Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Dolovanie súvisiacich patentov k vedeckým článkom

Bakalárska práca

2016
Vladimír Macko

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Dolovanie súvisiacich patentov k vedeckým článkom

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	2508 Informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	Mgr. Vladimír Boža

Bratislava, 2016
Vladimír Macko

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Vladimír Macko
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Dolovanie súvisiacich patentov k vedeckým článkom
Mining of related patents for scientific papers

Cieľ: Cieľom práce je vytvoriť systém, ktorý bude pomocou metód strojového učenia vyhľadávať patenty súvisiace so zadaným vedeckým článkom. Práca by mala obsahovať porovnanie viacerých metód a vhodnú evaluáciu kvality použitých metód.

Vedúci: Mgr. Vladimír Boža
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 16.10.2015

Dátum schválenia: 27.10.2015 doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie: Rád by som poďakoval môjmu školiteľovi Vladimírovi Božovi a zvyšku tímu projektu SciCurve, teda Martinovi Majerníkovi. Poskytli mi nielen problém bakalárskej práce, ktorého riešenie možno niekomu naozaj pomôže, ale aj všetky dostupné prostriedky potrebné na jeho riešenie. Veľká vďaka patrí aj pani učiteľke Darine Sýkorovej, lebo bez jej gramatickej korektúry by bola táto práca nepublikovateľná.

Taktiež ďakujem svojim rodičom a priateľke Barborke. Za všetko.

Abstrakt

Hľadanie relevantných patentov k minulému alebo prebiehajúcemu výskumu v biomedicíne je zložité, ale dôležité pre výskum a vývoj biotechnického priemyslu. V tejto práci sme otestovali potenciál metód LSA a TFIDF, ktoré by mohli tento proces výrazne uľahčiť. Na určenie úspešnosti sme použili nami vytvorený súbor ručne anotovaných dát a nami získané dáta o citovaní článkov v patentoch. Použitím len verejne dostupných dát dokážeme automaticky nájsť relevantný patent k biomedicínskemu článku s presnosťou 58%. Naše výsledky ukazujú, že testované metódy dokážu pomôcť v odhaľovaní vzťahu medzi globálnou publikačnou činnosťou a podávaním patentov.

Kľúčové slová: patent, vyhľadávanie, článok, relevantný, LSA, TFIDF, biomedicína

Abstract

Searching for a relevant patents related to past and ongoing research in biomedicine is complicated, but important for R& D in Biotech industry. In this thesis, we tested TFIDF and LSA methods for their potential to significantly ease this process. Hand-annotated data-set and extracted data on specific patent-related research publications were used to determine precision of tested methods. Using only publicly available data, we were able to automatically determine relevant patents for biomedical research publication with precision of 58%. Our results show, that tested methods have potential to facilitate relationship between global publication research and patent submission activity.

Keywords: patent, search, retrieval, article, paper, relevant, LSA, TFIDF, biomedicine, biotech

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Táto práca je súčasťou projektu SciCurve[9], dátovej platformy vytvorenej na generovanie, vizualizovanie a interpretovanie metadát pre potreby výskumu a vývoja v biomedicíne.

Biomedicínsky priemysel v súčasnosti patrí k najprosperujúcejším na svete[4]. Preto je ľubovoľné zlepšenie alebo optimalizácia procesov v ňom chceným a hodnotným prínosom. Jedným z hlavných problémov v tomto odvetví je časová a aj finančná náročnosť vývoja novej technológie. Napríklad pri vývoji liečiv je potrebné overenie ich dlhodobých účinkov, ktorému treba venovať rozsiahle štúdie.

Farmaceutickým firmám potom o to viac záleží na tom, aby mal ich výskum na konci aj naozajstný efekt, teda že dané liečivo bude fungovať a nikto iný ho nevyvinie skôr. Preto vynakladajú veľké finančné, aj ľudské prostriedky na skúmanie a sledovanie stavu vedy v ich oblasti záujmu. To zahŕňa sledovanie nových relevantných trendov a objavov tak v teoretickej vede, ako aj v praktických aplikáciach. Problémom je ale vágna definícia slova „relevantné “. Ani samotný vedec totižto často nevie, ako by popísal relevantný článok či patent. Pozná názov svojej paradigmy, ktorý môže použiť ako kľúčové slovo a nájsť všetky publikácie, ktoré ho obsahujú. Nástroje, ktoré mu to umožnia, už existujú a niektoré z nich sú aj voľne dostupné. Algoritmy použité v týchto nástrojoch sú väčšinou príliš strojové a nezohľadňujú do dostatočnej miery rôzne aspekty jazyka, ako sú synonymá, príslušnosť rôznych slov do jednej paradigmy, poprípade konkretizácie všeobecných pojmov. Navyše si vedec pri vyhľadávaní ďalších publikácií aj tak musí sám zvoliť nové kľúčové slová, pre ktoré proces hľadania zopakuje.

Okrem názvu svojej paradigmy a niekoľkých kľúčových slov ale vedec ešte disponuje niekoľkými článkami, ktoré pokladá za relevantné. Bolo by teda ideálne mať nástroj, ktorý by dokázal pri hľadaní nových relevantných publikácií použiť, aj informácie o už získaných článkoch. Pre začiatok by stačilo, keby sme vedcovi po príchode na stránku článku vedeli ponúknuť aj zoznam publikácií, ktoré s týmto článkom súvisia.

Hľadanie článkov k článkom je trochu jednoduchšie. Hľadaný dokument je totiž z rovnakej distribúcie ako ten, podľa ktorého chceme hľadať. Túto funkcionalitu poskytuje aj stránka projektu SciCurve.

Problém hľadania patentov k článkom je ale zložitejší a nepreskúmanejší. Presne

tomuto problému sa budeme v práci venovať. Naším cieľom je navrhnúť systém, ktorý používateľovi po príchode na stránku článku na portáli SciCurve v reálnom čase odporučí zoznam patentov súvisiacich s týmto článkom. Samozrejmosťou je tento systém aj otestovať a zistiť jeho úspešnosť.

Kapitola 1

Problematika

V tejto kapitole bližšie popíšeme problém, ktorý sme riešili, a požiadavky, ktoré musí naše riešenie spĺňať. Tiež sa oboznámime s dátami, ktoré pri riešení používame. Popíšeme ich vlastnosti, obmedzenia a možnosti.

1.1 Problém

Cieľom tejto práce je vytvoriť systém, ktorý dokáže vyhľadať relevantné patenty k zadanému vedeckému článku. Boli by sme radi, aby bol náš systém aj naozaj prakticky použiteľný a bolo možné ho implementovať na stránke SciCurve. Keď teda užívateľ príde na stránku nejakého článku, náš systém by mu mal rovno odporučiť niekoľko patentov, ktoré by ho mohli zaujímať. Preto požadujeme, aby vyhľadávanie patentov prebiehalo v reálnom čase a nemalo prílišné výpočtové nároky.

Je nutné si uvedomiť, ako dôležitá a obmedzujúca je podmienka odpovede v reálnom čase. Z dôvodu veľkého množstva článkov v našej databáze a veľkého množstva patentov, v ktorých chceme vyhľadávať, potrebujeme algoritmus, ktorý bude buď ostro lepší ako lineárny od počtu publikácií, alebo lineárny s veľmi nízkou konštantou. V skutočnosti nebude rozhodovať teoretická časová zložitosť, ale reálny čas trvania výpočtu programu v prevádzke.

Okrem toho požadujeme, aby sa náš systém dokázal učiť bez učiteľa. To znamená, že mu neukážeme, čo znamená relevantný dokument, ale budeme dúfať, že sa to naučí sám. Na učenie s učiteľom totižto nemáme dostatok dát a to málo, ktoré máme alebo sme ochotní vyrobiť, radšej použijeme na testovanie. Taktiež by sme sa chceli vyhnúť akémukoľvek nezautomatizovateľnému spracovaniu dát, ktoré je časovo náročné a zle škálovateľné.

Čo sa týka dát, požadujeme, aby náš algoritmus pracoval len s čo najvšeobecnejšou podobou patentov a článkov. Preto sa obmedzíme len na názvy a abstrakty týchto publikácií. Potom máme istotu, že v budúcnosti budeme môcť použiť aj ľubovoľné

ďalšie dostupné zdroje publikácií.

Pred tým, než sa naozaj pustíme do práce, musíme zodpovedať ešte jednu dôležitú otázku. Je to vôbec možné? Existuje naozaj spojitosť medzi patentami a článkami a ak áno, dá sa odhaliť len z ich abstraktov a nadpisov? V nasledovných kapitolách ukážeme, že odpovede na tieto otázky sú kladné.

1.2 Dáta

Tu sa bližšie oboznámime s charakteristikami dát, s ktorými budeme pracovať. Konkrétnymi číselnými charakteristikami dát a detailným spôsobom ich získavania sa budeme venovať až v časti 3.1.

1.2.1 Články

Databázu článkov zadarmo poskytol portál Pubmed[2] firme Scicurve, ktorá ju poskytla mne. Portál obsahuje len články týkajúce sa biomedicíny, čo je hlavným dôvodom našej špecializácie na tento obor. Druhým je lukratívnosť farmaceutického priemyslu, ktorý je s biomedicínou spojený.

Informácie, ktoré o každom článku máme, sú dobre štrukturované vo formáte xml a obsahujú: jednoznačný identifikátor (id článku v databáze), názov, abstrakt, dátum publikovania, zoznam autorov, kľúčové slová, zoznam inštitúcií, ktoré sa podieľali na publikovaní a zoznam citácií iných článkov. Rozhodli sme sa použiť len názvy a abstrakty.

Zoznam autorov je takmer nečlenený text s veľmi slabou, respektíve neexistujúcou špecifikáciou. Rôzne permutácie, skracovanie a vynechávanie mien ho potom robia veľmi ťažko štandardizovateľným a počítačovo spracovateľným. Situáciu zhoršujú napríklad aj čínske mená, vyznačujúce sa množstvom krátkych častí. Nie je totižto jednoznačne určiteľné, ktoré zápisy mien naozaj reprezentujú jedného autora. Napríklad pod menom *Yu W H* môžeme nájsť 106 článkov a pre toto meno nájdeme 19 jeho alternatívnych možných zápisov. Známym problémom sú tu aj zmeny priezvisiek po svadbe a dokonca sme sa stretli aj so zmenou mena po zmene pohlavia.

Zoznam inštitúcií je taktiež nepoužiteľný z podobných dôvodov ako zoznam autorov. V článkoch sa nachádza v textovom poli s názvom afiliácia. V ňom sú za sebou zreťazené identifikátory jednotlivých univerzít, firiem alebo ústavov. Tieto identifikátory pozostávajú z náhodných kombinácií názvov, adries, faxov, emailov a web-stránok v rôznych jazykoch. Príkladmi naozaj ťažko spracovateľných afiliácií sú

Tsentralnyi nauchno-issledovatelski institut stomatologii i cheliustno-litsevoi khirurgii Minzdrava Rossii, Moskva, Rossiia ,

alebo

Department of Chemistry, University of Pennsylvania, 231 S. 34 th St., Philadelphia, PA 19104, USA. Web: http://titanium.chem.upenn.edu/walsh/index.html ; Istanbul Technical University, Department of Chemistry, 34469 Maslak, Istanbul, Turkey. University of Houston, Houston, TX, USA, http://cbl.uh.edu ; Laboratoire MAS, Ecole Centrale Paris, France ; Equipe GALEN, INRIA Saclay – Ile–de–France.

Z pohľadu relevantnosti párov dokumentov nepredstavujú veľkú pridanú hodnotu. To, že dve publikácie sú z rovnakej inštitúcie, hovorí len veľmi málo o tom, ako veľmi sú si relevantné.

Kľúčové slová sme nepoužili, pretože nevieme kontrolovať ich kvalitu a niektoré články nemajú.

Máme dáta o tom, ako sa navzájom jednotlivé články citovali. Tieto však nie sú pre nás prínosné, nakoľko riešime vzťah článkov k patentom. Články môžu citovať aj patenty. Dáta o citovaní patentov ale nemáme, respektíve ich nemáme v spracovateľnej podobe. Ak by sme ich aj mali, veľmi by nám to nepomohlo. Články, ktoré citujú patenty, sú väčšinou len patentové prehľady, ktoré majú oproti ostatným článkom svoju špecifickú formu.

