Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

FIIT-5220-29626

Bc. Michal Kompan

PERSONALIZOVANÉ ODPORÚČANIE ZAUJÍMAVÝCH TEXTOV

Diplomová práca

Študijný program: Softvérové inžinierstvo
Študijný odbor: 9.2.5 Softvérové inžinierstvo
Miesto vypracovania: Ústav informatiky a softvérového inžinierstva. FIIT STU Bratislava
Vedúca práce: prof. Ing. Mária Bieliková, PhD.

máj 2010

ANOTÁCIA

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: SOFTVÉROVÉ INŽINIERSTVO

Autor: Bc. Michal Kompan
Diplomová práca: Personalizované odporúčanie zaujímavých textov
Vedúca diplomovej práce: prof. Ing. Mária Bieliková, PhD.
máj, 2010

Predkladaná diplomová práca sa venuje problematike personalizovaného odporúčania textov v doméne spravodajského portálu. Doména spravodajstva je charakteristická dynamickými zmenami a rapídnym znižovaním hodnoty informácií v čase. Rovnako sa problémom stáva množstvo informácií obsiahnutých v internetovom denníku. Jedným z možných riešení problému zahltenia je personalizované odporúčanie obsahu.

Práca obsahuje analýzu existujúcich prístupov a systémov, ktoré sa problematike venujú. Výsledkom práce je návrh metódy pre personalizované odporúčanie založené na obsahu. S dôrazom na hľadanie podobných článkov na základe obsahu a krátkej, ale efektívnej reprezentácie jednotlivých článkov, ktoré môžu byť využité pre odporúčanie v reálnom čase. Reprezentácia článku je založená na viaczložkovom vektore, ktorý reprezentuje článok na základe názvu, korelácie medzi nadpisom a obsahom, kategórie článku, kľúčových slov či indexu čitateľnosti. Takáto reprezentácia umožňuje aj reprezentáciu článkov, ktoré neobsahujú text, ale video, fotografie a pod. Navrhnuté riešenie sme overili v doméne spravodajského portálu SME.SK.

ANNOTATION

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: SOFTWARE ENGINEERING

Author: Bc. Michal Kompan
Thesis: Personalized recommendation of interesting articles
Supervisor: prof. Ing. Mária Bieliková, PhD.
2010, May

The main topic of this document is personalized recommendation of news articles. The domain of newscast is typical example of dynamic highly changing domain with enormous information value decreasing. Second aspect of this domain is the amount of information included in average news portal. One of the possible solutions of this problem is personalized recommendation.

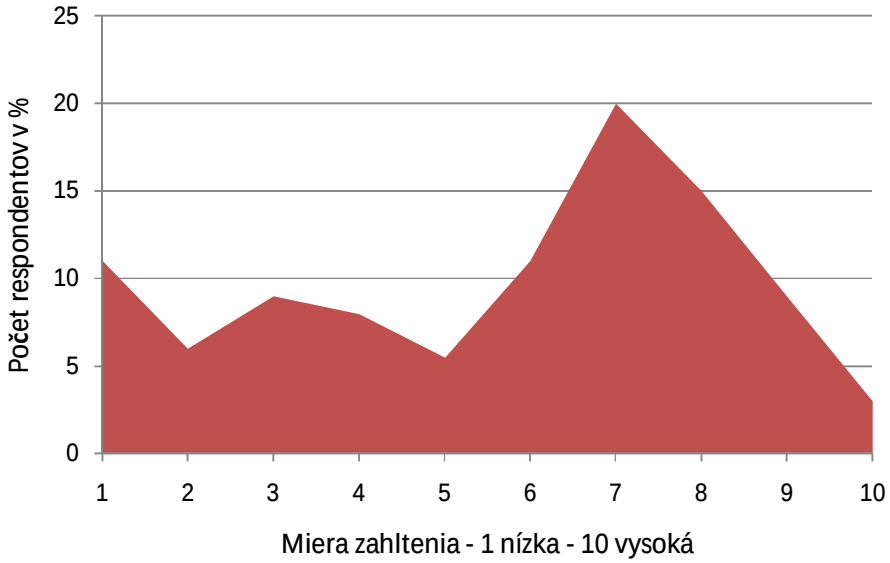
We present an analysis of existing solutions and systems, which cover this topic. The result of our work is proposal of the novel method for personalized recommendation based on the article content. We focused on fast content similarity search, and on short and fast article representation, which can be used for real-time recommendation. The article representation is based on multicompound article vector, which represents the article based on article title, correlation between article title and content, article category, several keywords or readability index. This kind of representation allows us to represent non text articles too (e.g. video or photo content). Proposed methods were verified in the domain of news portal SME.SK.

OBSAH

1 ÚVOD

Množstvo informácií obsiahnutých vo víkendovom vydaní priemerných novín je porovnateľné s množstvom informácií, ktoré získal obyčajný človek pred sto rokmi za celý svoj život. V súčasnosti je jedným z najväčších, ak nie najväčším a najprístupnejším zdrojom informácií web. Známa fráza „na webe je všetko“ sa pomaly stáva skutočnosťou. A práve z tejto, naoko pozitívnej správy, vznikajú problémy.

