazyku Java. Vstupem do programu jsou již lemmatizované česky psané texty v souborech ve formátu csv. Jako vstup do programu se dají použít i neupravené texty v českém jazyce nebo jiném jazyce, ale jejich použití by mohlo zvýšit výpočetní náročnost a velmi snížit výsledné podobnosti.

Program není určen pro rozsáhlé kolekce dokumentů a obsáhlé dokumenty. Implementované algoritmy nejsou optimalizovány pro větší výpočetní rychlost ani menší náročnost.

První podkapitola se bude věnovat menší kolekci dokumentů pro ukázání základních principů algoritmů. Ve druhé podkapitole bude použita větší kolekce dokumentů a rozebrány výsledky na rozsáhlých kolekcích.

5.1 Menší kolekce dokumentů

Pro první ukázku si vezmeme kolekci skládající se ze čtyř dokumentů, ve kterých je v každém jedna věta:

1. Brokolici si dám v pátek.
2. Kočka si ráda spí v krabici.
3. Kočky si nikdy nesní brokolici.
4. Počasí v naší oblasti si nedá říct!

V lemmatizované podobě budou dokumenty vypadat takto:

1. brokolice si dát v pátek
2. kočka si rád spí v krabici
3. kočka si nikdy nejíst brokolice
4. počasí v naší oblast si nedát říkat

Kdybychom těmto větám chtěli přiřadit témata, tak u prvního dokumentu by se dalo říct, že pojednává o jídle, druhý, že pojednává o kočkách, třetí pojednává jak o kočkách, tak i o jídlu a čtvrtý o počasí.

5.1.1 Model vektorového prostoru

Po převedení těchto dokumentů do vektorového prostoru, dostaneme 16 jedinečných slov, které abecedně seřazené vypadají takto:

- brokolice, dát, kočka, krabice, naší, nedát, nejíst, nikdy, oblast, pátek, počasí, rád, si, spát, v, říkat

Matice četnosti slov (TF), by vypadala takto:

1. brokolice 1 0 1 0
2. dát 1 0 0 0
3. kočka 0 1 1 0
4. krabice 0 1 0 0
5. naší 0 0 0 1
6. nedát 0 0 0 1
7. nejíst 0 0 1 0
8. nikdy 0 0 1 0
9. oblast 0 0 0 1
10. pátek 1 0 0 0
11. počasí 0 0 0 1
12. rád 0 1 0 0
13. si 1 1 1 1
14. spát 0 1 0 0
15. v 1 1 0 1
16. říkat 0 0 0 1

Každý řádek v této matici odpovídá danému slovu z kolekce uspořádaných abecedně. Sloupec této matice pak reprezentuje vektor dokumentu. Tato matice má většinu hodnot nulovou.

Vidíme tedy, že první a třetí dokument spolu sdílí slovo *brokolice (1)*, druhý a třetí slovo *kočka (3)*. Všechny dokumenty sdílí slovo *si (13)* a až na třetí dokument slovo *v (15)*.

Podle našeho přiřazení témat bychom řekli, že jsou si podobné pouze první a třetí dokument a druhý a třetí dokument. Jejich kosinová podobnost (zaokrouhlená) je v tomto vektorovém prostoru:

- 1. a 3. 0,4000
- 2. a 3. 0,3651

První a poslední dokument, které spolu tematicky nesouvisí, mají podobnost:

- 1. a 4. 0,3381

Tedy skoro stejně velkou podobnost jako tematicky související dokumenty, protože spolu sdílejí dvě stop slova. Tento problém by měla vyřešit reprezentace dokumentů pomocí TF-IDF, podle J. Materny ((1) 2011) dána:

- TF-IDF = TF(i) * log(n / DF(i))

Pomocí této rovnice popsané podrobněji v kapitole 2.1 získáme tuto matici:

1. brokolice 0,30 0,00 0,30 0,00
2. dát 0,60 0,00 0,00 0,00
3. kočka 0,00 0,30 0,30 0,00
4. krabice 0,00 0,60 0,00 0,00
5. naší 0,00 0,00 0,00 0,60
6. nedát 0,00 0,00 0,00 0,60
7. nejíst 0,00 0,00 0,60 0,00
8. nikdy 0,00 0,00 0,60 0,00
9. oblast 0,00 0,00 0,00 0,60
10. pátek 0,60 0,00 0,00 0,00
11. počasí 0,00 0,00 0,00 0,60
12. rád 0,00 0,60 0,00 0,00
13. si 0,00 0,00 0,00 0,00
14. spát 0,00 0,60 0,00 0,00
15. v 0,12 0,12 0,00 0,12
16. říkat 0,00 0,00 0,00 0,60

Jak vidíme, slovo *si (13)* má u všech dokumentů nulovou hodnotu a slovo *v (15)* výrazně nižší než ostatní slova. Díky této reprezentaci podobnosti dokument budou vypadat takto:

- 1. a 3. 0,1044
- 2. a 3. 0,0871
- 1. a 4. 0,0127

Kvůli menším hodnotám slov jsou hodnoty podobností také nižší, ale podobnost tematicky nesouvisejících dokumentů 1 a 4 výrazně poklesla oproti podobnosti tematicky souvisejících dokumentů.

Pokud bychom hypoteticky v každém dokumentu měli navíc nějaké plnovýznamové slovo, hodnota tohoto slova by v každém dokumentu byla také nula, proto v práci Husbandse, Simone a Dinga (2005) je upraven výpočet TF-IDF:

- TF-IDF = TF(*i*) * log(1 + n / DF(*i*))

Díky tomuto výpočtu dostaneme matici:

1. brokolice 0,48 0,00 0,48 0,00
2. dát 0,70 0,00 0,00 0,00
3. kočka 0,00 0,48 0,48 0,00
4. krabice 0,00 0,70 0,00 0,00
5. naší 0,00 0,00 0,00 0,70
6. nedát 0,00 0,00 0,00 0,70
7. nejíst 0,00 0,00 0,70 0,00
8. nikdy 0,00 0,00 0,70 0,00
9. oblast 0,00 0,00 0,00 0,70
10. pátek 0,70 0,00 0,00 0,00
11. počasí 0,00 0,00 0,00 0,70
12. rád 0,00 0,70 0,00 0,00
13. si 0,30 0,30 0,30 0,30
14. spát 0,00 0,70 0,00 0,00
15. v 0,37 0,37 0,00 0,37
16. říkat 0,00 0,00 0,00 0,70

Hodnoty často vyskytujících se slov jsou nižší, ale nulové jsou pouze ty hodnoty, kde se slova v dokumentu nevyskytují, podobnosti pak budou vypadat takto:

- 1. a 3. 0,2156
- 2. a 3. 0,1862
- 1. a 4. 0,1157

Podobnost tematicky souvisejících dokumentů je vyšší než podobnost tematicky nesouvisejících dokumentů a rozdíl těchto podobností je vyšší než u četnosti slov.