Zbežná prehliadka potvrdila hypotézu, že v abstraktoch sa nachádza dostatočná informácia o obsahu článku a že absencia celého znenia nie je problémom. Ukázala aj, ako veľmi špecifická slovná zásoba sa v tejto oblasti používa a ako veľmi odlišná je od bežnej reči. Vyskytujú sa v nej názvy rôznych chorôb, symptómov alebo iné odborné pojmy, ktoré nie sú v bežnej reči používané. Zmenené sú aj frekvencie a často aj hlavné významy niektorých bežných slov. Tak isto sa často vyskytujú chemické zlúčeniny zložené z čísiel a pomlčiek. Tie napríklad prinášajú nové problémy pri určovaní hraníc slov, ktorým sa v klasickej teórii vyhľadávania dokumentov nevenuje veľká pozornosť.

1.2.2 Patenty

Databázu patentov sme zadarmo stiahli z voľne prístupnej stránky[3]. Tu Google v spolupráci s USPTO zverejnil všetky patenty uznané USPTO (Úrad Patentov a Obchodných Známok Spojených Štátov Amerických), EPO (Európsky Patentový Úrad) alebo WIPO (Medzinárodná Organizácia Duševného Vlastníctva). Táto stránke už ale nie je aktualizovaná a momentálne je potrebné použiť stránku Reed [1].

Patenty sú dobre štruktúrované vo formáte xml a obsahujú: jednoznačný identifikátor (číslo patentu), názov, abstrakt, zoznam autorov, zoznam inštitúcií, dátum požiadania o patent, dátum uznania, zoznam citácií, kompletné znenie patentu, klasifikáciu

patentu podľa USPTO, obrázky a iné prílohy. Aj tu sme sa rozhodli použiť len názov s abstraktom a to hlavne kvôli konzistentnosti s článkami.

Kompletné znenie patentu sme nepoužili kvôli jeho vysokej technickosti a veľkému rozsahu. Jeho spracovávanie by potom len spôsobovalo časové a pamäťové problémy. Okrem toho sa jedná o takmer právnické dokumenty, takže obsahujú množstvo pre význam patentu nerelevantnej informácie. Z podobných dôvodov sme nepoužili ani prílohy a obrázky. Tak isto veríme, že abstrakt nesie veľkú časť informácie o podstate patentu.

Autori a inštitúcie spojené s patentom sú zaznamenané oveľa lepšie ako pri článkoch. Keďže ich pri článkoch nemáme v použiteľnej podobe, nemáme dôvod ich použiť ani pri patentoch.

V našej databáze sa nachádzajú patenty z rôznych oblastí. Od pohrebníctva (kategória 027) cez včelárstvo (169) až po horológiu (968). Toto delenie sme použili ako prvotný filter na zbavenia sa očividne nesúvisiacich patentov.

Naša databáza patentov obsahuje tri druhy citácií. Citácie iných amerických patentov, citácie iných neamerických patentov a citácie iných zdrojov. Citácie medzi patentami pre nás nie sú zaujímavé. Keď patent cituje článok, dostávame priamu informáciu o tom, že s ním súvisí. Tieto citácie majú v praxi niekoľko významov. Prvým je legislatívny. Je totiž potrebné vedieť, na akých ostatných objavoch patent závisí a ktorým teda treba pripísať zásluhy. Tak isto je dôležité vedieť, o aké teoretické základy sa patent opiera a z ktorých článkov ich čerpá.

Niekoľko výskumníkov sa už na tieto citácie začalo pozerať ako na indikátor toku myšlienok. Idea je, že skúmaním citácií článkov je možné skúmať prechod myšlienok z teoretickej vedy, do reálneho sveta. To umožňuje hodnotiť vedecké žurnály podľa ich pridanej hodnoty do aplikovanej vedy, hodnotiť inštitúcie podľa ich efektivity alebo odhadovať stav rôznych vedeckých oblastí a identifikovať príležitosti v nich. Kľúčovým slovom je tu *knowledge flow*.

Toto postavenie patentových citácií nám dáva nádej, že môžu byť pre náš problém relevantné. Tieto dáta síce nepoužijeme pri samotnom hľadaní relevantných patentov, ale použijeme ich na testovanie našich metód. Ak totižto patent A citoval článok B, musel byť preň tento článok naozaj relevantný. Na druhej strane, ak by sme chceli nájsť všetky relevantné patenty pre článok B, tak medzi nimi pravdepodobne bude patent A. Na tejto úvahe môžeme postaviť vyhodnocovanie úspešnosti našich metód.

Podobne ako pri článkoch sme sa zbežnou prehliadkou uistili, že abstrakty dostatočne pokrývajú myšlienku patentu. Podobne sa stretávame s problémom špecifickej a technickej slovnej zásoby.

Kapitola 2

Teoretické východiská a súčasný stav

V tejto kapitole si zhrnieme teoretické základy, ktoré budeme potrebovať pri implementácií nášho systému. Tiež sa oboznámime so súčasným stavom používania tejto teórie v praxi.

2.1 Teória vyhľadávania dokumentov

Nasledovné definície sú z veľkej časti prebrané a voľne preložené z knihy *Introduction to information retrieval*[16].

Získavanie dokumentov (document retrieval) je hľadanie materiálov (väčšinou dokumentov) neštruktúrovaného charakteru (väčšinou text) vo veľkej kolekcii (korpusu, väčšinou uloženého na počítači), ktoré spĺňajú informačné potreby používateľa.

Pod pojmom dokument myslíme akúkoľvek jednotku, nad ktorou sa rozhodneme vyhľadávať. Môže sa teda jednať o krátky text, kapitolu alebo aj celú knihu.

Bežne poznáme dokumenty ako postupnosti znakov. Táto reprezentácia ale pre počítače nie je veľmi vhodná. Väčšinou nás totiž nezaujímajú jednotlivé znaky, ale len slová. Tie dokážeme jednoduchým spôsobom kódovať ako čísla. Zoberieme všetky slová v našom korpuse a postupne im priradíme rôzne prirodzené čísla. Toto priradenie si zapamätáme ako slovník a dokumenty si potom môžeme pamätať ako postupnosti čísel. Ak by náš korpus obsahoval len dve vety: *Antibiotiká liečia infekcie. Antibiotiká zabíjajú baktérie.*, slovu *Antibiotiká* by sme priradili číslo 0, *liečia* 1, *infekcie* 2 a tak ďalej. Zakódovaný korpus by vyzeral ako $0, 1, 2, 0, 3, 5$. Týmto spôsobom sme všetky slová reprezentovali ako body v jednorozmernom priestore.

Pre množstvo metód na prácu s textom je ale vhodnejšie, aby malo každé slovo vlastnú dimenziu. To ale znamená, že reprezentácia jedného slova by mala toľko dimenzií, koľko je slov v slovnej zásobe nášho korpusu. Ak by sme dokument reprezentovali

ako postupnosť reprezentácií jednotlivých slov, dostali by sme veľmi veľa rozmerné reprezentácie. Ďalej teda zabudneme na poradie slov v dokumentoch a budeme si ich pamätať len ako vrecia slov, *bag of words*. Dokument bude vektor, ktorého i-ty prvok bude určovať, či sa v ňom nachádza slovo s kódom i z nášho slovníka. Tento vektor bude pravdepodobne obsahovať veľa núl a v praxi je teda lepšie si ho pamätať ako riedku maticu. Náš korpus je potom len jedna veľká riedka matica, ktorej riadky sú vrecia slov dokumentov. Túto maticu nazývame termínová matica, *term matrix*.

Relevantný dokument je taký, ktorý používateľovi podľa jeho mienky prináša informačnú hodnotu s ohľadom na jeho informačné potreby.

Keďže ale presne nepoznáme informačné potreby užívateľa, naším najlepším tipom je hľadať dokumenty, ktoré sú si navzájom podobné. To, ako si definujeme podobnosť dokumentov, sa priamo premietne na metódach, ktoré použijeme.

Model logického získavania (boolean retrieval model) je model informačného získavania, pri ktorom môžeme položiť dotaz vo forme, v ktorej sú hľadané termíny spojené logickými spojkami konjunkcie, disjunkcie a negácie.

Predstavme si kolekciu medicínskych článkov. Príkladom logického získavania by bolo nájdenie všetkých článkov, v ktorých sa spomínala rakovina pľúc, ale nespomínalo sa v nich fajčenie. Hodnoty v termínovej matici potom môžeme chápať ako logické hodnoty daného slova. Pri vyhodnotení dotazu stačí len vyhodnotiť jeho logickú hodnotu pre všetky dokumenty. Do formuly dosadíme logické hodnoty slov v danom dokumente. Nakoniec vrátime tie dokumenty, pre ktoré bola vyhľadávaná logická formula pravdivá.

Najväčším problémom tejto metódy je, že je príliš binárna. Nie všetky slová totiž majú pre nás rovnaký význam a bolo by dobré, keby s tým naša metóda dokázala pracovať. To, či sa v dokumente nachádza nejaká spojka, nás vôbec netrápi, ale či je tam rakovina je celkom podstatné. Podobne nám prekáža aj binárnosť výstupu. Ak budeme mať veľa relevantných dokumentov, tak nevieme, ktoré z nich sú najrelevantnejšie, respektíve ktoré nerelevantné dokumenty máme použiť, ak nám nebudú stačiť tie relevantné.

Ak by sme sa obmedzili len na logické formuly bez konjunkcií, vedeli by sme si vyhodnocovanie formúl uľahčiť. Môžeme zobrať termínový vektor pre hľadaný dotaz a ľubovoľnú metriku. Pre každý dokument potom zrátame jeho vzdialenosť od dotazu a máme usporiadanie dokumentov podľa podobnosti. Ak by sme si ako metriku zvolili skalárny súčin, dostali by sme presne usporiadania podľa logických hodnôt. Možnosťou by mohla byť aj euklidovská vzdialenosť, ktorá má ale pre porovnávanie veľa rozmerných vektorov zlé vlastnosti.

V praxi sa na porovnávanie veľa rozmerných vektorov používa kosínusová podobnosť. Tá vyjadruje uhol, ktorý medzi sebou zvierajú dva vektory. Pridáme ale ešte korekciu, aby nám táto podobnosť pre rovnaké vektory vyšla nulová a dostávame vzťah

$$cos(a, b) = 1 - \frac{a.b}{\|a\|.\|b\|}$$

Za zmienku stojí, že pre dva náhodné vektory na veľa rozmernej jednotkovej guli je táto kosínusová vzdialenosť takmer vždy veľmi blízka 1.

My ale nehľadáme podľa dotazu alebo logickej formuly, ale podľa iného dokumentu. Keďže máme na vstupe dokument, nie je jasné, aký dotaz chceme vyhľadávať. Prvá možnosť je použiť všetky slová v dokumente. Tým ale dávame rovnakú váhu všetkým slovám. Zo skúsenosti ale vieme, že nie všetky slová prispievajú k významu textu rovnako. Pojmová matica ale môžeme obsahovať aj iné čísla ako jednotky a nuly. Tie potom budú zodpovedať tomu, ako bolo dané slovo podstatné v našom dokumente.

Ak je nejaké slovo dôležitejšie, bude mať v pojmovej matici väčšiu hodnotu. Pri rátaní podobnosti potom bude mať toto slovo väčšiu váhu. Presne na to slúži $TFIDF$. $TFIDF$ je skratka pre *term frequency -inverse document frequency* a ide o číselnú hodnotu, ktorá má zachytiť, ako významné je slovo pre dokument v korpuse. Táto hodnota sa skladá z dvoch členov. Člen TF vyjadruje relevantnosť slova pre konkrétny dokument a člen IDF vyjadruje relevantnosť slova v kontexte celého korpusu. Myšlienkou je, že dôležitosť slova závisí od jeho počtu výskytov v dokumente (TF). Čím viac sa slovo vyskytuje v dokumente, tým relevantnejšie by preň malo byť. Slová, ktoré sa ale vyskytujú vo veľa dokumentoch, neprinášajú veľa informácie o týchto dokumentoch (IDF), a preto ich chceme penalizovať. Ak by sa v každom dukomente vyskytovalo slovo rakovina a len v niekoľkých slovo pľúca. Potom pre dokumet, v ktorom sa vyskytuje rakovina pľúc, je ovaľa podstatnejšie slovo pľúca ako rakovina.