Množstvo dát, ktoré sa na webe nachádza, sa samo o sebe stáva závažným problémom. Veď komu sú potrebné milióny a milióny informácií, keď nevie, ktoré z nich sú relevantné. V prípade, že sa nám podarí hľadané informácie po strastiplnej ceste zadávania kľúčových slov a preklikania sa v desiatkach odkazov získať, objavia sa ďalšie problémy, ako napríklad dôveryhodnosť získaných informácií. Počet slov na stránke sa v roku 2003 takmer zdvojnásobil, rovnako neustále narastá počet odkazov, obrázkov, tabuliek, reklám a podobne. Viac ako 60% respondentov ankety IDC tvrdí, že sú zahltení informáciami viac ako polovicu času (Obr. 1).



Obr. 1 - Frekvencia zahltenia informáciami [IDC, jeseň 2008, U.S., vzorka 500 respondentov]

Pravdepodobne jedným z najpoužívanejších zdrojov informácií pre bežného používateľa sú rôzne spravodajské portály. Dôležitosť a atraktívnosť aktuálnych správ na webe dokazuje aj nespočetné množstvo príkladov, kedy sa na bežnej nespravodajskej stránke zobrazujú v skrátenej forme aktuálne správy. S postupným rozširovaním internetu a globalizáciou spoločnosti už nestačí, aby bol človek informovaný o dianí vo svojom najbližšom okolí. V dnešnej dobe je nevyhnutné získať

prehľad o dianí v širšom okolí, resp. informácie o dianí po celom svete. S týmto postupným vývojom tak neustále narastá množstvo informácií, ktoré človek spracúva. Samotné portály neustále rozširujú svoje sídla a obohacujú ich o nové informácie v snahe získať nových používateľov. Je zrejmé, že bežný používateľ nemôže spracovať všetky ponúkané informácie a musí si z dostupnej ponuky vyberať. Tým pádom sa výber správnej informácie vhodnej pre daného používateľa stáva z pohľadu daného sídla extrémne dôležitým. V doméne internetového spravodajského portálu treba zohľadniť aj fakt, že informácie dynamicky pribúdajú a pomerne rýchlo v čase „starnú“ a prestávajú byť zaujímavé.

Tým, že sa čitateľ dostane k článkom, ktoré sú pre neho zaujímavé, potrebné a dôležité, čitateľovi ušetrí tak nielen nezanedbateľné množstvo času. Zároveň zredukovaním „nepotrebných“ správ dávame čitateľovi šancu zapamätať si a spracovať informácie, ktoré môžu byť dôležité pre jeho ďalšie aktivity. Jedným z možných prístupov pre riešenie problému s množstvom informácií pri spravodajských portáloch, ktorým sa zaoberáme v tejto práci, je personalizácia, čiže upozornenie jednotlivého používateľa na vybrané správy, ktoré by si pravdepodobne pozrel a považoval ich za užitočné.

V tejto práci sa zameriame na odporúčanie založené na podobnosti článkov. Opíšeme navrhnutý model pre reprezentáciu článkov na vysoko reprezentatívnej a redukovanej úrovni, ktorý je kľúčový pre zisťovanie podobnosti veľkého množstva dynamicky sa meniacich článkov a následne aplikujeme získané výsledky do odporúčania samotných článkov, resp. textov používateľom definovaním metódy personalizovaného odporúčania. Metódu sme overili v doméne internetového spravodajského portálu.

Táto práca podáva pohľad na aktuálne dostupné riešenia v doméne, ale aj v širšom kontexte, odporúčania a personalizácie. V časti 2-Personalizované odporúčanie predkladáme prehľad bežne používaných prístupov pre riešenie personalizovaného odporúčania a ich základné delenie. Kapitola 3-Existujúce systémy pre odporúčanie stručne opisuje systémy, ktoré personalizované odporúčanie ponúkajú. Ciele práce sú formulované v rovnomennej časti 4-Ciele práce. Samotné riešenie je opísané v časti 5-Metóda zisťovania podobnosti a 6-Metóda odporúčania na základe obsahu. Doménovo závislé informácie spojené s konkrétnou realizáciou uvádzame v kapitole 7-Realizácia odporúčania pre SME.SK. Overenie navrhnutých metód predkladáme v časti 8-Overenie riešenia - experimenty. V Závere hodnotíme výsledky práce a predkladáme možné oblasti ďalšieho rozšírenia.

2 PERSONALIZOVANÉ ODPORÚČANIE

História personalizovaného odporúčania siaha do začiatku 90-tych rokov, kedy systém Xerox PARC [12] umožňoval používateľom vyhľadávať dokumenty na základe predchádzajúceho hodnotenia inými používateľmi systému. S narastajúcim množstvom informácií pribúdali aj ďalšie systémy a paradigmy personalizovaného odporúčania zastúpené spoločnosťami Bellcore, Microsoft, MIT, Amazon a pod. Všetky tieto systémy a výskum v oblasti vyústil do dvoch základných typov odporúčacích systémov. Prvou skupinou sú systémy založené na kolaborácii používateľov. Druhá skupina je založená na podobnosti odporúčaného obsahu. V praxi sa využívajú aj kombinácie týchto dvoch prístupov. Nevyhnutnosť pri odporúčaní ako takom je vedomosť o záujmoch používateľa. Na ich reprezentáciu sa používa model používateľa.