5.1.2 Latentní sémantická analýza

Pokud vezmeme matici četností slov, kterou vložíme jako vstup do latentní sémantické analýzy a určíme počet témat na čtyři, dostaneme tyto vektory dokumentů:

1. 1,60 0,36 0,92 1,20
2. 1,82 0,66 -1,49 0,18
3. 1,38 1,18 0,73 -1,09
4. 1,89 -1,80 0,12 -0,40

Řádky odpovídají dokumentům a sloupce jednotlivým tématům. Pokud vektory dokumentů porovnáme pomocí kosinové podobnosti, získáme tyto podobnosti:

- 1. a 3. 0,4000
- 2. a 3. 0,3651
- 1. a 4. 0,3380

Dosáhli jsme skoro stejných výsledků, jako v prvním případě modelu vektorového prostoru, ale oproti porovnávání vektorů dlouhých 16 dimenzí jsme museli porovnávat jen čtyři.

Pokud počet témat snížíme jen na dvě, dostaneme tuto matici:

1. 1,60 0,36
2. 1,82 0,66
3. 1,38 1,18
4. 1,89 -1,80

Porovnáním těchto vektorů získáme hodnoty:

- 1. a 3. 0,8843
- 2. a 3. 0,9361
- 1. a 4. 0,5535

S menším počtem témat se podobnosti razantně změnili. Podobnost druhého a třetího dokumentu je vyšší než v případě prvního a třetího a podobnost prvního a čtvrtého dokumentu je o poznání nižší než první dvě podobnosti.

Obecný postup pro výběr počtu témat není. „Vždy je počet témat ovlivněn minimálně dvěma faktory – použitými daty a konkrétní aplikací.“ (Materna (7) 2012)

Pokud počet témat vrátíme na čtyři a místo četnosti slov použijeme matici TF-IDF (druhý případ s přičtenou jedničkou) dostaneme tyto vektory dokumentů:

1. 0,41 -0,44 -0,67 0,79
2. 0,58 -0,99 0,77 0,09
3. 0,34 -0,69 -0,65 -0,71
4. 1,49 0,67 0,03 -0,09

Podobnosti pak jsou následující:

- 1. a 3. 0,2156
- 2. a 3. 0,1816
- 1. a 4. 0,1157

Tedy opět téměř stejné výsledky jako v případě vektorového prostoru, jen s menším počtem dimenzí.

Pokud snížíme počet dimenzí na dvě, dostaneme vektory dokumentů:

1. 0,41 -0,44
2. 0,58 -0,99
3. 0,34 -0,69
4. 1,49 0,67

V tomto případě budou podobnosti dokumentů:

- 1. a 3. 0,9589
- 2. a 3. 0,9971
- 1. a 4. 0,3195

Tedy tematicky související dokumenty jsou skoro stejné (jejich podobnosti jsou téměř jedna) a tematicky nesouvisející dokumenty jsou výrazně méně podobné.

Pokud použijeme výpočet TF-IDF bez přičtené jedničky, tedy slova obsažená ve všech dokumentech budou mít nulovou hodnotu, a počet dimenzí nastavíme opět na dvě, získáme tyto vektory:

1. 0,02 -0,12
2. 0,03 -1,05
3. 0,01 -0,33
4. 1,35 0,03

S podobnostmi sledovaných dvojic dokumentů:

- 1. a 3. 0,9855
- 2. a 3. 0,9999
- 1. a 4. 0,1641

Kde jsou tematicky podobné dokumenty ještě více podobné než v předchozím případě a tematicky nepodobné dokumenty mají ještě nižší podobnost.

5.1.3 Pravděpodobnostní latentní sémantická analýza

Pro první vstup do pravděpodobnostní latentní sémantické analýzy zvolíme opět četnost slov s počtem dimenzí 4, přičemž získáme tyto vektory:

1. 0,00 0,98 0,01 0,00
2. 0,98 0,01 0,01 0,00
3. 0,00 0,01 0,99 0,00
4. 0,00 0,00 0,00 0,99

Pravděpodobnostní latentní sémantická analýza nastavila každému dokumentu, že nejvíce spadá do jednoho tématu, do kterého jiný dokument skoro nespadá. Podobnosti díky tomu budou vypadat takto:

- 1. a 3. 0,0193
- 2. a 3. 0,0132
- 1. a 4. 0,0102

Podobnosti všech dokumentů jsou velmi nízké a rozdíl mezi podobnostmi tematicky souvisejících a nesouvisejících dokumentů jsou skoro stejné. Tyto výsledky tedy nemají skoro žádnou vypovídací hodnotu.

Pokud počet dimenzí snížíme na dvě, dostaneme vektory:

1. 0,99 0,01
2. 0,99 0,01
3. 0,16 0,84
4. 0,01 0,99

A podobnosti dvojic dokumentů z kolekce budou:

- 1. a 3. 0,1952
- 2. a 3. 0,1887
- 1. a 4. 0,0242

V tomto případě byly první dva dokumenty zařazeny skoro jen do prvního tématu, čtvrtý dokument do druhého témat a třetí dokument z větší části do druhého tématu. Podobnosti jsou stále malé, ale v tomto případě mají dostatečnou vypovídací hodnotu. Ale pokud spustíme pravděpodobnostní latentní sémantickou analýzu na stejných vstupních datech se stejným počtem témat, můžeme dostat jiné, méně vypovídající hodnoty. Proto doporučuje Farahat a Chen (2006) při použití této metody několik průběhů a vybrat ten s největší pravděpodobností.