Označme si $tf(t, d)$ ako počet výskytov slova t v dokumente d, n počet všetkých dokumentov, s_d počet slov v dokumente d a n_t počet dokumentov, v ktorých sa nachádza slovo t. Potom pre $TF_{d,t}$ máme nasledovné používané možnosti:

a) $TF_{d,t} = sign(tf(t, d))$, prítomnosť alebo neprítomnosť slova v dokumente

b) $TF_{d,t} = tf(t, d)$, počet slov v dokumente

c) $TF_{d,t} = \frac{tf(t,d)}{s_d}$, tvar používaný pri dokumentoch rôznych dĺžok

d) $TF_{d,t} = 1 + log(tf(t, d))$

Ak se dané slovo v dokumente nenachádza, tak $TF_{d,t} = 0$. Pre IDF sú možnosti $IDF_t = 1$ a $IDF_t = log(\frac{n}{n_t})$. Váha slova t v dokumente d je potom $TF_{t,d}.IDF_t$ pre niektorú kombináciu TF a IDF. Pri porovnávaní dokumentov sa nemusíme pozerať na váhy všetkých ich slov. Môžeme si namiesto toho pre každý dokument pamätať len

prvých k slov s najväčšou váhou. Tým docielime akési primitívne zníženie dimenzie našej reprezentácie dokumentov.

Voľba členov v TF-IDF je síce intuitívna, no je problematické ju korektne teoreticky odôvodniť. Teoretici sa o to snažili [21, 6] pomocou pravdepodobnostného pohľadu, respektíve teórie informácie a kódovania, no narážali na problémy s definovaním správneho pravdepodobnostného priestoru.

Prístupy založené na exaktnom výskyte, poprípade absencii slova v dokumente majú niekoľko nevýhod. Sú citlivé na rôzne tvary slov, množné čísla, predpony, diakritiku, interpunkciu, veľké písmená a iné podobné zmeny, ktoré síce veľmi nemenia význam slova, ale menia jeho text. Jednoduchým zlepšením je predspracovanie textu. Jedná sa o odstránenie všetkých nealfabetických znakov a kapitalizácie písmen. Pri komplikovanejších metódach môžeme slová nahradiť lexémami, slovotvornými základmi slov. Tokenizácia je zaujímavá skôr z praktickej stránky, preto sa jej viac budeme venovať až v podkapitole 3.1

Zatiaľ sme za podobné dokumenty považovali len tie, v ktorých sa nachádza veľa rovnakých slov. Vieme ale, že v skutočnosti aj rôzne slová môžu mať podobný význam. Preto by bolo dobré, keby to naša metóda dokázala zohľadniť. Pri logickom vyhľadávaní sa dajú použiť synonymické slovníky a množstvo disjunkcií. Tým nám narastie dĺžka formuly, ktorú potrebujeme vyhodnocovať a pribudne nám dátový zdroj, o ktorý sa musíme starať. Podobne by sme pri $TFIDF$ vedeli do dokumentu umelo pridať ďalšie synonymá s vhodne upravenými váhami. Stále ale nedosiahneme požadovaný výsledok, nakoľko stále nebudeme schopní zachytiť slová, ktoré sú podobné, aj keď nie sú synonymá.

Riešením je algebraický model. Doteraz boli dimenzie slová, boli by sme radi, kedy dimenzie boli významy v nejakom sémantickom priestore. Potom dokumenty s rovnakým významom budú mať podobné vektory, a to je presne to, čo potrebujeme. Ako to docieliť? Vo všeobecnosti sa pokúsime nájsť nejakú menej rozmernú reprezentáciu dokumentu, pomocou ktorej sa bude dať čo najpresnejšie zrekonštruovať. To nás donúti skomprimovať informáciu, ktorú o dokumente máme. Potom nám ostáva len dúfať, že táto nová reprezentácia zachytí a vynesie na povrch skutočný význam dokumentu. Otázkou ostáva, ako presne túto novú reprezentáciu hľadať, ako chceme naspäť dokument rekonštruovať a čo bude znamenať, že sme ho zrekonštruovali dobre.

2.2 LSA - latentná sémantická analýza

Máme pojmovú maticu M pre náš korpus. Táto matica má v riadkov a n stĺpcov, pričom jej stĺpec d_i reprezentuje i-ty dokument. Radi by sme našli novú maticu s rozmermi $k \times n$, pričom $k < v$, z ktorej sa dá M naspäť zrekonštruovať.

Ak urobíme singulárny rozklad (SVD) pre maticu M, dostaneme tri nové matice U, Σ aV^T, pre ktoré platí

$$M = U\Sigma V^T$$

Pričom U a V sú ortogonálne a Σ je diagonálna matica kladných čísel. Tento proces nazveme trénovanie LSA. U má ale rozmery $v \times v$, Σ má $v \times n$ a V^T má $n \times n$. Σ má na diagonále singulárne hodnoty σ. Ukáže sa, že keď zoberieme len k najväčších singulárnych hodnôt, im prislúchajúcich k stĺpcov z U a k riadkov z V^T, dostaneme aproximáciu M ranku k s minimálnou odchýlkou pri Frobéniovej norme. Toto zapíšeme ako $M_k = U_k\Sigma_k V_k^T$. Tým vo V_k dostaneme vektory $\hat{d}_i$ ako menej rozmerné aproximácie pôvodných dokumentov.

Kosínusová podobnosť dokumentov z M_k je potom $M_k^T.M_k = V_k\Sigma_k U_k^T U_k\Sigma_k V_k^T = V_k\Sigma_k\Sigma_k V_k^T$. To znamená, že ak chceme porovnať dva dokumenty z M_k, stačí nám porovnať ich reprezentácie vo V_k preškálované pomocou Σ_k. Nanešťastie najväčšie členy v Σ_k zodpovedajú rozmeru vo V_k, ktorý býva silno korelovaný so sumou čísel v patričnom riadku M_k, teda neurčuje to, ktoré slová sa v danom dokumente nachádzajú. Aj tak je mu ale priradená veľká singulárna hodnota, a teda by mal mať pri porovnávaní veľký význam. My sme preto používali porovnávanie bez škálovania pomocou Σ_k. Tým pádom sme každému rozmeru v novom priestore dali rovnakú váhu. Ak teda chceme zistiť podobnosť stĺpcových vektorov d_i a d_j, stačí nám vyrátať $\hat{d}_i\hat{d}_j$.

Vieme, že pre dokumenty $\hat{d}$ z nášho korpusu platí $d = U_k\Sigma_k\hat{d}_k$, teda $\Sigma^{-1}U_k^T d = \Sigma^{-1}U_k^T U_k\Sigma_k\hat{d}_k$. Keďže je U singulárna platí $\hat{d} = \Sigma^{-1}U^T d$. Vyrátať Σ_k^{-1} nie je problém, nakoľko Σ_k je diagonálna, tak Σ_k^{-1} je tiež diagonálna pozostávajúca z obrátených hodnôt čísel z diagonály Σ.

Ak chceme hľadať dokumenty podľa dotazu, môžeme sa na dotaz pozerať ako na dokument. Zakódujeme ho pomocou LSA a nájdeme patenty, ktoré majú od neho najmenšiu vzdialenosť. Tie potom môžeme usporiadať a postupne ponúkať užívateľovi ako naše odporúčania.

Drobnou nevýhodou je, že nedokážeme ľahko meniť matice U a Σ pre nový dokument. Podobne nevieme ani pridávať nové slová do slovnej zásoby. To by nemal byť problém, pokiaľ bola naša pôvodná matica M dosť veľká. Okrem toho máme ešte niekoľko teoretických problémov, ktorých sa nedokážeme zbaviť. Výsledný priestor nemá rozumnú ľudskú interpretáciu. Nedokážeme zachytiť slová s viacerými významami. V skutočnosti predpokladá Gausovo rozdelenie tam, kde bolo pozorované Poissonovo rozdelenie. Kosínusová podobnosť nie je určená na distribúciu, ktorú budú mať naše dokumenty v novom priestore.

V našom prípade sú články z inej distribúcie ako patenty. Vyvstáva potom otázka, na čom by sa mala LSA natrénovať a aké $TFIDF$ váhy by sme mali použiť. Tieto otázky sa väčšinou ignorujú s tým, že buď máme všetky dokumenty z rovnakej distribúcie,

alebo sa tak aspoň tvárime. Problém je, že akonáhle sú články z inej distribúcie, nevieme o nich vlastne po teoretickej stránke nič povedať. Možnou alternatívou by bola natrénovať *LSA* všetkých dokumentoch naraz. Tam narazíme na problém s veľkým množstvom dát (článkov je približne 30 000 000). Robiť SVD na takto veľkej pojmovej matici nezvládame. Preto by sme museli voliť nejaký spôsob výberu len časti článkov a patentov, čo by prinášalo ďalšie hyperparametre, ktoré by sme mohli potenciálne zvoliť zle.

2.3 Vyhodnocovanie úspešnosti

Tu sa oboznámime s problémom vyhodnocovania úspešnosti algoritmov na vyhľadávanie dokumentov.

Definícia 2.3.1 Presnosť (precision) *algoritmu je podiel naozaj relevantných dokumentov zo všetkých, ktoré boli nájdené algoritmom.*

Definícia 2.3.2 Senzitivita (recall) *algoritmu je podiel nájdených relevantných dokumentov a všetkých relevantných dokumentov v korpuse.*

Uvažovať len o presnosti alebo len o senzitivite je nebezpečné. Najlepšiu senzitivitu totiž vieme ľahko dosiahnuť, keď prehlásime všetky dokumenty za relevantné. Podobne vieme najlepšiu presnosť dosiahnuť vtedy, keď za relevantný neprehlásime žiaden dokument. Presnosť aj senzitivitu môžeme vyjadriť pomocou F_1 skóre.

Definícia 2.3.3 F_1 skóre *algoritmu je podiel súčinu jeho presnosti so senzitivitou a súčtu jeho presnosti so senzitivitou.*

Tieto tri definície poskytujú základný nástroj na meranie kvality klasifikačného algoritmu. V našom prípade chceme dokumenty skôr zoraďovať ako klasifikovať. Okrem toho by sme presnosť, senzitivitu a ani F_1 skóre v takýchto formách nevedeli použiť. Nevieme totiž (a v rozumnom čase to ani nemáme ako zistiť), koľko dokumentov v korpuse ja naozaj relevantných. V praxi užívateľovi neukážeme všetky zoradené dokumenty, ale len niekoľko prvých. Preto je rozumnejšie definovať trochu inú presnosť.

Definícia 2.3.4 Presnosť@x (precision at x), $P@x$ *algoritmu je podiel naozaj relevantných dokumentov z prvých x, ktoré boli algoritmom nájdené.*

$P@x$ ale nezohľadňuje usporiadanie v prvých x odporúčaniach. Ak by nás zaujímala $P@2$ a prvý dokument by bol relevantný a druhý nie, dostali by sme rovnakú hodnotu ako keby bol relevantný len druhý dokument.

Definícia 2.3.5 Vážená Presnosť@x (average precision at x), $AP@x$ *algoritmu je* $\sum_{i=1}^{x} P@i \triangle r_i$.

$\triangle r_i$ je zmena senzitivity, ktorá má hodnotu $\frac{1}{x}$, ak bol i-ty dokument relevantný, inak 0.

Ak by sme odporúčali dva dokumenty a relevantný by bol len prvý $AP@x = 1\frac{1}{2} + \frac{1}{2}0 = \frac{1}{2} = 0,5$. Ak by bol relevantný len druhý, dostaneme $AP@x = 0,0 + \frac{1}{2}\frac{1}{2} = \frac{1}{4} = 0,25$.

Definícia 2.3.6 Priemerná vážená Presnosť@x, (mean average precision at x), $MAP@x$ *algoritmu je* $\sum_{dotaz} AP@x$.

$MAP@x$ je teda len priemerom $AP@x$ cez všetky dotazy ktoré sme urobili a určuje efektívnosť nášho algoritmu na všetkých dotazoch, na ktorých ho testujeme. Nanešťastie takéto testovanie nie je najlepšie. Najlepšie by bolo, keby sme mohli nechať každú metódu odporučiť niekoľko najrelevantnejších patentov, a tieto odporúčania potom skontrolovať.

2.4 Súčasný stav

V tejto kapitole si predstavíme niektoré konkrétne riešenia podobných problémov a ich možné nevýhody a úskalia. Pre podrobnejší prehľad súčastného stavu odporúčame článok *A survey of current work in biomedical text mining*[10].

Počas nášho hľadania sa nám nepodarilo nájsť článok alebo nástroj, ktorý by riešil presne náš problém, teda mal na vstupe článok a na výstupe zoznam k nemu relevantných patentov. Identifikovali sme ale niekoľko hlavných smerov, ktorými sa výskum v tejto oblasti uberá. Tiež uvedieme, prečo sme sa týmito smermi neuberali, respektíve, ako sme z nich čerpali pri našom riešení.