2.1. Model používateľa

Pri všetkých typoch personalizácie je základnou podmienkou, bez ktorej sa nedá nič odporučiť ani tou najlepšou metódou, poznať záujmy, resp. aktivitu používateľa. Existuje niekoľko prístupov, ako model (aktivitu) používateľa získať. Profil používateľa je hlavnou a kľúčovou zložkou modelu požívateľa, používaného pri samotnej personalizácii v navrhnutej metóde, opísanej v časti 6-Metóda odporúčania na základe podobnosti obsahu.

2.1.1. Používateľom vypĺňaný model

Bežný používateľ internetu nie je schopný a častokrát aj ochotný dostatočne dobre špecifikovať svoje požiadavky do kľúčových slov, a tým uľahčiť tvorbu personalizovaného obsahu. Tento problém sa snažia vyriešiť viaceré prístupy ako napríklad web so sémantikou [23] tým, že sa obsah obohatí o sémantickú vrstvu, ktorá umožňuje lepšie strojové spracovanie a odvodzovanie. My sa však na tento problém pozeráme z iného hľadiska.

Najjednoduchším spôsobom zistenia preferencií používateľa je požiadanie používateľa, aby vyplnil svoj profil, kde mu ponúkneme možnosti reflektujúce obsah a štruktúru poskytovaných informácií (hotmail.com, lifestream.com a iné). Takýto profil môže byť reprezentovaný ako „klasický“ vektor, kde každá zložka indikuje preferenciu danej oblasti.

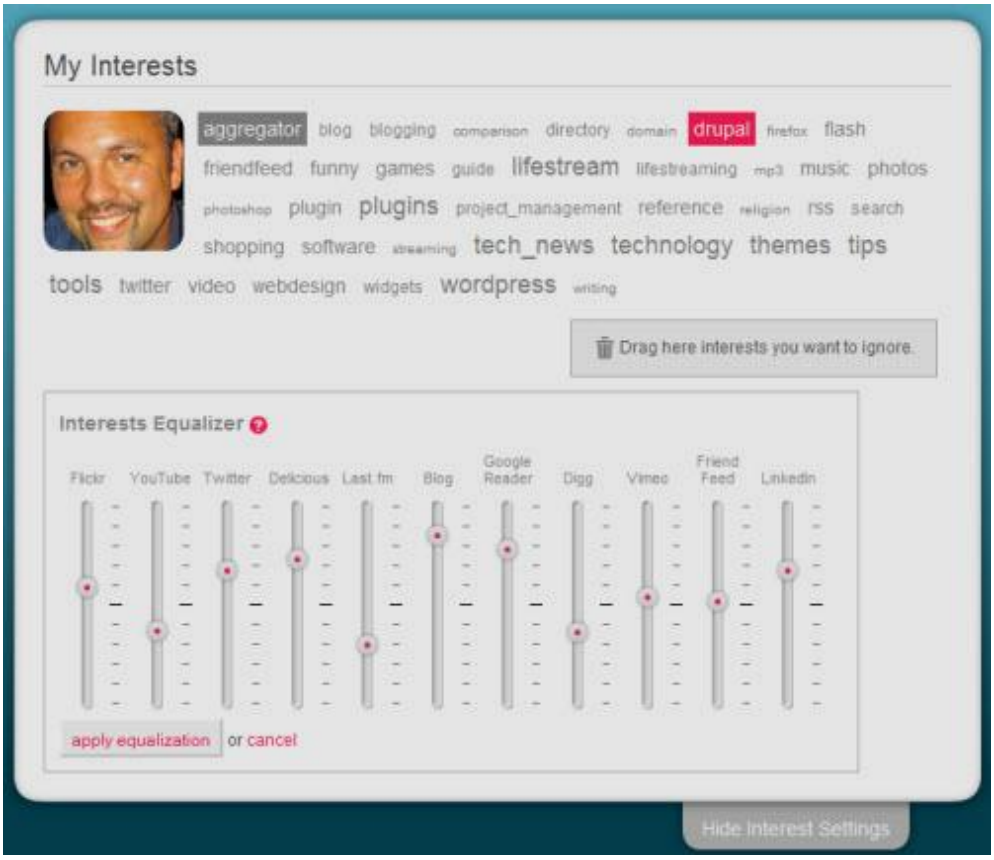
Iným spôsobom je reprezentácia používateľských preferencií pomocou množiny pravidiel. Jedná sa o najjednoduchší a najpriamočiarejší prístup k identifikácii preferencií používateľa. Avšak prináša hneď niekoľko nevýhod a problémov. V prvom rade je nevyhnutné, aby sa používateľ prihlásil na daný portál. Iným spôsobom je prihlásenie opomenúť a identifikovať používateľa len na základe dočasných záznamov prehliadača, v ktorom sa stránka zobrazuje („cookies“), avšak takéto riešenie prináša iné negatíva (vymazanie dočasných dát prehliadača, viacero používateľov na jednom PC a pod.). Ďalším problémom pri vytváraní týchto preferenčných profilov je fakt, že zaťažujú používateľa. Bežný používateľ chce pristupovať

k informáciám okamžite a vypĺňanie dlhých formulárov v profiloch ho nielen vyslovene zaťažuje, ale často sa to odrazí aj na samotnej informačnej hodnote vyplnených údajov [31, 35].