Dalším důvodem špatných vypovídacích hodnot u pravděpodobností latentní sémantické analýzy je v tomto případě malá vstupní kolekce dat, kde inicializační náhodné hodnoty silně ovlivní výsledek, a proto je lepší tuto metodu používat na větších kolekcích dat.

Se vstupními hodnotami této kolekce vypočítanými pomocí TF-IDF si pravděpodobnostní latentní sémantická analýza poradí ještě hůře. Hofmann (1999) doporučuje při použití této metody používat vstupy v podobě četnosti slov.

5.1.4 Latentní Dirichletova alokace

Při použití latentní Dirichletovy alokace je vstupní kolekce jiná než v předchozích případech a nelze přepínat mezi četností slov a TF-IDF. Výsledné vektory s počtem témat čtyři budou (každé spuštění metody může mít mírně rozdílné výsledky i se stejnými vstupními parametry):

1. 2,00 0,00 0,00 3,00
2. 0,00 6,00 0,00 0,00
3. 0,00 2,00 0,00 3,00
4. 4,00 0,00 3,00 0,00

Podobnosti v tomto případě budou tyto:

- 1. a 3. 0,6923
- 2. a 3. 0,5547
- 1. a 4. 0,4438

Podobnosti jsou vyšší než v předchozím případě u pravděpodobností latentní sémantické analýzy, ale rozdíl mezi podobností tematicky souvisejících a nesouvisejících dokumentů není takový, aby měl dobrou vypovídací hodnotu. Ve vektorech dokumentů lze také vidět, že druhý dokument má nenulovou podobnost pouze s třetím dokumentem.

Pokud snížíme počet témat na dvě, dostaneme tyto vektory:

1. 1,00 4,00
2. 0,00 6,00
3. 0,00 5,00
4. 7,00 0,00

Podobnosti dvojic dokumentů se změní na tyto:

- 1. a 3. 0,9701
- 2. a 3. 1,0000
- 1. a 4. 0,2525

Druhý a třetí dokument je tedy podle této metody tematicky stejný, první a třetí téměř stejný a první a čtvrtý má oproti ostatním podobnosti velmi nižší podobnost.

Latentní Dirichletova alokace opět dosahuje lepších výsledků při použití větší vstupní kolekce dokumentů.

5.2 Větší kolekce dokumentů

Pro druhou ukázku je určená kolekce dvanácti dokumentů, jejichž texty jsou obsahem příslušných článků na české wikipedii (cs.wikipedia.org), které byly lemmatizovány. Každý dokument obsahuje několik stovek slov, a proto nemusí být podobnosti dobře patrné. Dokumenty v kolekci:

1. Březinka.csv – článek pojednávající o srubu těžkého opevnění republiky Československé
2. ELO.csv – Electric Light Orchestra byla britská hudební rocková skupina
3. HK.csv – článek o Hradci Králové
4. KrálovehradeckýKraj.csv – článek o historii, geografii a dopravě v kraji
5. UHK.csv – historie a popis Univerzity Hradec Králové
6. fotbal.csv – popis a pravidla tohoto sportu
7. kunětickáHora.csv – o hradu u Pardubic a jeho historii
8. lkvz26.csv – lehký kulomet vzor 26 určený mimo jiné i do opevnění
9. obrana.csv – krátký článek popisující obranu z vojenského hlediska
10. opevneni.csv – popis a historie Československého opevnění z let 1935-1938
11. rock.csv – popis a historie hudebního stylu
12. tvrze.csv – článek pojednávající o pilíři Československého opevnění, dělostřeleckých tvrzích

Podle tohoto seznamu s krátkým popisem článků by spolu měli souviset některé dokumenty ve velké míře, některé v menší a několik dokumentů vůbec. Nejvíce by spolu měli souviset články o opevnění (10), tvrzích (12) a Březince (1), se kterými by měli o něco méně souviset také články o lehkém kulometu vz. 26 (8) a obraně (9) a v menší míře i text o hradu Kunětická hora (7). Větší podobnost by také měli mít články o Hradci Králové (5) a Královehradeckém kraji (4) a méně souviset s nimi by měl i článek o Univerzitě Hradec Králové (5). A nakonec texty z článků o skupině ELO (2) a o rocku (11), který skupina hraje, by měli být také podobné.

Podobnosti byly vypočteny jako v předchozím případě kosinovou metrikou.

5.2.1 Model vektorového prostoru

Program v této kolekci zařadí mezi slova i letopočty a jiné číslovky, spoustu názvů a zkratek a také cizí slova často uváděná na začátku článků. V případě četnosti slov budou mít vybrané dokumenty tyto podobnosti:

- Březinka a opevnění: 0,6301
- Březinka a tvrze: 0,6096
- ELO a rock: 0,4840
- Opevnění a obrana: 0,5136
- Březinka a ELO: 0,4505
- Kunětická hora a fotbal: 0,3633

První dvě dvojice vybraných dokumentů by měli velmi podobné, což v tomto případě odpovídá s podobností 0,6 u obou dvojic. Další dvě dvojice by měli být také podobné, ale v menší míře, což také odpovídá s podobností 0,5 a 0,48. Poslední dvě dvojice by si tematicky neměli být podobné vůbec, podobnosti jsou v tomto případě menší než u ostatních dvojic, ale podobnost článků o srubu opevnění Březince a hudební kapele ELO je skoro stejná jako podobnost skupiny ELO a žánru rock, který skupina hraje.

U těchto větších dokumentů se začíná projevovat nevýhoda četnosti slov, vysoká váha často opakujících se slov, proto se většinou při sémantické analýze z textů vyloučí stop slova, nebo se použije metoda vážení slov, která tento problém eliminuje, jako TF-IDF. Po použití této metody, slova, vyskytující se ve všech dokumentech, budou mít hodnotu nulovou a podobnosti se zlepší:

- Březinka a opevnění: 0,0646
- Březinka a tvrze: 0,1398
- ELO a rock: 0,0630
- Opevnění a obrana: 0,0621
- Březinka a ELO: 0,0314
- Kunětická hora a fotbal: 0,0107

Většina podobnosti je sice o řád menší, ale vypovídací hodnota je mnohem větší. Nejvíce podobná dvojice článků je Březinka a tvrze Československého opevnění, poté jsou podobnosti dvojic Březinka a opevnění, ELO a rock a opevnění a obrana, které jsou skoro stejně velké. Oproti nim poloviční podobnost má dvojice dokumentů pojednávající o Březice a skupině ELO, kde můžeme soudit, že stále spolu sdílí stop slova, která nejsou obsažena ve všech dokumentech, a tedy nějakou hodnotu mají. Nejmenší podobnost má hrad Kunětická hora a fotbal, která je třetinová oproti podobosti Březinky a skupiny ELO a asi desetkrát menší než podobnost článků o Březince a tvrzích.