Prvým smerom je hľadanie relevantných dokumentov podľa používateľom zadaného dotazu. Tu sa kladie dôraz hlavne na optimalizovanie databázových systémov, objavovanie nových metód indexovania, ukladania dát a vyhodnocovania rôznych filtrov. Systémy zaoberajúce sa hlavne prehľadávaním neštruktúrovaného textu sa nazývajú plno textové, *full text*, vyhľadávacie systémy. Takýmto systémom je Postgres[19] so svojimi rozšíreniami špeciálne určenými na tieto účely. Ich základom je kvalitná tokenizácia textu na lexémy, ktorá je závislá od jazyka písaného textu. Následne je dokument reprezentovaný ako *tsvector*, čo je vektor na textové vyhľadávanie[20]. Druhým podobným systémom je ElasticSearch. Tieto systémy sú veľmi podobné ako naša metóda $TFIDF$ a taktiež majú problém s rôznymi slovami s rovnakým významom. Pri použití $TFIDF$ ale máme väčšiu kontrolu nad spôsobom hodnotenia podobnosti. Na druhej strane sú tieto metódy oveľa rýchlejšie a pravdepodobne pamäťovo efektívnejšie.

Stále ide len o akýsi druh textového vyhľadávanie veľmi podobný logickému. Absentuje v ňom pochopenie skutočného významu slov a spolieha sa len na absolútnu textovú zhodu. Je snaha tieto problémy prekonať a nájsť spôsob, ako by sa význam slov dal počítačovo reprezentovať a ako by sa s ním dalo ďalej pracovať. Landauer[14], Mikolov[17] ako aj iní [15, 25] publikovali práce, v ktorých je prezentovaný spôsob reprezentácie zmyslu slov pomocou vektorov. Tie potom ponúkajú premostenie medzi matematickými vektorovými operáciami a významom slovných vzťahov, ktoré sú nimi reprezentované. Jednou z možností bolo namiesto LSA použiť takéto komplikovanejšie reprezentácie. Tieto metódy sú výpočtovo oveľa náročnejšie a obsahujú veľké množstvo hyperparametrov.

Okrem toho skrývajú jeden časový problém pri vyhľadávaní. Keď sme chceli hľadať pri textovom vyhľadávaní, mohli sme použiť techniku spätného indexovania. Pri nej si pamätáme, v ktorých dokumentoch sa nachádzalo dané slovo. Vďaka tomu sa nemusíme pozerať na všetky dokumenty v našom korpuse. Ak ale dotaz zakódujeme do sémantického vektora a hľadáme k nemu tie najbližšie z nášho korpusu, spätné indexovanie je nepoužiteľné. Ostáva nám len porovnať náš dotaz so všetkými vektormi. Otázka je, či to nejde aj rýchlejšie. Problémom v literatúre je, že veľa experimentov sa robili na datasetoch s len niekoľko tisíc dokumentami. Tam je čas potrebný na lineárne porovnanie naozaj zanedbateľný.

Aby sme využili poznatky o vyhľadávaní dokumentov podľa dotazov, museli by sme vedieť článok premeniť na dotaz, respektíve z neho vedieť extrahovať podstatnú informáciu, ktorá je v ňom obsiahnutá. Presne tieto problémy rieši automatická sumarizácia dokumentov[11, 24]. Zaoberá sa technikami zjednodušovania dokumentov a identifikácie podstatných informácii v nich. Môžeme teda skúsiť vygenerovať sumarizáciu článku a tú potom použiť ako dotaz pre vyššie uvedené nástroje.

Ďalším smerom je vyhľadávanie patentov podľa patentov, poprípade článkov podľa článkov. Tento smer sa stretáva s uplatnením hlavne pri posudzovaní podaných patentov. Je totižto potrebné zistiť, či patent naozaj priniesol niečo nové, prípadne aké štúdie sú s ním spojené. Taký je napríklad nástroj *Google prior art search* [5]. Ten z textu extrahuje kľúčové slová a frázy, ktoré následne použije ako dotazy do ostatných Google vyhľadávacích nástrojov. Toto je ale len spojenie postupov popísaných vyššie.

Výskum sa robí aj v oblasti zapojenia ďalších zdrojov informácií, ktoré máme o dokumentoch. Takými sú napríklad citácie. Fujji[12] ukázal metódu na kombinovanie textového vyhľadávanie s dátami o citáciách. Následne ju použil na hľadanie patentov, ktoré by mohli ohroziť uznanie nového patentu.

Čo sa týka prepojenia patentov a článkov, zaujímavý je prechod myšlienok z vedy do praxe. Skupina výskumníkov [23] ukázala, ako sa dujú prepájať témy v patentoch s témami v článkoch, a teda že súvis medzi článkami a patentami sa naozaj dá odhaliť len z ich textov. Tento výsledok je pre nás veľmi dvôležitý a dáva nám nádej, že

vyhľadávanie patentov k článkom je naozaj možné.

Záverom je, že sa nám nepodarilo nájsť žiaden dostupný nástroj, ktorý by riešil náš problém. Existuje množstvo systémov určených na hľadanie patentov alebo článkov. Všetky sú ale založené na vyhľadávaní podľa užívateľom napísaných dotazov. Tieto dotazy majú formu čistého textu alebo logickej formuly (slová spojené rôznymi logickými operátormi).

Kapitola 3

Naše riešenie

V tejto kapitole oboznámime čitateľa s tým, aké riešenia sme použili, s ich implementačnými detailami a samozrejme aj ich úspešnosťou.

3.1 Príprava dát

Prvým krokom bolo získať dáta o publikáciach a predspracovať ich. Projekt SciCurve už disponuje databázou článkov spolu s nástrojmi na prácu s nimi. Tieto nástroje nám poslúžili na jednoduché nájdenie identifikátorov, názvov a abstraktov článkov. Získavanie patentov bolo problematickejšie. Projekt SciCurve používal ako zdroj patentov stránku Google [3], ktorá sa ale prestala aktualizovať. Preto sme museli naprogramovať nástroje na sťahovanie z iného zdroja, konkrétne zo stránky REED [1].

Išlo o jednoduchý skript, ktorý postupne sťahoval správne súbory s patentami a ukladal ich na disk. Súbory boli skomprimované a nachádzali sa v nich zreťazené patenty vo formáte xml. Bolo teda nutné ich rozbaliť, rozdeliť v nich text na jednotlivé xml záznamy a nájsť v nich potrebné informácie. Chceli sme odfiltrovať patenty netýkajúce sa biomedicíny a patenty, ktoré neboli uznané. Uznané boli patenty, ktorých koreňový xml element bol *us-patent-grant*. To, či sa patent týkal biomedicíny, sme zistili podľa toho, do akých tried ho UPSTO zaradilo. Ponechali sme si len patenty s triedami: *424 Drug, bio-affecting and body treating compositions; 435 Chemistry: molecular biology and microbiology; 436 Chemistry: analytical and immunological testing; 514 Drug, bio-affecting and body treating compositions; 530 Chemistry: natural resins or derivatives, peptides or proteins, lignins or reaction products thereof; 536 Organic compounds – part of the class 532-570 series; 600 Surgery; 604 Surgery; 702 Data processing: measuring, calibrating, or testing; 800 Multicellular living organisms and unmodified parts thereof and related processes*. Táto kategorizácia je pre každý patent popísaná v elementoch *classification-national*, pričom sme používali len klasifikáciou, ktorá mala nastavenú krajinu na americké štáty, teda element *country* mal hodnotu

US. Názov článku sa nachádzal v xml elemente *invention-title* a abstrakt v elemente *abstract*.

Na spracovanie textu sme použili štandardné postupy. Odstránili sme nepodstatné slová (stop words), diakritiku a interpunkciu, všetky písmená sme transformovali na malé a následne sme text premenili na zoznam slov. Za slovo sme považovali akýkoľvek reťazec čísel, písmen a pomlčiek, aby sme dokázali pracovať napríklad aj s chemickými zlúčeninami a odborným označením. Použili sme knižnicu nltk, ktorá obsahuje funkcie na normalizáciu a tokenizáciu textu (funkcia word_tokenize), ako aj veľký zoznam nepodstatných slov. Taktiež sme použili Stemer na odstránenie rôznych tvarov slov, ako sú napríklad množné číslo a časovanie. Konkrétne sme použili triedu SnowballStemmer("english") taktiež z knižnice nltk. [8].

Tabuľka 3.1: Charakteristika databázy patentov

Počet dokumentov	369 778
Počet všetkých slov	30 102 310
Priemerná dĺžka dokumentu	81
Počet unikátnych slov	244 017
Maximálny počet výskytov slova	2 786 987 (9, 2% slov)

V tabuľke 3.1 môžeme vidieť charakteristiku našej databázy patentov. Po odfiltrovaní slov, ktoré sa nachádzajú len v jednom dokumente alebo sú použité menej ako 5-krát, ostalo len 48 347 unikátnych slov. Slovná zásoba patentov je mierne iná ako v bežnej reči. Napríklad máme nové slová, ktoré sú veľmi často používané a z hľadiska informačnej hodnoty v kontexte patentov sú takmer bezcenné. Príkladmi takýchto slov (slová po normalizácií a stemovaní) sú: system, use, includ, devic. Príkladmi stredne používaných slov sú: color-cod, bullet, gradiomet, ortholog alebo zirconium.

Tabuľka 3.2: Charakteristika databázy článkov

Počet dokumentov	25 781 454
Počet všetkých slov	2 588 581 775
Priemerná dĺžka dokumentu	100
Počet unikátnych slov	16 303 664
Maximálny počet výskytov slova	169 983 513 (6, 5% slov)

V tabuľke 3.2 vidíme charakteristiku našej databázy článkov. Náš článkový korpus

obsahuje až 25 781 454 dokumentov a až 16 303 664 unikátnych slov. K počtu unikátnych slov prispievajú najmä označenia génov a chemických zlúčenín. Po odfiltrovaní slov, ktoré sa nachádzajú len v jednom dokumente, alebo sú použité menej ako 5-krát, ostalo 2 052 442 unikátnych slov. Tak isto ako pri patentoch máme nové často používané slová s malou informačnou hodnotou: patient, use. Príkladmi stredne používaných slov sú: doubled-haploid, bipolar-ii, sulfobetain, o-glucosid, dimethylarsin.

Vidíme, že počet unikátnych slov v článkoch je naozaj veľký. Keďže ale budeme hľadať len patenty, budeme používať len slová, ktoré sa vyskytujú v patentovej slovnej zásobe. Pre ostatné by sme totiž vedeli, aký majú v kontexte patentov význam.

3.2 Patentové citácie

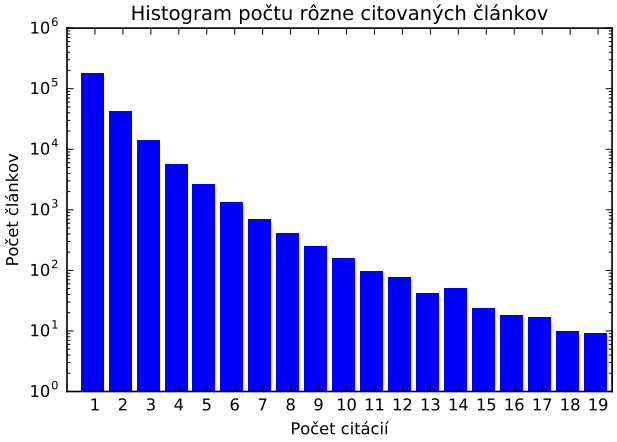
Databáza patentov obsahuje aj informácie o citáciách patentov. Ako sa ukázalo, použitie patentových citácií nie je až také jednoduché. Citácie patent-patent sú zaznamenané pomocou jednoznačného identifikátora citovaného patentu. Oproti tomu citácie článkov sú zaznamenané v podobe surového textu, ako napríklad *Abraham, E, et al., Lenercept (p55 tumor necrosis factor receptor fusion protein) in severe sepsis and early septic shock: A randomized, double-blind, placebo-controlled, multicenter phase III trial with 1,342 patients. Crit Care Med 2001 vol. 29, No. 3, p. 503-510.* alebo dokonca: *Neurosurg Psychiatry, 2003. 74(9): p. 1200-5.* Aj keď sa v tomto texte nachádza dostatok informácie na jednoznačné identifikovanie citovaného článku, je ťažké túto identifikáciu automatizovať. Získavaniu citácií v štruktúrovanej forme sa venovali Kousha a Thelwall [13]. Tých sme kontaktovali s prosbou na zaslanie ich dát, teda už normalizovaných citácií. Prvý-krát nám odpísali, že plné znenie článkov nemajú, čo sme v prosbe ani nespomínali, a na ďalšie maily už nereagovali.