Nezanedbateľným problémom týchto profilov je fakt, že z časového hľadiska sa preferencie používateľov menia, preto je nevyhnutné tieto profily upravovať, a tým sa používateľ dostáva do začarovaného kruhu.

Preferencie sa však menia aj v oveľa kratšom časovom období ako je bežné obnovenie preferencií v profile reálne. Ako príklad môžeme uviesť majstrovstvá v ľadovom hokeji. Náš používateľ bežne správy o športe a ľadovom hokeji nesleduje. Avšak počas majstrovstiev sa zaujíma o úspechy reprezentačného tímu. Systém s personalizovaným odporúčaním by mal identifikovať tento záujem a ponúknuť používateľovi správy tohto druhu, avšak po niekoľkých zápasoch tím vypadne a používateľ sa o šport zase prestane zaujímať. Túto skutočnosť by systém mal taktiež dokázať odhaliť a športové správy eliminovať. Pri prístupe s používateľom vytváraným profilom však systém pravdepodobne takéto správanie neodhalí.

V dnešnej dobe pozorujeme snahy o vylepšenie, resp. sprehľadnenie vypĺňania týchto profilov, kedy sa jednotlivé polia nahradzujú posuvníkmi alebo inými používateľsky príťažlivejšími prvkami (Obr. 2).



Obr. 2 – Model preferencií používateľa. [lifestream.com]

2.1.2. Explicitná spätná väzba

Veľmi častým prístupom pri odporúčacích systémoch je využitie explicitnej spätnej väzby. Jedná sa o veľmi jednoduchý princíp, ktorý však vyžaduje následné dôkladné kategorizovanie dokumentov. Spätná väzba sa spravidla získava možnosťami „+“, alebo „-“, ktoré sa používateľovi zobrazia na danej stránke a jednoducho znamenajú viac alebo menej stránok podobných s aktuálne prezeranou stránkou. Inými variáciami je využitie stupnice spokojnosti s danou stránkou (grafická alebo textová). Existuje niekoľko variácií tohto prístupu. Vo všeobecnosti môžeme povedať, že sa využívajú intuitívne zväčša grafické prvky, ktorými je možné veľmi rýchlo a pomerne presne vyjadriť názor používateľa, pričom formuláre s možnosťou vyjadrenia rozsiahlejšieho názoru nie sú využívané.

2.1.3. Automatické identifikovanie modelu používateľa

Posledný typ napĺňania modelu používateľa je automatická detekcia záujmov a cieľov používateľa [30]. Toto je možné identifikovať na základe dlhšieho časového obdobia, kedy vieme určiť hlavné skupiny stránok, o ktoré ma používateľ záujem. Rovnako je možné do istej miery identifikovať stránku, na ktorej používateľ strávi netriviálny čas, kedy môžeme predpokladať, že ho obsah zaujal a číta ho.

Tu je nevyhnutné zobrať do úvahy čas, ktorý treba na prečítanie danej stránky v závislosti od dĺžky článku. Komplikáciou je rôzny multimediálny obsah, ktorý môže byť „ľubovoľne“ dlhý. Pri takomto riešení je nevyhnutné spúšťať na stránke rôzne doplnkové prvky („scripty“), ktoré zabezpečia detekciu aktivity. Existujú prístupy, kedy sa pomocou vizuálnej techniky sleduje poloha používateľových zreničiek [38], ktorá sa následne mapuje na miesto obrazovky. Tým vie systém detegovať časť, ktorú používateľ číta, resp. časť, ktorá ho zaujala a toto využiť pri odhaľovaní jeho záujmov.

2.2. Prístupy k personalizovanému odporúčaniu

Najznámejšími prístupmi k tvorbe personalizovaných odporúčaní sú odporúčania založené na obsahu a kolaborácii (sociálnych vzťahoch). Rozdiel medzi nimi spočíva v tom, že pokiaľ pri odporúčaní založenom na obsahu sa pozeráme výlučne na samotný odporúčaný obsah, pri kolaboratívnom odporúčaní sa odporúča na základe aktivity podobných používateľov. Rozšírením je využitie oboch prístupov ich kombináciou.

2.2.1. Odporúčanie založené na obsahu

Základy odporúčania založeného na obsahu sa viažu k doméne objavovania znalostí („Information Retrieval“). Tento spôsob odporúčania stavia na vzťahoch odporúčaného obsahu a preferencií používateľa (modelu používateľa) [28]. Prístup je založený na princípe, že pokiaľ vieme, aký obsah má používateľ rád, budeme mu odporúčať najrelevantnejšie objekty z daného obsahu z pohľadu vopred definovanej podobnosti objektov (text, hudba a pod.) daného obsahu. Inými slovami odporúčame len na základe informácií, ktoré môžu byť odvodené z vlastností daného obsahu.