Pokud použijeme druhou verzi výpočtu TF-IDF s přičtenou jedničkou v logaritmu, aby nulovou hodnotu dostala jen slova se v dokumentech nevyskytující, dostaneme podobnosti následující:

- Březinka a opevnění: 0,2289
- Březinka a tvrze: 0,2751
- ELO a rock: 0,1512
- Opevnění a obrana: 0,1792
- Březinka a ELO: 0,1237
- Kunětická hora a fotbal: 0,0754

Podobnosti jsou vyšší než v předchozím případě, ale velký počet výskytu stop slov opět velmi ovlivnil výsledky. Nejvíce podobné dvojice jsou stejné, jako v minulém případě, ale podobnost článků o Březince a skupině ELO není o moc nižší než podobnost článků o skupině ELO a rocku.

5.2.2 Latentní sémantická analýza

Při použití četnosti slov a nastavení počtu témat na dvanáct dostaneme podobnosti u sledovaných dvojic stejné jako v případě modelu vektorového prostoru s použitím četnosti slov. Ale pokud počet témat zkusíme snížit na pět, dostaneme:

- Březinka a opevnění: 0,8392
- Březinka a tvrze: 0,8206
- ELO a rock: 0,8421
- Opevnění a obrana: 0,9880

- Březinka a ELO: 0,7804
- Kunětická hora a fotbal: 0,6311

Podobnosti se razantně zvýšili, ale nejvyšší podobnost má dvojice opevnění a obrana, která v předchozích případech byla vždy o něco menší než první dvě podobnosti.

Pokud počet témat zvýšíme na sedm, podobnosti se změní na:

- Březinka a opevnění: 0,7644
- Březinka a tvrze: 0,6413
- ELO a rock: 0,6403
- Opevnění a obrana: 0,9879
- Březinka a ELO: 0,6349
- Kunětická hora a fotbal: 0,3473

V tomto případě se podobnost opevnění a obrany téměř nezměnila, ale ostatní ano. Podobnosti dvojic Březinka a tvrze, ELO a rock a Březinka a ELO jsou téměř stejné a podobnost dojice Kunětická hora a fotbal je oproti nim téměř poloviční.

Pokud využijeme TF-IDF bez přičtené jedničky v logaritmu, která nám v modelu vektorového prostoru zajistila nejlepší výsledky, a počet témat nastavíme opět na sedm, podobnosti budou:

- Březinka a opevnění: 0,3182
- Březinka a tvrze: 0,1877
- ELO a rock: 0,0627
- Opevnění a obrana: 0,9167
- Březinka a ELO: 0,0329
- Kunětická hora a fotbal: 0,0220

Dvojice s nejvyšší podobností opět zůstává opevnění a obrana, po které následuje třetinová podobnosti dvojice Březinka a opevnění. Dvojice Březinka a tvrze má podobnosti poloviční oproti dvojici Březinka a opevnění, které v předchozích případech vycházely podobně. Podobnost tematicky nesouvisejících článků Březinka a ELO je poloviční oproti článkům ELO a rock.

Pokud budeme zkoušet měnit počet témat, zjistíme, že podobnost dvojice obrana a opevnění bude jiné pouze, pokud počet témat zůstane dvanáct, jelikož dvanácté téma vypočítané latentní sémantickou analýzou má hodnotu u článku o obraně zápornou a u článku o opevnění kladnou. Tedy v tomto případě je optimální počet témat dvanáct a podobnosti jsou stejné, jako v modelu vektorového prostoru:

- Březinka a opevnění: 0,0646
- Březinka a tvrze: 0,1398
- ELO a rock: 0,0630
- Opevnění a obrana: 0,0621
- Březinka a ELO: 0,0314
- Kunětická hora a fotbal: 0,0107

Podobnosti jsou tedy stejné, jako ve vektorovém prostoru, ale velikost porovnávaných vektorů a výpočetní náročnost algoritmu je nižší než ve vektorovém prostoru. Caid, Dumais a Gallant (1995) uvádějí, že na vstupní matici obsahující kolem 2 000 dokumentů a 5 000 slov trval (v roce 1995) výpočet latentní sémantické analýzy kolem dvou minut a u matice o přibližně 60 000 dokumentech a 80 000 slovech se tento čas zvýšil asi na 18 hodin (opět v roce 1995) a oproti modelu vektorového prostoru si vede v některých případech až o 30% lépe nebo, jako v tomto případě, alespoň stejně dobře. K problému s určením počtem témat uvádějí příklad, kdy v kolekci s něco málo než 70 000 dokumenty určovali počet témat, který nakonec nastavily na 199.

Husbands, Simon a Ding (2005) uvádějí, že na větších kolekcích výkon latentní sémantické analýzy již není tak dobrý. Na kolekci o přibližně 528 000 dokumentech a 115 000 slov byl již velký problém určit optimální počet témat, a tak zkoušeli různé hodnoty až do počtu témat 1 000, ale výsledky ve všech případech si byly velmi podobné. V případě velkých kolekcí je zvýraznění méně častých slov pomocí TF-IDF nedostačující, proto uvádějí normalizovanou latentní sémantickou analýzu.