Títo páni predstavili metódu pomocou vyhľadávača Bing a jeho api. Texty citácií používali priamo ako dotazy do vyhľadávača a hodnotili, s akou spoľahlivosťou im Bing vráti článok, ktorý bol naozaj citovaný. Nakoľko my máme prístup k celej databáze článkov, nemáme dôvod používať Bing. Naša databáza je postavená na technológii ElastickSearch, ktorá je priamo určená na textové vyhľadávanie. Stačilo nám teda použiť názov článku z citácie ako dotaz na našu databázu článkov. Ako toto vyhľadávanie urobiť naozaj precízne, by mohlo byť témou samostatnej práce. Pre naše účely ale môžeme obetovať veľkú časť senzitivity (nájsť naozaj všetky články) na úkor presnosti. Nepotrebujeme totižto nájsť všetky citácie, ktoré by sa nájsť dali.

Zvolili sme jeden z častých formátov citácií, z ktorého sa dal extrahovať názov publikácie. Následne sme sa zaoberali len citáciami, ktoré boli v tomto formáte. Konkrétne išlo o formát, v ktorom bola časť citácie v špeciálnych úvodzovkách \xe2 \x80 \x9c a \xe2 \x80 \x9d. Tú sme považovali za názov článku, ktorý sme sa pokúsili vyhľadať

v našej databáze článkov. Nakoľko používame textové vyhľadávanie, sme odolní voči drobným zmenám v názvoch ako sú veľké a malé písmená, interpunkcia a diakritika, dokonca drobné variácie textu.

Exaktnou presnosťou tohto hľadania sa pre naše účely nemá zmysel zaoberať. Ak sme aj našli nejaký iný článok, ktorý mal len rovnaké alebo veľmi podobné meno, pravdepodobne bude tiež relevantný.



Obr. 3.1: Histogram citovanosti článkov

Pred tým, ako sme získali tieto dáta, sme sa obávali, že citácií článkov bude relatívne málo a že bude len veľmi málo článkov, ktoré sú citované viacerými patentami. Podarilo sa nám získať 370 289 citácií z 45 188 rôznych patentov. Tieto patenty spolu citovali 250 013 rôznych článkov a najcitovanejší z nich mal až 60 citácií. Konkrétne počty článkov podľa počtu ich citácií vidíme v grafe 3.1. Vďaka tomu, že sme našli dostatočné množstvo článkov s viacerými patentovými citáciami, sme mohli vyrobiť testovacie dáta vhodné na testovanie úspešnosti našich metód. Chceme totižto hodnotiť, ako dobre naše metódy usporiadali relevantné a nerelevantné patenty, čo by sa robilo horšie, ak by sme mali vždy len jeden relevantný patent.

Z týchto dát o citáciách sme zostrojili testovaciu množinu. Zobrali sme články, na ktoré sa odkazovalo práve 5 patentov. Týchto článkov bolo 2611. Patenty, ktoré citovali daný článok, sme prehlásili za k nemu relevantné a pridali sme k nim ešte 95 náhodných patentov, ktoré sme prehlásili za nerelevantné. Tým sme dostali zoznam článkov a k nim relevantných, respektíve nerelevantných patentov. Použili sme len články s rovnakým počtom citácií kvôli tomu, aby boli naše dáta konzistentné a ľahšie sa nám na nich vyhodnocovala úspešnosť. Konštanty 5 a 95 boli zvolené kvôli tomu, že počet patentov s väčším množstvom citácií bol len polovičný. Na druhej strane sme chceli mať dostatočný počet relevantných patentov a aj zachytiť to, že v praxi je nerelevantných patentov oveľa viac ako relevantných.

3.3 Ručne anotované dáta

Každý, kto robil nejaký výskum v tejto oblasti, narazil na problém, že nemá naozajstné trénovacie, respektíve testovacie dáta, teda zoznam dokumentov, o ktorých vie, či sú si navzájom relevantné. Tieto dáta sú kľúčové pri testovaní a vyhodnocovaní skúmaných postupov. Preto je väčšina výskumníkov nútená k pracnému manuálnemu vytvoreniu takéhoto súboru dát. Tieto súbory, bohužiaľ, nie sú zverejnené. Aj napriek tomu, že sme získali dáta o citáciách, sme sa nakoniec k takémuto zberu dát uchýlili aj my. Našli sme kompetentných anotátorov, ktorými boli dve študentky medicíny. Následne sme ich nechali hodnotiť relevantnosť patentov ku konkrétnym článkom.

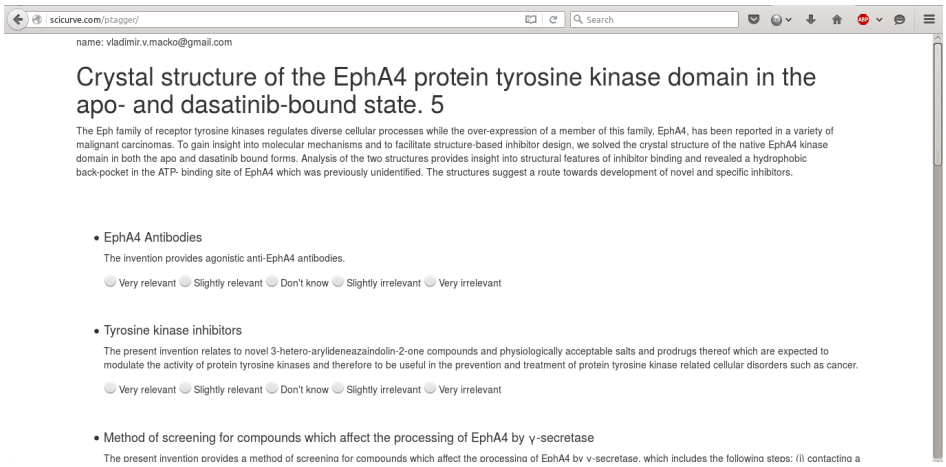
Otázkou bolo, ako presne ich chceme nechať hodnotiť. Možným riešením bolo ukázať anotátorovi náhodný patent s náhodným článkom a nechať ho rozhodnúť, či je táto dvojica relevantná alebo nie. Tento prístup má dve nevýhody. Veľká väčšina náhodných dvojíc je nerelevantná, a teda by sme len zbytočne mrhali časom nášho anotátora. Okrem toho je tento spôsob trochu neefektívny. Anotátor totižto musí na označenie jednej dvojice prečítať jeden článok a jeden patent.

Druhú nevýhodu sme sa rozhodli odstrániť tak, že sme anotátorovi predložili jeden článok a desať potenciálne relevantných patentov. Pre každý z nich sme ho potom nechali rozhodnúť, či je relevantný. Tým sa nám podarilo zmenšiť množstvo potrebného čítaného textu na jeden označený pár takmer na polovicu.

Anotátor mal pri označovaní päť možností. Mohol patent označiť ako relevantný, trochu relevantný, nerelevantný, trochu nerelevantný alebo to neurčiť. Označiť patent ako relevantný znamenalo, že išlo o naozaj dobré odporučenie, s ktorým by bol užívateľ pravdepodobne spokojný. Trochu relevantné boli patenty, ktoré sa síce týkali témy článku, ale nemuseli by užívateľa zaujímať. Trochu nerelevantné patenty boli tie, ktoré boli napríklad z príliš všeobecnej, alebo len veľmi málo podobnej oblasti. Nerelevantné patenty boli také, ktoré by sme užívateľovi rozhodne nechceli odporúčať. Toto jemnejšie rozdelenie sme použili pre prípad, ak by sme chceli s týmito anotovanými dátami ďalej pracovať.

Prvá nevýhoda sa dala adresovať ručným nájdením článkov a k nim relevantných a nerelevantných patentov. Toto je veľmi pracné a navyše by boli naše dáta veľmi skreslené spôsobom vyberania. Preto sme radšej zobrali tisíc článkov a pre každý z nich sme pomocou metódy $TFIDF$ našli desať potenciálne najrelevantnejších patentov. Tie sme potom v náhodnom poradí predložili pred anotátora.

Samotné anotovanie prebiehalo pomocou jednoduchého webového rozhrania priamo na stránke Scicurve. Ukážku tohoto rozhrania môžeme vidieť na obrázku 3.2. Anotátor sa prihlásil na stránku a išiel na adresu `http://scicurve.com/ptagger/`, ktorá je ešte stále aktívna. Tam sa mu zobrazil názov s abstraktom článku a zoznam názvov a abstraktov patentov. Pri každom patente bolo 5 tlačidiel na zaznamenanie relevantnosti.

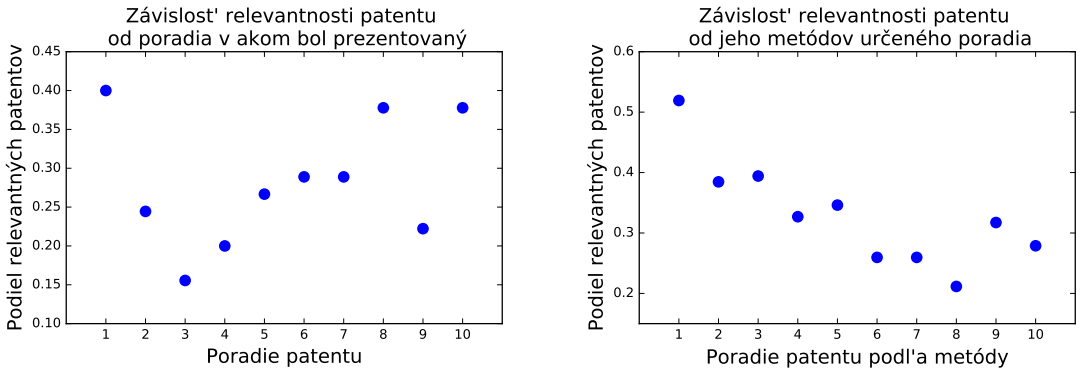


Obr. 3.2: Ukážka aplikácie na anotovanie

Keď bol anotátor hotový so všetkými desiatimi patentami prislúchajúcimi k jednému článku, potvrdil svoje hodnotenie a pokračoval na ďalší článok s ďalšími patentami. Zaznamenávali sme jeho odpovede a aj poradie, v akom boli patenty zoradené. Poradie článkov, aké anotátori dostávali, bolo rovnaké. Pre každý článok máme teda odpovede od dvoch anotátorov.

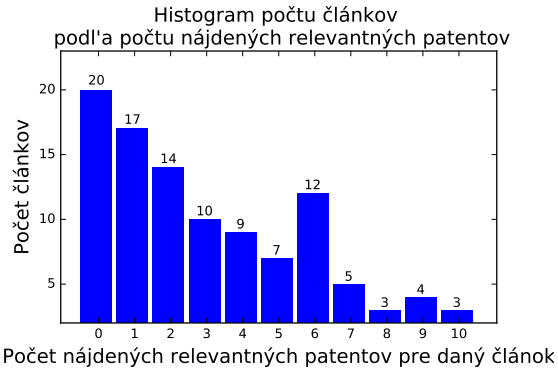
Uvažovali sme aj nad tým, že by sme anotátorov nechali porovnávať dve metódy. Nanešťastie sme nevedeli, ktorá kombinácia parametrov bude fungovať najlepšie. Preto sme ostali pri použití len jednej metódy na generovanie dát.

Vďaka spôsobu, akým sme vyberali patenty, vieme presne určiť úspešnosť metódy, pomocou ktorej sme tento výber robili. Anotátori označili spolu 1040 patentov. Relevantných bolo 5%, trochu relevantných bolo 27%, neurčiteľné boli 2%, trochu nerelevantných bolo 29% a nerelevantných bolo 34%. Presným vyhodnotením úspešnosti sa budeme zaoberať až v časti 3.5. Pre ďalšie účely sme za relevantný patent považovali ten, ktorý anotátri označili za relevantný alebo trochu relevantný.