Práve definovanie podobnosti daného obsahu a jej vyhľadávanie sú hlavným problémom tohto prístupu, kedy je odporúčanie silne zviazané s daným objektom odporúčania, čo nevidíme pri

kolaboratívnom odporúčaní. Ďalším problémom, pokiaľ sa zameriame na odporúčanie textov, resp. novinových stránok alebo akéhokoľvek obsahu reprezentovaného textom, je viacznačnosť slov a silná závislosť od jazyka. Existujú prístupy, ktoré zahŕňajú sémantiku, prípadne prekladajú daný obsah do konkrétneho jazyka, s ktorým sa následne pracuje, avšak prinášajú so sebou ďalší netriviálny problém – výpočtovú náročnosť, ktorá je kritická najmä v doménach s veľkou dynamikou ako doména spravodajského portálu.

Typický prístup, ako odporúčať a identifikovať profil používateľa, je aplikovanie strojového učenia. Jedná sa o odporúčanie založené na podobnosti medzi dátami a profilmi. Jedným z problémov pri odporúčaní založenom na obsahu je indexovanie, resp. reprezentácia modelov používateľa pre efektívne priradenie k dátam. Vzhľadom na to, že metóda je založená na „obsahu“, jej efektivita priamo závisí od toho, ako „dobre“ používateľské profily reprezentujú záujmy používateľov.

Na druhej strane je nevyhnutné mať rovnako dobre definované aj samotné stránky. Ako hlavný problém tu vidíme identifikáciu vhodných kľúčových slov, ktoré reprezentujú podstatu stránky. Pri odporúčaní niektorých tém, ako napríklad vtipov, ale aj básní, nezískame aplikovaním metód ako frekvencie výskytu slov dostatok informácií, ktoré môžu byť užitočné pri vytváraní profilu používateľa [26]. Preto sa odporúčanie založené na obsahu hodí najmä do domén, kde sú informácie štruktúrované, ako napríklad filmy, reštaurácie a pod. V týchto prípadoch je možné samotné vektory, ktorými reprezentujeme jednotlivé preferencie váhovať a tak získať lepšie výsledky [25]. Keďže odporúčanie založené na obsahu tvorí jadro práce, budeme sa mu ešte venovať v ďalších častiach.

Pri odporúčaní založenom na obsahu treba zohľadniť problém „úzkej špecializácie“ – odporúča sa podobný obsah. Jedným z možných riešení tohto problému je zavedenie náhody, kedy sa medzi odporučený obsah dostanú aj náhodné prvky, prípadne rozšírenie odporúčania o myšlienky prevzaté z evolučných algoritmov ako napr. mutácia.

Odporúčanie založené na pravidlách

Vo väčšine prípadov sa jedná o sídla „E-biznis“, ktoré na základe skupiny pravidiel používateľom odporúčajú daný obsah (napr. Amazon.com). Tieto pravidlá sú zväčša statického charakteru (napr. „Keď kúpi CD, odporuč audio zostavu.“) Sofistikovanejším prístupom je zaznamenávanie histórie používateľa (vytváranie modelu používateľa), kedy sú na základe podobnosti odporúčaného obsahu používateľovi odporúčané jednotlivé produkty (iná kniha od rovnakého autora a pod.). Tento spôsob odporúčania je však dnes už málo účinný, nakoľko používatelia vedia, aký obsah im bude odporúčaný a ako táto technika funguje [28]. Prístup môže byť využitý aj ako kolaboratívne odporúčanie v prípade, že sa pre vytvorenie pravidiel využije aktivita používateľov.

2.2.2. Kolaboratívne odporúčanie

Tradičné kolaboratívne odporúčanie je prístup, ktorý zahŕňa sociálny prvok. Používatelia sú zoskupení do skupín na základe podobných preferencií, návykov a hodnotení poskytovaného obsahu alebo aktivity. Následne sa pri odporúčaní nového preferovaného objektu (článok, film a pod.) pozrieme, ako tento „objekt“ hodnotili ostatní používatelia v skupine a odporučíme ho na základe zistených referencií o hodnotení inými používateľmi. Vďaka jednoduchosti je tento

prístup momentálne široko rozšírený a používaný, pričom silne závisí od štruktúry a veľkosti komunity, ktorá k danému sídlu pristupuje.

Tento prístup v sebe zhŕňa dve základné časti: kolaboratívne odporúčanie založené na používateľovi a kolaboratívne odporúčanie založené na predikcii hodnotenia.

1. Kolaboratívne odporúčanie založené na používateľovi

Jedná sa o tradičné kolaboratívne odporúčanie, kedy predpokladáme, že ak hodnotenia niektorých stránok sú medzi dvoma používateľmi podobné, potom aj hodnotenie ďalšej stránky bude pravdepodobne podobné. Systémy založené na kolaboratívnom odporúčaní využívajú štatistické metódy pre hľadanie najbližšieho suseda a na základe jeho hodnotenia predikujú zaujímavosť a relevantnosť nehodnotenej stránky. Proces možno rozdeliť na identifikáciu dát, hľadanie podobných používateľov a vytvorenie odporúčaní [20].

Identifikácia dát

Hodnotenia používateľov môžu byť jednoucho reprezentované maticou *A* o rozmeroch *m*n*, kde *m* reprezentuje počet používateľov, resp. jedného používateľa a *n* reprezentuje počet stránok, resp. jednu stránku. Položka R_{ik} potom znamená hodnotenie používateľa *i* a stránky *k* (Tab. 1).

Tab. 1 - Matica hodnotení stránok používateľmi [20].