5.2.3 Pravděpodobnostní latentní sémantická analýza

Pokud jako vstup do pravděpodobnostní latentní sémantické analýzy zvolíme četnost slov a počet témat dvanáct, podobnosti sledovaných dvojic dokumentů budou (každý průběh algoritmu může vykazovat různé výsledky):

- Březinka a opevnění: 0,0143
- Březinka a tvrze: 0,0148
- ELO a rock: 0,0114
- Opevnění a obrana: 0,9998
- Březinka a ELO: 0,0026
- Kunětická hora a fotbal: 0,0015

Nejvyšší podobnost, je jako u latentní sémantické analýzy, kde jsme snížili počet témat, u dvojice opevnění a obrana a to téměř jedna. Ostatní podobnosti jsou velmi nižší, ale je vidět, že dvojice Březinka a opevnění a Březinka a tvrze spolu souvisí více, než ostatní dvojice. Dvojice ELO a rock mají o třetinu menší podobnost, než předchozí dvojice, tedy spolu tematicky souvisí méně, ale stále souvisí, oproti tomu dvojice Březinka a ELO a hrad Kunětická hora a fotbal mají podobnosti v řádech tisícin, tedy spolu souvisí o poznání méně, než ostatní dvojice.

Pokud spustíme pravděpodobnostní latentní sémantickou analýzu znovu na četnosti slov a se stejným počtem parametrů, můžeme dostat jiné hodnoty a z nich tyto podobnosti:

- Březinka a opevnění: 0,0163
- Březinka a tvrze: 0,0166
- ELO a rock: 0,0143
- Opevnění a obrana: 0,0198
- Březinka a ELO: 0,0099
- Kunětická hora a fotbal: 0,0105

V tomto případě je stále podobnost opevnění a obrany nejvyšší, ale již v uvěřitelné míře oproti ostatním podobnostem. Rozdíly mezi jednotlivými podobnostmi jsou menší, než v předchozím případě, ale výsledky jsou podobné, kromě dvojice opevnění a obrany. Farahat a Chen (2006) ve své práci při používání pravděpodobnostní latentní sémantické analýzy nejdřív pomocí několika desítek spuštění algoritmu zjistí optimální hodnoty logaritmické hodnoty celku a poté používají jen ty instance, které mezi tyto hodnoty zapadají a navrhují zlepšení výsledků s lepšími metody inicializace, než náhodnými hodnotami.

Pokud počet témat snížíme na osm a necháme četnost slov jako vstupní hodnoty, podobnosti budou:

- Březinka a opevnění: 0,0362
- Březinka a tvrze: 0,0328
- ELO a rock: 0,0336
- Opevnění a obrana: 0,0177
- Březinka a ELO: 0,0075
- Kunětická hora a fotbal: 0,0041

V tomto případě je největší podobnost dvojice Březinka a opevnění, ale druhá největší je ELO a rock, která v předchozích případech bývala menší, než první dvě dvojice, podobná s dvojicí opevnění a obrana. Ostatní podobnosti jsou na tom podobně, jako v předchozích případech, nejmenší je u dvojice článků Kunětická hora a fotbal a o něco větší je Březinka a ELO, která je více jak dvakrát menší, než opevnění a obrana, která je poloviční oproti tematicky nejpodobnějším článkům.

Hofmann (1999) porovnává pravděpodobnostní latentní sémantickou analýzu a latentní sémantickou analýzu s tvrzením, že prvně zmíněná překonává druhou ve větší míře. Ve všech případech použití obou algoritmů dosahuje značně lepšího výkonu a uvádí, že pravděpodobnostní latentní sémantická analýza si vede dobře i v případech, kdy latentní sémantická analýza selhává, ale uvádí, že je možné použít kombinaci obou algoritmů, která si vede lépe, než algoritmy samotné.

Farahat a Chen (2006) uvádějí, že pro zlepšení výsledků, lze použít zprůměrovaná data z několika instancí pravděpodobnostní latentní sémantické analýzy a navíc přidává jednu instanci inicializovanou pomocí dat z rozkladu na singulární hodnoty, takto získané výsledky si vedou dobře i na malých kolekcích dat, kde algoritmus této metody, jako i v našem příkladu, selhává. Dále srovnává zprůměrované modely obsahující několik instancí pravděpodobnostní latentní sémantické analýzy i latentní sémantické analýzy.

Všechny citované zdroje doporučují a sami používají tuto metodu se vstupními daty v podobě četnosti slov, maximálně s vyjmutými stop slovy, ale i s těmi si algoritmus poradí, jak můžeme vidět i v tomto příkladu. Při použití TF-IDF má algoritmus větší potíže, protože ve vstupní matici je více hodnot nulových

a tak se jednotlivé výsledky můžou velmi lišit a jejich většina nemusí mít žádnou vypovídací hodnotu.

5.2.4 Latentní Dirichletova alokace

Pokud použijeme na tuto kolekci dokumentů metodu latentní Dirichletovi alokace s počtem témat dvanáct, dostaneme podobnosti sledovaných dvojic článků ve třech instancích algoritmu:

- Březinka a opevnění: 0,2741 0,2272 0,1985
- Březinka a tvrze: 0,5268 0,3548 0,3394
- ELO a rock: 0,3461 0,3501 0,3147
- Opevnění a obrana: 0,9521 0,6580 0,5859
- Březinka a ELO: 0,0990 0,1395 0,1136
- Kunětická hora a fotbal: 0,2689 0,2757 0,3394

Z těchto podobností vidíme, že opět nejpodobnější je označena dvojice opevnění a obrana, v prvním případě s podobností 0,95. Kromě prvního případu, jsou první tři dvojice na tom podobnostně stejně. Podobnost dvojice článků Březinka a ELO je nejnižší, a to přibližně třetinová oproti dvojici ELO a rock, ale podobnost Kunětické hory a fotbalu je vyšší, ve třetím případě i stejně velká, jako podobnost dvojice Březinka a tvrze.

Pokud počet témat snížíme na osm, podobnosti se změní na:

- Březinka a opevnění: 0,4020 0,2982 0,4542
- Březinka a tvrze: 0,4207 0,4162 0,8938
- ELO a rock: 0,9926 0,3302 0,3087
- Opevnění a obrana: 0,5389 0,4887 0,5223
- Březinka a ELO: 0,1951 0,1047 0,1883
- Kunětická hora a fotbal: 0,3212 0,0898 0,3557

V prvním případě je podobnost dvojice ELO a rock téměř jedna, ve třetím zase podobnost dvojice Březinka a tvrze je skoro 0,9. Můžeme tedy vidět, že i v latentní Dirichletově alokaci můžou některé výsledky poskočit od optimálních hodnot.