Obr. 3.3: Graf závislosti anotátorovho hodnotenia od poradia

K prvým 59 článkom sme anotátorovi dali zoznam potenciálne relevantných patentov v takom poradí, v akom ich zoradila metóda, ktorá ich našla. Potom sme sa rozhodli, že budeme patenty predkladať pred anotátora v náhodnom poradí. Obávali sme sa totiž, že by poradie patentov mohlo ovplyvňovať to, ako ich bude anotátor hodnotiť. Napríklad by mal patent, ktorý bol prezentovaný neskôr, väčšiu šancu byť označený ako relevantný, pretože anotátor by sa čítaním predchádzajúcich patentov lepšie zorientoval v konkrétnej problematike a začal by si uvedomovať nové súvislosti. Z grafu ale môžeme usúdiť 3.3, že k takémuto skresleniu našich výsledkov pravdepodobne nedošlo, a ak aj áno, tak je to zanedbateľné oproti úspešnosti metód.



Obr. 3.4: Počet článkov podľa počtu nájdených relevantných patentov

Ako vidíme v grafe 3.4, pre 20 článkov sa nám nepodarilo nájsť medzi prvými desiatimi výsledkami ani jeden relevantný patent, pre 3 bolo relevantných všetkých 10 odporučení.

3.4 Implementácia metód

Pri implementácií sme sa rozhodli používať jazyk python. Máme s ním dobré skúsenosti a je to najčastejšie používaný jazyk v projekte SciCurve, čo nám uľahčuje integráciu do produkcie. Táto voľba sa nám vyplatila, nakoľko sme potrebovali robiť maticové operácie, kresliť grafy, prehľadávať niekoľko stránok a dokonca napísať webovú aplikáciu. Pri všetkých týchto činnostiach nám python výrazne pomohol. Pri analýze dát sme používali nástroj ipython notebook [18], ktorý nám umožnil agilnú prácu s dátami a ich jednoduché vizualizovanie.

Prvým problémom, ktorému sme sa venovali, bolo vyhľadávanie podobných vektorov. Chceme totiž pomocou nejakej metódy získať vektorovú reprezentáciu článku, pre ktorú chceme hľadať podobné reprezentácie patentov. Pre vektor článku chceme vedieť rýchlo nájsť najpodobnejšie vektory z množiny vektorov patentov. Problémom je, že patentov máme približne 300 000 a každému prislúcha približne 100 rozmerný vektor.

Obávali sme sa, že pozrieť sa na všetky tieto vektory, mať teda hľadanie v lineárnom čase, je príliš pomalé. Ukázalo sa, že keďže našou mierou podobnosti je kosínusová podobnosť, je jej rátanie ľahko vektorizovateľné a teda aj rýchle. Vďačíme za to hlavne knižnici numpy, v ktorej sme robili všetky maticové operácie.

Keď teda máme maticu M rozmerov 100×300000 a chceme nájsť indexy k vektorov, ktoré sú pri použití kosínusovej podobnosti najbližšie k vektoru q, s rozmermi 1×100, môžeme použiť nasledovný kód:

```
import numpy as np
from numpy.linalg import norm
def najpodobnejsi(M, q, k):
  #normalizujeme q
  nq = q / norm(q)
  #normalizujeme M
  nM = M / np.matrix(norm(M, axis=1)).T
  podobnosti = (1. - (nq*nM))
  #rychle ciastocne usporiadanie
  kandidati = np.argsort(podobnosti)[:k]
  return kandidati
```

Ako sa ukázalo, takýto výpočet trvá pre jeden dotaz len približne 60 milisekúnd, čo je na naše účely postačujúca rýchlosť. Pri použití riedkych matíc je možné použiť ekvivalentný kód s ekvivalentným časom behu.

Pokúsili sme sa aj o sofistikovanejšie riešenie pomocou zhlukovania. Konkrétne sme použili *sklearn.neighbors.NearestNeighbors*. Ten si dokáže vektory pamätať v štruktúrach, ktoré dokážu rýchlo hľadať najbližších susedov. Nanešťastie konštanta tohoto riešenia bola príliš veľká a čas hľadania sa ešte zhoršil. Vyhodnotenie jedného dotazu trvalo priemerne až 4 sekundy. Okrem toho narástli aj pamäťové nároky.

Sľubne vyzeralo aj použitie takzvaného *locality senzitiv hashing*. Ide o druh hašovacej funkcie, ktorá zachováva kosínusovú podobnosť. Nanešťastie je tento koncept aj podľa samotných autorov viac teoreticky zaujímavý ako prakticky použiteľný.

Čo sa týka LSA, vieme, že existujú knižnice, v ktorých je kompletne a kvalitne implementovaná. Takouto knižnicou je napríklad gensim, ktorá slúži na prácu s textom. My sme sa ale rozhodli použiť našu vlastnú implementáciu, aby sme si ju mohli ľahko prispôsobiť na naše konkrétne účely. Jednalo sa hlavne o nastavovanie váhových schém a používanie riedkych matíc.

Naša implementácia $TFIDF$ má dva hlavné parametre. Prvým je spôsob, akým budeme vyrábať pojmovú maticu, respektíve, ako budeme počítať váhy slov v nej. Implementovali sme niekoľko spôsobov, a to napríklad "bin", "tf", "tfidf", ïdf". Druhým parametrom je k, ktoré vyjadruje počet slov s najvyššou váhou ktoré uvažujeme pri

porovnávaní. Možné je totiž zobrať všetky slová, alebo len niekoľko najpodstatnejších.

LSA má taktiež parameter, ktorý určuje spôsob rátania váh slov. Druhým parametrom bolo, koľko dimenzií má mať matica V, a teda koľkorozmernú aproximáciu pôvodnej pojmovej matice budeme používať. Podobne ako pri $TFIDF$ sme ho označili ako k. Skúšali sme použiť hodnoty 50, 100, 150 a 200. Na rátanie SVD sme použili *scipy.sparse.linalg.svds*[7], ktorá dokáže pracovať s riedkymi maticami. Táto funkcia dokáže urobiť SVD rozklad priamo do požadovaného množstva dimenzií.

Spoločnou časťou oboch metód je vyrábanie termínových matíc pre korpus patentov a termínových vektorov pre články. Najprv potrebujeme pre všetky slová v našom patentovom korpuse určiť hodnotu n_t, teda počet dokumentov, v ktorých sa dané slovo nachádza, a priradiť im číselné identifikátory. Následne postupne spracovávame patenty. Pre každé slovo zo spracovávaného patentu zistíme jeho identifikátor a vyrátame jeho $TFIDF$ váhu, čím získame pojmový vektor. Keď všetky pojmové vektory spojíme do matice, dostaneme pojmovú maticu M patentového korpusu. Pri metóde LSA ešte potrebujeme vyrátať singulárny rozklad matice M na U, Σ a V.

Samotné hľadanie relevantných patentov k článkom potom pozostáva z nájdenia pojmového vektora daného článku a použitia kódu uvedeného vyššie. Pri $TFIDF$ hľadáme priamo v matici M, zatiaľčo pri LSA musíme pojmový vektor transformovať a následne vyhľadávať v matici V.

3.5 Vyhodnotenie

Každé naše testovacie dáto pozostávalo z jedného článku, niekoľkých relevantných patentov a niekoľkých nerelevantných. Každú metódu sme potom nechali usporiadať patenty podľa relevantnosti k patričnému článku. Pokiaľ priradila metóda dvom patentom rovnakú relevantnosť, zaradili sme všetky relevantné patenty za tie nerelevantné. Na tomto zoradení sme potom určili, aké dobré je.

Metriky, ktoré nás zaujímali, boli MAP@1, MAP@5 a MAP@10. MAP@1 vyjadruje šancu, že prvý odporučený patent bude relevantný. MAP@10 je všeobecne používaný hlavne kvôli webovým vyhľadávačom, ktoré zväčša ukazujú na prvej strane práve 10 výsledkov. MAP@5 sme použili preto, lebo my budeme na stránke ukazovať pravdepodobne 5 výsledkov. Tieto metriky sme vyhodnotili pre všetky metódy na citačnom datasete a priradili sme im ranky. Pre jednoduché porovnávanie sme následne metódy usporiadali podľa súčtu rankov. Kompletné výsledky sa pre rozmernosť tabuľky nachádzajú v prílohe A2. Výsledky pre prvých 5 metód vidíme v tabuľke 3.3.

Parameter k, v druhom stĺpci, má rôzne významy pre rôzne metódy. Pri LSA sa jedná o počet dimenzií matice V, ktorou sme aproximovali pôvodnú pojmovú maticu M. Pri metóde $TFIDF$ sme uvažovali len prvých k slov s najväčšími váhami. Pokiaľ

Tabuľka 3.3: Výsledky najúspešnejších metód na citačných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
TFIDF	0	tfidf_bot	**0.76(1)**	**0.49(1)**	0.56(2)
TFIDF	0	tfidf_pat	0.75(2)	0.49(2)	**0.56(1)**
TFIDF	30	tfidf_bot	0.75(3)	0.48(3)	0.54(4)
TFIDF	0	tfidf_pap	0.74(4)	0.48(4)	0.55(3)
TFIDF	30	tfidf_pat	0.74(5)	0.47(5)	0.54(5)

sa v type váhy nachádza tf, znamená to, že váha závisela od $TF_{d,t} = \frac{tf(t,d)}{s_d}$, ak sa v ňom nachádza *idf*, znamená to, že váha závisela od $IDF_t = log(\frac{n}{n_t})$. *bin* vyjadruje, že váha bola binárna, teda $TF_{d,t}IDF_t = sign(tf(t,d))$. Pri metóde LSA sme IDF váhy rátali podľa počtu výskytov slov v patentoch nezávisle na tom, či sme chceli zakódovať článok, alebo patent. Pri metóde $TFIDF$ sme experimentovali s použitím frekvencií výskytu slov v článkoch, dokonca s kombináciou výskytov v článkoch aj patentoch. Tieto experimenty obsahujú *_pat*, *_pap* respektíve *_bot*.

Jednoznačne najlepšia sa ukázala byť metóda $TFIDF$, ktorá sa v rôznych variáciách umiestnila na prvých osemnástich miestach. Ako najefektívnejšia sa ukázalo uvažovať všetky slová, v tabuľke zaznačené ako $k = 0$, a použiť $tfidf$ váhy. Na tom, či sme použili frekvencie slov z patentov alebo článkov, príliš nezáležalo. Z kompletných výsledkov vidíme, že čím menšie k bolo použité, tým menšia bola presnosť našej metódy. Podobný trend vidíme aj pri použití váh. Ako najhoršie sa ukázali byť váhy typu *bin*, nasledovali tf a *idf* ako najlepšie sa ukázali váhy *idf* vo všetkých variáciách.

Tabuľka 3.4: Výsledky metódy LSA na citačných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
lsa	100	idf	0.51(26)	**0.30(20)**	0.37(20)
lsa	200	tfidf	0.47(29)	0.30(21)	**0.39(18)**
lsa	200	idf	**0.52(23)**	0.30(22)	0.36(24)
lsa	50	idf	0.51(27)	0.30(25)	0.38(19)
lsa	150	idf	0.51(25)	0.30(23)	0.37(23)

Za úspech považujeme výsledok $MAP@1 = 0.76$ pre najlepšiu metódu. Sklamaním boli výsledky metódy LSA. Kompletné výsledky sa nachádzajú v prílohe A1. V tabuľke 3.4 si môžeme pozrieť výsledky najlepších parametrov. Taktiež ako pri LSA pozoru-

jeme, že s kleajúcim k klesá aj úspešnosť metódy. Zaujímavé ale je, že váhy typu *idf* skončili lepšie, ako váhy typu $tfidf$. Podľa nás je to spôsobené tým, že *idf* zachytávajú menšiu časť formy článku. Čo, ako ukážeme v kapitole 3.6, môže robiť metóde LSA problémy.

Metóda LSA pri optimálnej kombinácií parametrov dosiahla iba $MAP@1 = 0.52$, čo je oveľa menej ako $MAP@1 = 0.76$. Horšia úspešnosť LSA ja podľa nás spôsobená efektami, ktoré sme popísali v kapitole 2.2. Problémom je, že nevieme, ktoré dimenzie v matici V sú dôležité pre významovú podobnosť dokumentov. Potvrdilo sa nám, že použiť váhovanie pomocou Σ nie je prospešné. Ak sme ho použili, dosiahli sme pre optimálnu kombináciu parametrov zanedbateľných $MAP@1 = 0.018$. Presné výsledky takéhoto LSA pre všetky kombinácie parametrov ani neuvádzame.