Používateľ	**Stránka**				
	$Stránka_1$	...	$Stránka_k$	...	$Stránka_n$
$Používateľ_1$	$R_{1,1}$	...	$R_{1,k}$	...	$R_{1,n}$
...		...	...	...	...
$Používateľ_i$	$R_{i,1}$	...	$R_{i,k}$	...	$R_{i,n}$
...	...	...	...	...	...
$Používateľ_m$	$R_{m,1}$	...	$R_{m,k}$	...	$R_{m,n}$

Pri spomínanej reprezentácii môže vzniknúť niekoľko problémov. V prvom rade pri veľkom počte používateľov, ale hlavne pri veľkom počte stránok nastáva situácia, kedy sa väčšina položiek R_{ik} rovná nule, a tým dostávame riedku maticu. Rovnako akákoľvek operácia nad maticou je náročná, a tým ohraničuje využitie tohto prístupu. Ako príklad môžeme uviesť problém s množstvom prázdnych miest v matici hodnotení stránok používateľa, kedy väčšina článkov používateľom hodnotená nie je. Pre ilustráciu uvažujme 1D priestor a interval [0,1], kde približne 100 inštancií pokryje množinu reálnych čísel dobre, avšak v prípade 10D priestoru bude týchto 100 inštancií predstavovať izolované body [17]. Existujú však prístupy ako „slovník kľúčov“, „zoznam koordinátov“ a iné, ktoré dokážu čiastočne eliminovať tento problém [34]. Ďalším faktom je neustále sa meniaci kontext v rámci domény spravodajského portálu, s ktorým sa je nutné vysporiadať.

Vytvorenie odporúčaní

Samotná tvorba odporúčaní je špecifická pre každý model a tvorí vlastne základ úspešnosti odporúčacieho systému. Vo všeobecnosti však podobnosť článku získame po zohľadnení podobnosti používateľov, hodnotení konkrétnej položky používateľom a priemerného hodnotenia, ktoré stránkam dal [20].

2. Kolaboratívne odporúčanie založené na predikcii hodnotenia

Proces odporúčania je v princípe totožný s procesom odporúčania pri kolaboratívnom odporúčaní založenom na používateľovi. Môžeme ho rozdeliť do troch hore uvedených častí. Rozdiel a vylepšenie však spočíva v tom, že pri výpočte podobnosti používateľov sa najskôr vypočíta podobnosť stránok, vyberie sa niekoľko stránok s najvyššou podobnosťou, tým zabezpečíme hodnotenie danej stránky od viac ako jedného používateľa (za predpokladu, že používateľ hodnotí podobné stránky približne rovnako). Následne je metóda založená na rovnakom princípe ako pri metóde založenej na používateľovi, avšak rozdiel hodnotenia a priemerného hodnotenia je nahradený hodnoteniami viacerých používateľov.

Existujú rôzne vylepšenia spomínaných prístupov, kde ide o malé vylepšenia v rámci konkrétnych domén, resp. vlastností, ako napr. efektívnosť, či pri určovaní skupín používateľov pre odporúčanie.

Spomenuté dva základné princípy sa v praxi často spájajú a vytvárajú tak jednu metódu odporúčania, kedy sa odporúčanie založené na obsahu použije v prípade, že sa jedná o novú, doposiaľ nehodnotenú stránku. V opačnom prípade sa využíva kolaboratívne odporúčanie. Iným spôsobom je využitie obidvoch prístupov naraz, pri použití váhovania.

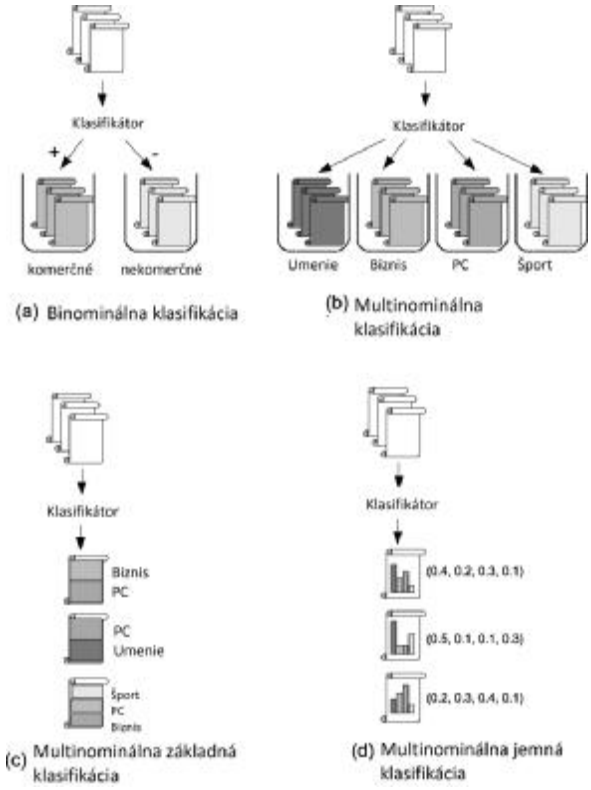
Pri personalizovanom odporúčaní je často účelné toto odporúčanie vytvárať na základe dvoch časových období – z krátkodobého hľadiska (preferencie sa menia pomerne rýchlo, ako napríklad víťazstvo na majstrovstvách sveta) a z dlhodobého hľadiska (preferencie sa menia len veľmi málo, ako napríklad šport, politika a podobne). Následne je dôležité správne identifikovať vhodný pomer medzi týmito obdobiami a vyvážiť tak obsah zaujímavý z dlhodobého hľadiska s obsahom, ktorý používateľa zaujal aktuálne. Pri hľadaní vhodnej hranice treba brať do úvahy aj možnosti získavania a uchovávania dlhodobých preferencií používateľov.