Blei, Ng a Jordan (2003) jako ukázku používají algoritmus latentní Dirichletovi alokace na kolekci dat o 16 000 dokumentech s odstraněnými stop slovy s určeným počtem témat na hodnotě 100. Kromě toho, že ukazují, jak je tato metoda výkonná, také poukazují na její omezení. Díky jinému formátu vstupních dat, než u předchozích metod, vzniká problém, kdy slova, která by měla spolu tematicky souviset, se mohou dostat do témat různých. Výhoda této metody oproti pravděpodobnostní latentní sémantické analýze je nižší výpočetní náročnost daná nižším počtem parametrů algoritmu. Ale největší výhoda je možnost porovnávat nové dokumenty, neobsažené v trénovací kolekci, což základní pravděpodobnostní latentní analýza neumožňuje.

6 Závěr

Základním způsobem převedení textu do podoby lépe čitelné počítačem je model vektorového prostoru, kde je text reprezentován maticí udávající každému slovu v každém dokumentu nějakou hodnotu, ať již binární, četnost slov nebo TF-IDF. Tato reprezentace u větších dokumentů obsahuje velké množství hodnot a často je nepřehledná a porovnávání vektorů v tomto prostoru výpočetně náročné, ale přesto svá využití najde. U menších kolekcí dokumentů, které spolu tematicky souvisejí a obsahují společné významová slova je tato reprezentace dostatečné i k jejich porovnávání či hledání v nich. Tato reprezentace si díky metodě TF-IDF také poradí se základním problémem a to se slovy, které se často vyskytují, ale nemají významovou hodnotu. Ale pro větší kolekce dokumentů je tento způsob reprezentace nedostatečný a výpočetně náročný.

S pomocí rozkladu matice na singulární hodnoty vznikla metoda latentní sémantické analýzy, která tohoto rozkladu využije a matici četnosti slov rozloží na matici, která vytvoří témata pro kolekci dokumentů a přiřadí jim hodnoty, aby je bylo možné porovnávat. Díky tomu lze snížit počet dimenzí vektorového prostoru na libovolný počet, ale určit tento počet správně je hned prvním problémem této metody, který se řeší empiricky podle dosažených výsledků. Při porovnávání dokumentů tedy porovnáváme daleko menší vektory a tím šetříme čas i výpočetní náročnost a výsledky s touto metodou dosažené jsou alespoň stejně dobré, jako v modelu vektorového prostoru nebo i lepší.

Abychom dosáhli ještě lepších výsledků, vznikla metoda pravděpodobnostní latentní sémantické analýzy, která pomocí statistického modelu a pravděpodobnostní vytvoří témata pro dokumenty a rozdělí je do nich. Jako v předchozí metodě, je počet témat téměř libovolný, a jeho určení je stejný problém. Ale oproti latentní sémantické analýze má tato metoda lepší základ a výsledky s ní dosažené jsou lepší. Ovšem při inicializaci tohoto algoritmu se nejčastěji používá náhodných dat, a ty mohou ovlivnit výsledky i do té míry, že budou naopak horší nebo nesmyslné. Nejvíce je to vidět u menších kolekcí dokumentů, se kterými má tato metoda obtíže. Řešení tohoto problému je více instancí algoritmu se stejnými parametry a použití pouze optimálních výsledků nebo jejich zprůměrování, nebo inicializace dat jiným způsobem.

Největší nevýhodou této metody je rostoucí počet parametrů a nemožnost porovnávání s novými dokumenty.

Řešením výše uvedených problémů je latentní Dirichletova alokace, která z pravděpodobnostní latentní sémantické analýzy vychází, ale má stejný počet parametrů, takže je výpočetně méně náročná a výsledky s ní dosažené jsou na lepší úrovni a většina výsledků je optimálních.

I když je latentní Dirichletova alokace na větších kolekcích dat nejlepší jak výkonnostně, tak ve výsledcích, všechny metody najdou své využití. Na menších datech je lepší používat latentní sémantickou analýzu nebo i model vektorového prostoru, který je v některých případech dostačující i s pouhou četností slov nebo binární reprezentací slov. Ale s rostoucí kolekcí dokumentů a jejich velikostí je zapotřebí používat lepší metody. Další možností je používat kombinace těchto metod a tím dosáhnout ještě lepších výsledků než při použití jednotlivých metod.

7 Zdroje

1. BAKER, Kirk. Singular Value Decomposition Tutorial. The Ohio State University 24 2005.
2. BLEI, David M., NG, Andrew Y., JORDAN, Michael I. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022
3. CAID, William R., DUMAIS, Susan T. a GALLANT, Stephen I. Learned Vector-Space Models for Document Retrieval. *Information Processing & Management*. 1995, 3(31), 419-429. 0306-4573(94)00056-5
4. FARAHAT, Ayman a CHEN, Francine. Improving Probabilistic Latent Semantic Analysis with Principal Component Analysis. 2006.
5. HOFMANN, Thomas. Probabilistic Latent Semantic Analysis. *Uncertainity in Artificial Intelligence*. 1999.
6. HOFMANN, Thomas. Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and Categorization. *Advances in Neural Information Processing Systems.* 2000, 12, 914-920.
7. HUSBANDS, Parry, SIMON, Horst a DING, Chris. Term norm distribution and its effects on Latent Semantic Indexing. *Information Processing and Management.* 2005, 41, 777-787.
8. KONTOSTATHSIS, April a POTTENGER, William M. A framework for understanding Latent Semantic Indexing (LSI) performance. *Information Processing and Management.* 2006, 42, 56-73.
9. MATERNA, Jiří. Sémantická analýza textů (1). Blog fulltextového týmu [online]. 2011 [cit. 12. 12. 2015]. Dostupné z: *http://fulltext.sblog.cz/2011/08/30/semanticka-analyza-textu-1/*
10. MATERNA, Jiří. Sémantická analýza textů (2). Blog fulltextového týmu [online]. 2011 [cit. 12. 12. 2015]. Dostupné z: *http://vyhledavani.sblog.cz/2011/09/12/semanticka-analyza-textu-2/*