Musíme ale pamätať na to, že úspešnosť na citačnom datasete nebude vyjadrovať presnú úspešnosť metódy v našom probléme. Tieto dáta totižto popisujú problém citovanosti a nie relevantnosti. Okrem toho by sa problém udaný našimi citačnými dátami dal považovať za oveľa ľahší ako hľadanie relevantných patentov v celom korpuse. Ak totižto patent citoval článok, pravdepodobne je s ním relevantný a je mu aj podobný. Na druhej strane je náhodný patent pravdepodobne nerelevantný a aj veľmi odlišný. Tým pádom nám chýbajú patenty, ktoré by boli podobné, ale nerelevantné. Ako uvidíme pri vyhodnotení na ručne anotovaných dátach, sú výsledky v týchto problémoch podobné.

Následne sme sa pozreli na úspešnosť našich metód na ručne anotovaných dátach. Časť výsledov pre metódu LSA môžeme vidieť v tabuľke 3.5, a pre metódu $TFIDF$ v tabuľke 3.6. Kompletné výsledky sa nachádzajú v prílohe A3 a A4. Vidíme, že poradie metód zostalo veľmi podobné a rozdiely môžeme pripísať šumu, nakoľko číselné hodnoty metrík pre jednotlivé metódy sú veľmi podobné. To znamená, že testovanie metód na citačnom datasete je do veľkej mieri konzistentné s testovaním na ručne anotovaných dátach.

Pri vyhodnocovaní úspešnosti na ručne anotovaných dátach musíme dávať pozor na interpretáciu našich výsledkov. Hlavným problémom je, že dáta, ktoré sme predkladali pred anotátora, neboli náhodné, ale boli to predpovede jednej z našich metód. Konkrétne sa jednalo o metódu $TFIDF$ s váhami $tfidf_pat$ a $k = 0$, čo je metóda, ktorá dosiahla na citačných dátach najlepšiu úspešnosť. Preto nemôžeme úspešnosť na týchto dátach pre ostatné kombinácie parametrov interpretovať priamo ako úspešnosť vo vyhľadávaní patentov, pretože ju nemeriame na reprezentatívnej vzorke. To sa týka len presných $MAP@$ hodnôt a poradie metód ostáva reprezentatívne, pokiaľ zohľadníme možné vplyvy šumu.

Keďže ručne anotované dáta presne určujú presnosť metódy $TFIDF$ s konkrétnymi parametrami, dostávame priamy odhad toho, ako súvisí úspešnosť metódy na citačných dátach s jej naozajstnou úspešnosťou pri vyhľadávaní relevantných patentov. Vidíme,

Tabuľka 3.5: Výsledky najúspešnejších metód na ručne anotovaných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
TFIDF	0	tfidf_pat	**0.58(2)**	**0.50(1)**	**0.58(1)**
TFIDF	20	tfidf_pat	**0.58(2)**	0.48(2)	0.57(3)
TFIDF	5	tfidf_pat	0.56(5)	0.48(3)	0.58(2)
TFIDF	30	tfidf_pat	0.54(8)	0.47(4)	0.57(4)
TFIDF	0	tfidf_bot	0.56(5)	0.46(7)	0.56(6)

Tabuľka 3.6: Výsledky metódy *LSA* na ručne anotovaných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
lsa	100	idf	0.46(21)	**0.44(13)**	**0.55(12)**
lsa	150	idf	0.46(21)	0.42(20)	0.54(15)
lsa	150	tfidf	**0.47(17)**	0.42(19)	0.53(22)
lsa	50	idf	0.41(28)	0.43(18)	0.54(16)
lsa	200	tfidf	**0.47(17)**	0.41(24)	0.52(27)

že úspešnosť $MAP@1 = 0.58$ na anotovaných dátach zodpovedá úspešnosti $MAP@1 = 0.75$ na citačných dátach. Tento záver môžeme urobiť len vďaka tomu, aké jednoduché boli naše metódy. Ani jednej z nich sme totižto nedali informáciu o tom, čo v skutočnosti znamená, že patent je relevantný k nejakému článku. Pri trénovaní metódy sme totižto nepoužili žiadne citačné a ani anotované dáta. Naše metódy len hľadali významové podobnosti, čo sa ukázalo ako postačujúce.

K úspešnosti sme dostali aj vyjadrenie od anotátorov. Hlavným problémom podľa nich bolo, že naša metóda niekedy prisúdila najväčšiu váhu slovu, ktoré nebolo najpodstatnejšie alebo nezachytilo konkrétnu tému článku. Pre článok o chorobách pri tehotenstve napríklad odporučila patent na tehotenskú spodnú bielizeň. Toto správanie je očakávateľné, keďže naše váhovanie slov je veľmi jednoduché. Optimálne by bolo, keby sme presne vedeli povedať, ktoré slová sú pri porovnávaní relevantnosti naozaj najdôležitejšie. Na to by sme mohli použiť niektoré z metód strojového učenia. Problémom je, že by sme potrebovali každému slovu v našej slovnej zásobe priradiť jedno číslo, takže by sme potrebovali natrénovať približné 250 000 parametrov, pričom máme k dispozícií približne 270 000 citácií. Vyriešili by sme tým len malú časť problému. Ako sme videli v príklade s tehotenskou bielizňou, problém nie je, že kľúčové slovo tehotenstvo by bolo

zlé, len na zachytenie významu v danom kontexte jednoducho nestačí. Tieto problémy mala prekonať metóda LSA, ktorej výsledky ale neboli také dobré, ako sme očakávali.

3.6 LSA dimenzie

Hlavným dôvodom neúspechu LSA podľa nás bolo, že niektoré dimenzie nového priestoru nemusia vyjadrovať význam, ale skôr formu dokumentu. Potom by tieto dimenzie nemali byť podstatné pre hodnotenie relevantnosti. Potrebovali by sme teda nájsť také váhy jednotlivých dimenzií, ktoré nám umožnia najlepšie predikovať relevantnosť dokumentov.

Vďaka tomu, že používame kosínusovú podobnosť to nie je problém. Kosínusová podobnosť je vlastne len skalárny súčin dvoch vektorov, čo je sume prvkov vektorového súčinu daných vektorom. Pre jeden článok a jeden patent teda máme vektor reprezentujúci tento súčin a chceme nájsť také váhy jednotlivých dimenzií, ktoré najlepšie klasifikujú túto dvojicu ako relevantnú, respektíve nerelevantnú. Presne s tým nám vie pomôcť lineárna regresia.

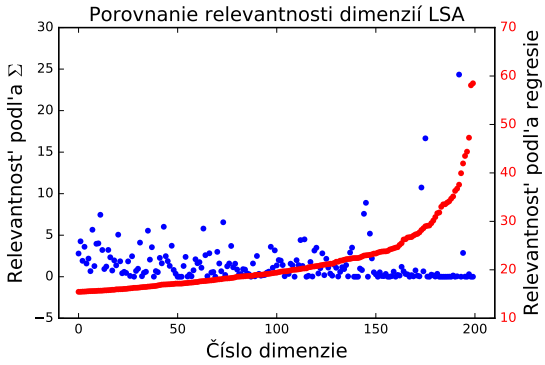
Pre každú dvojicu patent-článok z citačného datasetu sme zrátali vektorový súčin reprezentácií jednotlivých dokumentov v priestore danom LSA. Vektory sme následne použili ako vstupy pre lineárnu regresiu, ktorá mala predikovať relevantnosť patentu k danému článku.

Rozhodli sme sa takýmto spôsobom optimalizovať kombináciu parametrov metódy LSA kde $k = 200$ a váhy boli tfidf. Aj keď táto kombinácia parametrov nemala najlepšie výsledky, rozhodli sme sa pre ňu kvôli veľkej hodnote k, vďaka ktorej budeme mať väčšiu šancu, že nájdeme aj naozaj relevantné dimenzie.

Na trénovanie sme použili *sklearn.linear_ model.LogisticRegression* so štandardnými parametrami. Po natrénovaní sme dostali váhy, ktoré by mali zodpovedať podstatnosť jednotlivých dimenzií pre náš problém. Ako vidíme v grafe 3.5, tieto čísla nezodpovedajú hodnotám v matici Σ pre našu metódu. Predpoklad, že niektoré dimenzie zachytávajú pre nás nepodstatné informácie sa potvrdil. Naše rozhodnutie nepoužiť škálovanie pomocou Σ sa teda ukázalo ako naozaj správne.

3.7 Zhodnotenie výsledkov a diskusia

Ako najlepšia sa ukázala metóda $TFIDF$ ktorá uvažovala všetky slová ($k = 0$) a používala váhy typu $tfidf$. Táto metóda mala najlepšie výsledky na citačnom datasete, ktorý podľa nás dobre odzrkadľuje relevantnosť dokumentov, čo sa nám potvrdilo na výsledkoch na ručne anotovaných dátach. Táto metóda kvôli svojej jednoduchosti trpí niekoľkými nedostatkami. Tie sa nám nepodarilo odstrániť, nakoľko anotovaných



Obr. 3.5: Relevantnosť dimenzií LSA

dát, ktoré presne popisujú náš problém, máme málo. Pri citačných dátach, ktorých máme dostatok, riskujeme, že ak ich použijeme na trénovanie parametrov našej metódy, tak prestaneme riešiť problém relevantnosti a začneme predikovať citácie.

Nedostatky metódy $TFIDF$ mala prekonať metóda LSA. Tá, síce dosiahla uspokojivé výsledky ale $TFIDF$ neprekonala. Identifikovali sme aj hlavný problém, a spôsob ako ho vyriešiť. Na samotné riešenie sme ale nemali dostatok dát.

Naše výsledky pokladáme v kontexte vyhľadávania relevantných dokumentov za vierohodné, aj keď naše testovacie dáta popisovali trochu iný problém.

3.8 Ďalšie plány

Ďalším krokom je spustenie našej metódy priamo na stránke SciCurve. Aj keď už máme kód ktorý na to potrebujeme pripravený, k tomuto spusteniu ešte nedošlo. Naša metóda totiž potrebuje niekoľko gigabajtov operačnej pamäte, a na servery nám momentálne beží niekoľko pamäťovo náročných procesov. Okrem toho si ale ešte potrebujeme ujasniť spôsob, akým budeme zbierať informácie o interakcií užívateľov s našimi patentovými odporúčaniami. Chceme mať totiž kvalitnú spätnú väzbu o tom, či bol naše odporúčanie naozaj správne. Nie sme si ale istý, či nám bode stačiť len jednoduché podiel prekliku užívateľov, alebo na hodnotenie ich spokojnosti s výsledkami vymyslíme nejaký iný spôsob.

Keď získame dostatočné množstvo dát od užívateľov, plánujeme vyskúšať niektoré s komplikovanejších metód, ktoré potrebujú väčšie množstvo trénovacích dát. Dôležité je, že budeme mať trénovacie dáta presne pre problém, ktorý chceme riešiť. Jedným možným vylepšením našich metód je odporúčať patent na základe niekoľkých posledných článkov, ktoré užívateľ navštívil. Vďaka tomu by sme mali dostať presnejšiu informáciu o tom, aká paradigma paradigma je pre užívateľa naozaj zaujímavá.

Okrem toho plánujeme pokračovať v ručnom anotovaní dát. Anotovacia web stránka

je stále aktívna a chceli by sme, aby nám v anotovaní pomohli aj používatelia stránky SciCurve. Chceli by sme dať užívateľovi motiváciu, aby nám označil niekoľko dvojíc patentov a článkov, za čo sa mu potom odomkne ďalšia funkcionalita stránky.

Taktiež budeme pokračovať v hľadaní iných nových zdrojov dát o relevantnosti respektíve nerelevantnosti patentov a článkov.

Záver

Našu prácu hodnotíme ako úspešnú. Popísali sme podstatnosť problému hľadania relevantných patentov k článkom, našli sme výskum, ktorý ukazuje, že tento problém je riešiteľný a ukázali sme niekoľko možností na samotné riešenie. Tieto možnosti sme prehodnotili a vybrali z nich podľa nás tie najlepšie, ktoré sme implementovali a vyhodnotili ich úspešnosť.

Vysporiadali sme sa s dvomi hlavnými problémami. Prvým bola práca s dátami z reálneho sveta. Preto sme museli venovať veľa čas ich získavaniu, predspracovávaniu a normalizovaniu. Naskytla sa nám tým ale príležitosť získať z neštruktúrovaných dát informácie, ktoré nie sú v štruktúrovanej podobe nikde prístupné. Takto sme získali dáta o citáciách článkov v patentoch. Okrem toho sme pracovali s relatívne veľkým množstvom dokumentov, čomu sme museli podriadiť výber metód ktoré použijeme vo finálnom riešení.