2.3. Klasifikácia webových dokumentov

Pre využitie metód odporúčania založených na obsahu je často dôležité určitým spôsobom poznať, resp. klasifikovať dáta, ktoré chceme odporúčať (v našom prípade webové stránky obsahujúce spravodajské články). V procese klasifikácie priradíme každej entite jednu triedu v rámci skupiny vzájomne sa neprekrývajúcich sa tried [14]. Ako by sa na prvý pohľad mohlo zdať, pri vyššie spomenutom kolaboratívnom odporúčaní nie je potrebné s dátami „nič“ robiť, avšak opak je pravdou. V drvivej väčšine prípadov nechceme používateľa zaťažovať vytváraním jeho profilu a táto činnosť je automatizovaná. Práve tu je teda nevyhnutné stránky spracovať a na základe výsledkov vytvoriť profily používateľov, na základe ktorých bude prebiehať kolaboratívne odporúčanie.

Klasifikáciu stránok môžeme rozdeliť do niekoľkých úloh. Prvou je klasifikácia predmetu stránky (šport, umenie a pod.), ďalej sú to problémy ako klasifikácia funkcie stránky (osobná domáca stránka, stránky kurzov a pod.) a iné [27]. V našej doméne (spravodajské portály) sa budeme zameriavať hlavne na klasifikáciu stránky vzhľadom na predmet – typ správy, ktorú nesie (napríklad správy z domova, šport, ekonomika a podobne).

Vzhľadom na počet tried, ktoré klasifikovaný objekt (stránka) nesie, môžeme klasifikáciu rozdeliť na binominálnu (Obr. 3a) a multinomiálnu (Obr. 3b). Binominálny klasifikátor klasifikuje články na „dve“ základné skupiny. Ako je zrejmé, multinominálny rozpoznáva viac kategórií článkov. Ďalšie dva prípady možnej klasifikácie, ktoré sa však pravdepodobne v našej doméne vyskytovať nebudú, je multinomiálna základná klasifikácia (Obr. 3c) a multinomiálna jemná klasifikácia (Obr. 3d).

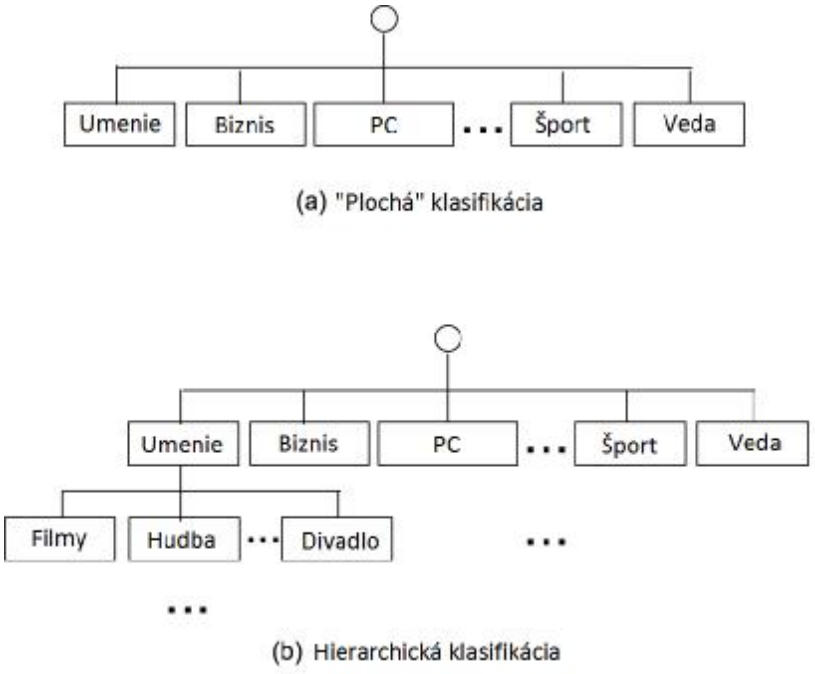


Obr. 3 - Typy klasifikácie [27].

V prvom prípade sa jedná o klasifikovanie častí dokumentov ako pri obyčajnej multinomálnej klasifikácii. Inými slovami klasifikovanú kategóriu nedostaneme pre dokument ako celok, ale pre jeho jednotlivé časti. V druhom prípade dostaneme percentuálne rozloženie kategórií, do ktorých klasifikujeme pre každý spracovávaný dokument. Spomínaný posledný typ klasifikácie by bolo možné uplatniť už v prípade rôznych komentárov, ktoré sa na spravodajských serveroch nachádzajú a na základe toho ich zaradiť do príslušnej kategórie.

Ďalšou možnosťou je delenie na základe „hĺbky“. Ide o „plochú“ alebo hierarchickú klasifikáciu. Pri plochej klasifikácii (Obr. 4a) kategórie považujeme za paralelné a rovnocenné.