12. MATERNA, Jiří. Sémantická analýza textů (3). Blog fulltextového týmu [online]. 2011 [cit. 12. 12. 2015]. Dostupné z: *http://vyhledavani.sblog.cz/2011/09/22/semanticka-analyza-textu-3/*

13. MATERNA, Jiří. Sémantická analýza textů (4). Blog fulltextového týmu [online]. 2011 [cit. 12. 12. 2015]. Dostupné z: *http://vyhledavani.sblog.cz/2011/10/17/semanticka-analyza-textu-4/*

14. MATERNA, Jiří. Sémantická analýza textů (5). Blog fulltextového týmu [online]. 2011 [cit. 12. 12. 2015]. Dostupné z: *http://vyhledavani.sblog.cz/2011/12/04/semanticka-analyza-textu-5/*

15. MATERNA, Jiří. Sémantická analýza textů (6). Blog fulltextového týmu [online]. 2011 [cit. 12. 12. 2015]. Dostupné z: *http://vyhledavani.sblog.cz/2011/12/22/semanticka-analyza-textu-6/*

16. MATERNA, Jiří. Sémantická analýza textů (7). Blog fulltextového týmu [online]. 2012 [cit. 12. 12. 2015]. Dostupné z: *http://vyhledavani.sblog.cz/2012/02/14/semanticka-analyza-textu-7/*

17. ONEATA, Dan. Probabilistic Latent Semantic Analysis. The University of Edinburgh School of Informatic. Citováno 2016.

18. TURNEY, Peter D. a PANTEL, Patrick. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*. 2010, 37, 141-188. DOI: 10.1613/jair.2934.