Druhým problémom bola absencia vyhodnocovanie úspešnosti jednotlivých metód. Nemali sme totiž žiadne dáta o relevantnosti medzi patentami a článkami. Tento problém sme adresovali dvomi spôsobmi. Spracovali sme dáta o citáciách, čim sme získali relatívne veľké množstvo dát, ktoré ale nepopisujú presne problém, ktorý sa snažíme vyriešiť. Preto sme vytvorili aj ručne anotované dáta, ktoré sú na testovanie úspešnosti oveľa vhodnejšie, no máme ich oveľa menej.

So všetkými týmito problémami sa nám podarilo vysporiadať a implementovali sme niekoľko metód na hľadanie relevantných patentov. Tie sme vyhodnotili a našli najlepšiu metódu s optimálnou kombináciou parametrov. Aj keď boli naše metódy kvôli nedostatku kvalitných testovacích dát relatívne jednoduché, dosiahli sme $MAP@1 = 0.58$. Dokážeme teda uživateľovi odporučiť patent, ktorý bude relevantný na 58%. Pri odporúčaní piatich patentov sme dosiahli úspešnosť $MAP@5 = 0.50$.

Naše úsilie v tejto oblasti bude ďalej pokračovať získavaním nových dát od užívateľov a následným testovaním komplikovanejších metód.

Literatúra

[1] Databáza patentov. http://patents.reedtech.com/pgrbft.php. (navštívené dňa 20.4.2016).

[2] Databáza vedeckých článkov zo životných vied. http://www.ncbi.nlm.nih.gov/pubmed. (navštívené dňa 15.1.2016).

[3] Googlom zverejnená databáza patentov. https://www.google.com/googlebooks/uspto-patents.html. (navštívené dňa 15.1.2016).

[4] Zoznam desiatich najprofitujúcejších priemyslov v usa za rok 2014 podľa forbes. http://www.forbes.com/sites/liyanchen/2015/09/23/the-most-profitable-industries-in-2015/#6410eb587712. (navštívené dňa 2.3.2016).

[5] Nástroj od googlu na vyhľadávanie súvisiacich publikácií k patentom. https://www.google.com/patents/related, 2016. (navštívené dňa 27.1.2016).

[6] Akiko Aizawa. An information-theoretic perspective of tfidf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[7] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[8] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. Ö'Reilly Media, Inc.", 2009.

[9] BSR tím. Know your field of research! www.scicurve.com, 2014. (navštívené dňa 28.1.2016).

[10] Aaron M Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71, 2005.

[11] Hal Daumé III and Daniel Marcu. Induction of word and phrase alignments for automatic document summarization. *Computational Linguistics*, 31(4):505–530, 2005.

[12] Atsushi Fujii. Enhancing patent retrieval by citation analysis. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 793–794. ACM, 2007.

[13] Kayvan Kousha and Mike Thelwall. Patent citation analysis with google. *Journal of the Association for Information Science and Technology*, 2015.

[14] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[15] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[16] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[18] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007.

[19] PostgreSQL Global Development Group. PostgreSQL. `http://www.postgresql.org`, 2016. (navštívené dňa 23.1.2016).

[20] PostgreSQL Global Development Group. Úvod do práce s tsv vektormi a dátovými typmi. `http://www.postgresql.org/docs/8.3/static/datatype-textsearch.html`, 2016. (navštívené dňa 18.1.2016).

[21] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[22] USPTO. Návod na používanie portálu USPTO. `http://www.uspto.gov/sites/default/files/documents/7_Step_US_Patent_Search_Strategy_Guide_2015_rev.pdf`, 2015. (navštívené dňa 22.1.2016).

[23] Shuo Xu, Lijun Zhu, Xiaodong Qiao, Qingwei Shi, and Jie Gui. Topic linkages between papers and patents. In *Proceedings of the 4th International Conference on Advanced Science and Technology*, pages 176–183. Science and Engineering Research Support soCiety (SERSC), 2012.

[24] Xiaobing Xue and W Bruce Croft. Automatic query generation for patent search. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 2037–2040. ACM, 2009.

[25] Mo Yu and Mark Dredze. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242, 2015.

Appendix A

Tabuľka A1: Výsledky metódy *LSA* na citačných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
lsa	100	idf	0.51(26)	**0.30(20)**	0.37(20)
lsa	200	tfidf	0.47(29)	0.30(21)	**0.39(18)**
lsa	200	idf	**0.52(23)**	0.30(22)	0.36(24)
lsa	50	idf	0.51(27)	0.30(25)	0.38(19)
lsa	150	idf	0.51(25)	0.30(23)	0.37(23)
lsa	150	tfidf	0.44(31)	0.27(26)	0.36(25)
lsa	100	tfidf	0.40(33)	0.25(29)	0.33(27)
lsa	200	tf	0.39(34)	0.23(30)	0.29(29)
lsa	50	bin	0.37(35)	0.22(34)	0.29(30)
lsa	150	tf	0.36(36)	0.22(35)	0.28(31)
lsa	50	tfidf	0.33(38)	0.21(37)	0.30(28)
lsa	100	tf	0.33(39)	0.20(38)	0.25(35)
lsa	200	bin	0.35(37)	0.17(39)	0.22(38)
lsa	150	bin	0.31(40)	0.16(40)	0.20(41)
lsa	50	tf	0.26(44)	0.16(41)	0.22(40)
lsa	100	bin	0.29(42)	0.15(42)	0.19(42)
lsa	50	bin	0.22(45)	0.12(44)	0.16(44)

Tabuľka A2: Výsledky metódy $TFIDF$ na citačných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
TFIDF	0	tfidf_bot	**0.76(1)**	**0.49(1)**	0.56(2)
TFIDF	0	tfidf_pat	0.75(2)	0.49(2)	**0.56(1)**
TFIDF	30	tfidf_bot	0.75(3)	0.48(3)	0.54(4)
TFIDF	0	tfidf_pap	0.74(4)	0.48(4)	0.55(3)
TFIDF	30	tfidf_pat	0.74(5)	0.47(5)	0.54(5)
TFIDF	30	tfidf_pap	0.72(8)	0.46(6)	0.53(6)
TFIDF	0	idf_bot	0.73(6)	0.45(8)	0.52(7)
TFIDF	20	tfidf_bot	0.73(7)	0.46(7)	0.50(10)
TFIDF	20	tfidf_pat	0.72(9)	0.45(9)	0.51(9)
TFIDF	20	tfidf_pap	0.71(10)	0.45(10)	0.51(8)
TFIDF	30	idf_bot	0.70(11)	0.41(11)	0.45(13)
TFIDF	0	tf	0.64(16)	0.40(12)	0.48(11)
TFIDF	10	tfidf_pat	0.67(13)	0.39(13)	0.41(16)
TFIDF	30	tf	0.61(18)	0.38(15)	0.45(12)
TFIDF	10	tfidf_pap	0.66(15)	0.38(14)	0.40(17)
TFIDF	0	bin	0.63(17)	0.37(16)	0.44(14)
TFIDF	10	tfidf_bot	0.67(12)	0.36(17)	0.37(22)
TFIDF	20	tf	0.59(19)	0.35(19)	0.43(15)
TFIDF	20	idf_bot	0.66(14)	0.36(18)	0.37(21)
TFIDF	10	tf	0.53(22)	0.30(24)	0.36(26)
TFIDF	5	tfidf_pat	0.55(20)	0.26(27)	0.26(33)
TFIDF	5	tfidf_pap	0.54(21)	0.25(28)	0.25(34)
TFIDF	5	tfidf_bot	0.51(24)	0.23(32)	0.23(37)
TFIDF	30	bin	0.42(32)	0.22(33)	0.28(32)
TFIDF	5	tf	0.45(30)	0.23(31)	0.25(36)
TFIDF	10	idf_bot	0.50(28)	0.22(36)	0.22(39)
TFIDF	20	bin	0.28(43)	0.14(43)	0.18(43)
TFIDF	5	idf_bot	0.29(41)	0.10(45)	0.10(45)
TFIDF	10	bin	0.12(46)	0.06(46)	0.08(46)
TFIDF	5	bin	0.07(47)	0.04(47)	0.05(47)

Tabuľka A3: Výsledky metódy $TFIDF$ na ručne anotovaných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
TFIDF	0	tfidf_pat	**0.58(2)**	**0.50(1)**	**0.58(1)**
TFIDF	20	tfidf_pat	**0.58(2)**	0.48(2)	0.57(3)
TFIDF	5	tfidf_pat	0.56(5)	0.48(3)	0.58(2)
TFIDF	30	tfidf_pat	0.54(8)	0.47(4)	0.57(4)
TFIDF	0	tfidf_bot	0.56(5)	0.46(7)	0.56(6)
TFIDF	20	tfidf_bot	0.54(8)	0.46(5)	0.56(5)
TFIDF	30	tfidf_bot	0.56(5)	0.46(8)	0.56(7)
TFIDF	10	tfidf_pat	0.58(2)	0.45(11)	0.56(8)
TFIDF	0	tfidf_pap	0.53(12)	0.46(6)	0.56(9)
TFIDF	10	tfidf_bot	0.53(12)	0.45(9)	0.56(10)
TFIDF	30	tfidf_pap	0.53(12)	0.45(10)	0.55(11)
TFIDF	20	tfidf_pap	0.53(12)	0.44(12)	0.55(14)
TFIDF	0	idf_bot	0.49(15)	0.43(14)	0.55(13)
TFIDF	5	tfidf_pap	0.54(8)	0.43(15)	0.53(21)
TFIDF	10	tfidf_pap	0.53(12)	0.43(16)	0.53(18)
TFIDF	30	idf_bot	0.46(21)	0.43(17)	0.54(17)
TFIDF	5	tfidf_bot	0.47(17)	0.41(21)	0.53(19)
TFIDF	20	idf_bot	0.46(21)	0.41(25)	0.52(25)
TFIDF	0	tf	0.39(30)	0.40(29)	0.51(29)
TFIDF	10	idf_bot	0.46(21)	0.35(37)	0.48(35)
TFIDF	10	tf	0.34(37.5)	0.36(34)	0.48(37)
TFIDF	5	tf	0.36(34.5)	0.34(40)	0.47(38)
TFIDF	20	tf	0.34(37.5)	0.35(38)	0.47(39)
TFIDF	20	bin	0.34(37.5)	0.31(42)	0.46(42)
TFIDF	30	tf	0.27(44)	0.35(39)	0.47(41)
TFIDF	30	bin	0.25(45.5)	0.32(41)	0.45(43)
TFIDF	10	bin	0.25(45.5)	0.21(56)	0.39(50)
TFIDF	5	idf_bot	0.34(37.5)	0.19(61)	0.37(60)
TFIDF	5	bin	0.12(62)	0.15(63)	0.33(63)

Tabuľka A4: Výsledky metódy *LSA* na ručne anotovaných dátach

Metóda	k	typ váh	MAP@1 (rank)	MAP@5 (rank)	MAP@10 (rank)
lsa	100	idf	0.46(21)	**0.44(13)**	**0.55(12)**
lsa	150	idf	0.46(21)	0.42(20)	0.54(15)
lsa	150	tfidf	**0.47(17)**	0.42(19)	0.53(22)
lsa	50	idf	0.41(28)	0.43(18)	0.54(16)
lsa	200	tfidf	**0.47(17)**	0.41(24)	0.52(27)
lsa	150	bin	0.44(24)	0.41(26)	0.53(20)
lsa	200	bin	0.41(28)	0.41(22)	0.53(24)
lsa	200	idf	0.37(32)	0.41(23)	0.53(23)
lsa	100	bin	0.42(25.5)	0.40(28)	0.52(26)
lsa	100	tfidf	0.37(32)	0.41(27)	0.51(28)
lsa	200	tf	0.42(25.5)	0.37(33)	0.50(32)
lsa	50	tfidf	0.37(32)	0.40(30)	0.51(30)
lsa	150	tf	0.41(28)	0.38(32)	0.50(33)
lsa	100	tf	0.36(34.5)	0.39(31)	0.51(31)
lsa	50	bin	0.29(43)	0.36(35)	0.49(34)
lsa	50	tf	0.31(41.5)	0.35(36)	0.48(36)
lsa	50	bin	0.32(40)	0.31(43)	0.47(40)