Přílohy

Příloha 1: Popis algoritmů použitých v programu

Příloha 2: Návod k programu

Příloha 3: Struktura přiloženého CD s programem

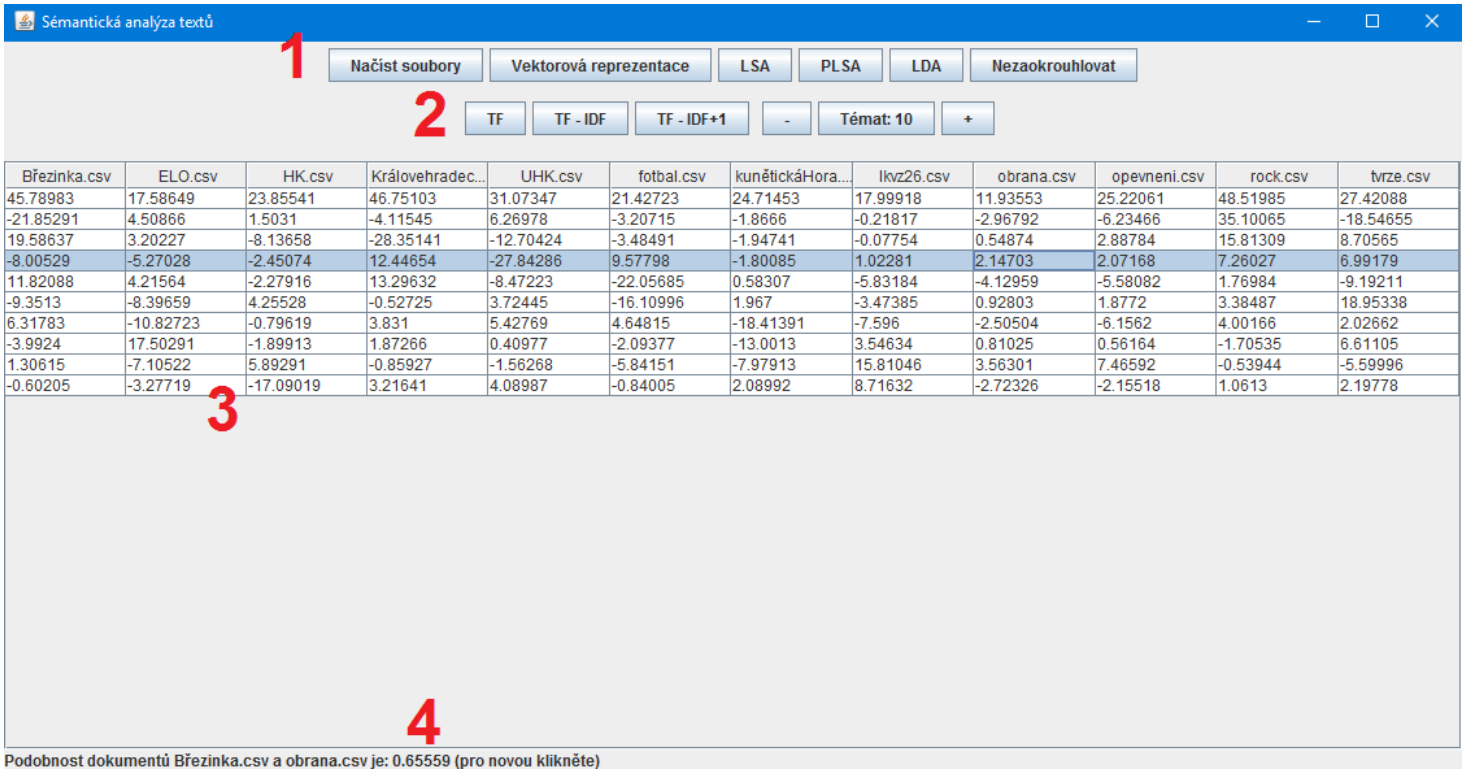
Příloha 4: CD s programem

Příloha 5: Zadání práce

Popis zdrojových kódů v programu

- aplikace
  - AppSemantickaAnalyza.java – Hlavní třída zodpovědná za spuštění programu
- gui
  - SemantickaAnalyzaGui.java – Třída zodpovědná za vytvoření uživatelského rozhraní a prezentování výsledků uživateli
- model
  - Dokument.java – Třida pro jeden dokument (větu nebo hledaný řetězec), obsahuje všechna jedinečná slova v dokumentu a počet jejich výskytů
  - Lda.java – Třída provádějící metodu latentní Dirichletovi aleokace pomocí Gibbsova vzorkování
  - Matice.java – Třída pro reprezentaci dvourozměrného pole čísel vycházející ze třídy Matrix knihovny Jama
  - PLSA.java – Třída provádějící metodu pravděpodobnostní latentní sémantické analýzy pomocí temperovaného algoritmu předpokladu a maximalizace
  - SemantickaSVD.java – Třída pro metodu latentní sémantické analýzy pomocí vstupu z rozkladu na singulární hodnoty z knihovny Jama
  - Soubor.java – Třída pro reprezentaci kolekce dokumentů, obsahuje výpočety četnosti slov a TF-IDF
  - Vektor.java – Třida pro reprezentaci jednorozměrného pole čísel a pro výpočty s ním spojené (kosinova metrika)
- sluzby
  - ImportSouboru.java – Třída zodpovědná za čtení souborů ve formátu csv a vytvoření vstupních dat pro reprezentaci dokumentů
  - Nacitac.java – Třída obsahující vlastní čtení dokumentu
  - SemantickaAnalyzaObsluha.java – metoda obsluhující pokyny z uživatelského rozhraní odpovědná za dodávání dat
  - TableModelSemAnalyza.java – třída tableModelu pro tabulku v uživatelském rozhraní, obsahující data z výstupních matic
- knihovny
  - Jama 1.0.3 – Maticová knihovna pro jazyk Java, v programu je využit výpočet rozkladu na singulární hodnoty a třída Matrix, která je rozšířena na třídu Matice, dostupná na: *http://math.nist.gov/javanumerics/jama/*

Návod k programu



1. Po spuštění programu bude viditelné pouze tlačítko *Načíst soubory* a *Nezaokrouhlovat*.
    a. Tlačítko *Načíst soubory* otevře nové okno s průzkumníkem souborů, ve kterém očekává vybrání vstupních souborů ve formátu CSV, souborů je nutné vybrat více najednou. Pokud načtení z nějakého důvodu selže, soubory mohou být načteny znovu a zobrazí se upozornění místo tabulky 3.
    b. Po načtení souborů se zobrazí *Vektorová reprezentace* kolekce dokumentů v tabulce 3 a zobrazí příslušná tlačítka v řade 2.
    c. *LSA* provede výpočet latentní sémantické analýzy a zobrazí výsledky v tabulce 3 a zobrazí příslušná tlačítka v řadě 2.
    d. *PLSA* vypočte pravděpodobnostní latentní sémantickou analýzu a zobrazí výsledky v tabulce 3 a zobrazí příslušná tlačítka v řadě 2.
    e. *LDA* provede výpočet latentní Dirichletovy alokace a zobrazí výsledky v tabulce 3 a zobrazí příslušná tlačítka.

   f. *Nezaokrouhlovat* / *Zaokrouhlovat* přepíná zaokrouhlování výsledků na pět desetinných míst.
2. Tyto tlačítka budou viditelná pouze po načtení souborů a v závislosti na vybrané metodě.
   a. *TF* přepne výpočet na četnost slov.
   b. Podobně jako *TF*, ale matice bude vypočítána pomocí TF-IDF, tedy četnosti slov dělené inverzní četnostní dokumentů.
   c. Viz předchozí případ, při výpočtu TF-IDF přičte k logaritmu inverzní četnosti dokumentů jedničku.
   d. Tlačítka + a - zvýší nebo sníží počet témat pro metody *LSA* apod.
   e. Tlačítko *Témat: X* provede znovu výpočet zvolené metody s aktuálně nastaveným počtem témat.
3. Tabulka zobrazující matici hodnot četnosti slov nebo výsledků některé z metod sémantické analýzy textů. Pokud nejsou načteny soubory, nebo probíhá výpočet, tabulka je zobrazena prázdná nebo pouze s hlavičkou a je zde umístěn informační text. Výsledky jsou zaokrouhleny podle stavu tlačítka.
4. *Podobnost dokumentů*. Tato položka se zobrazí po načtení dokumentů a pro její vypočtení je potřeba dvakrát kliknout levým tlačítkem myši do tabulky 3 na dva sloupce zobrazující dokumenty. Podobnost je vypočítána pomocí kosinové podobnosti.

Program není nijak výpočetně optimalizován a algoritmy nejsou nijak zlepšeny pro větší kolekce dokumentů, takže s větší kolekcí dokumentů nebo s velkým počtem slov může výpočet trvat delší dobu a dá se zastavit pouze vypnutím programu.

Vlastní vstupní dokumenty musejí být ve formátu CSV s e středníkem jako oddělovačem a s daty ve druhém sloupci. Data nemusejí býti lemmatizovaná, ani česky psaná, ale výsledky jiných dat nemusejí míti vypovídací hodnotu.

Struktura přiloženého CD s programem

- Základní adresář
  - Návod k programu ve formátu PDF
  - Spustitelný soubor programu ve formátu jar
  - Složka *res*
    - Složka *menší kolekce dat*
      - Obsahující čtyři CSV soubory s ukázkovými daty použitými v první části praktické části bakalářské práce
    - Složka *větší kolekce dat*
      - Obsahující dvanáct CSV souborů s ukázkovými daty použitými v druhé části bakalářské práce obsahující texty z wikipedie
  - Složka *zdrojové kódy*
    - Zdrojové kódy k programu ve složkách podle balíčků ve formátu java
  - Složka *lib*
    - Knihovna Jama využívaná programem

Zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2015/2016

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Fries Matěj	Karla Lánského 839, Jaroměř	I1301211

TÉMA ČESKY:

Sémantická analýza textů

TÉMA ANGLICKY:

Semantic texts analysis

VEDOUCÍ PRÁCE:

Mgr. Jiří Haviger, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl práce:
Popsat základní způsoby sémantické analýzy textů. Vytvoření aplikace na zpracování lematizovaných textů založené na popsaných algoritmech a datových reprezentacích.

Osnova:
1. Úvod
2. Vektorová reprezentace textu
3. Dimenzionalita
4. Latentní sémantická analýza
5. Ukázka na datech
6. Závěr

SEZNAM DOPORUČENÉ LITERATURY:

Jiří Materna, Sémantická analýza textů
http://fulltext.sblog.cz/2011/08/30/semanticka-analyza-textu-1/

Podpis studenta: ..... **Datum:** .....

Podpis vedoucího práce: ..... **Datum:** .....