Naopak pri hierarchickej (Obr. 4b) sú niektoré kategórie nadradené a vytvárajú stromovú štruktúru.



Obr. 4 - Plochá a hierarchická klasifikácia [27].

Pri klasifikácii môžeme rozlíšiť zdroj klasifikácie na webové stránky alebo text. Klasifikácia textu zväčša prebieha na štruktúrovaných textoch, zatiaľ čo klasickým webovým stránkam štruktúra ako taká chýba, resp. je tvorená samotným návrhom a navigáciou daného sídla. Webové stránky obsahujú hypertextové odkazy, HTML značky a pod., z ktorých možno získať ďalšie informácie o kategorizovaných dátach.

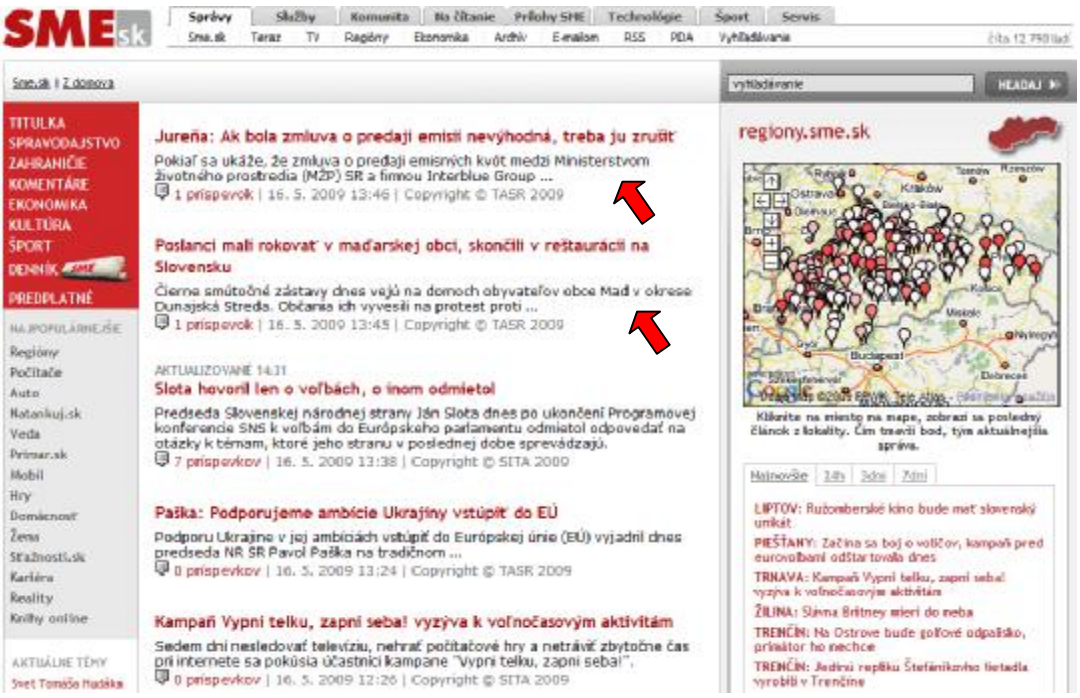
2.3.1. Textový obsah a HTML značky

Analýza a klasifikácia na základe textového obsahu je najpriamočiarejšia metóda, keďže informácie sa nachádzajú priamo na stránke. Problémom však môže byť veľké množstvo rušivých informácií, ktoré stránka môže obsahovať, a tým následné chyby klasifikácie. Preto boli preskúmané viaceré metódy ako využitie N-gramov, prípadne frekvencie výskytu slov alebo klasifikácie na základe hierarchie (Yahoo!) [27]. Výhoda prístupu N-gramov je v tom, že metóda neuvažuje len jednoduché slová, ale práve naopak v závislosti od zvoleného N generuje postupnosti slov. Napríklad jeden dokument obsahuje frázu „New York“ a druhý dokument obsahuje slová new a york. Tieto dokumenty by pomocou metódy N-gramov, kde N>1, neboli označené za podobné. Treba však podotknúť, že metóda N-gramov má tiež nedostatky. Obyčajne vygeneruje omnoho väčší priestor ako obyčajná frekvencia výskytu. Tým dostávame väčšie dimenzie, čo môže spôsobovať značný problém nielen z pohľadu efektívnosti výpočtu.

Ako sme už spomínali, webové stránky obsahujú HTML značky, ktorých zaradenie do klasifikácie môže výrazne napomôcť samotnej klasifikácii, a tým značne vylepšiť výsledky. Žiaľ,

vo väčšine prípadov sú HTML značky zamerané viac na vizuálnu stránku reprezentácie než na sémantiku, avšak pri aplikácii na konkrétnu doménu je možné využiť aj takúto „nevýhodu“. Napríklad Shanks a Williams [2001] ukázali, že využitie prvých fragmentov (Obr. 5) zo spravodajských článkov dosahuje rýchle a presné výsledky klasifikácie.

Princíp je založený na predpoklade, že dobré zhrnutie dokumentu môže kvalitne reprezentovať hlavné prvky klasifikovaného textu.



Obr. 5 - Ukážka fragmentov spravodajských článkov [www.sme.sk].

2.3.2. Vizuálna analýza

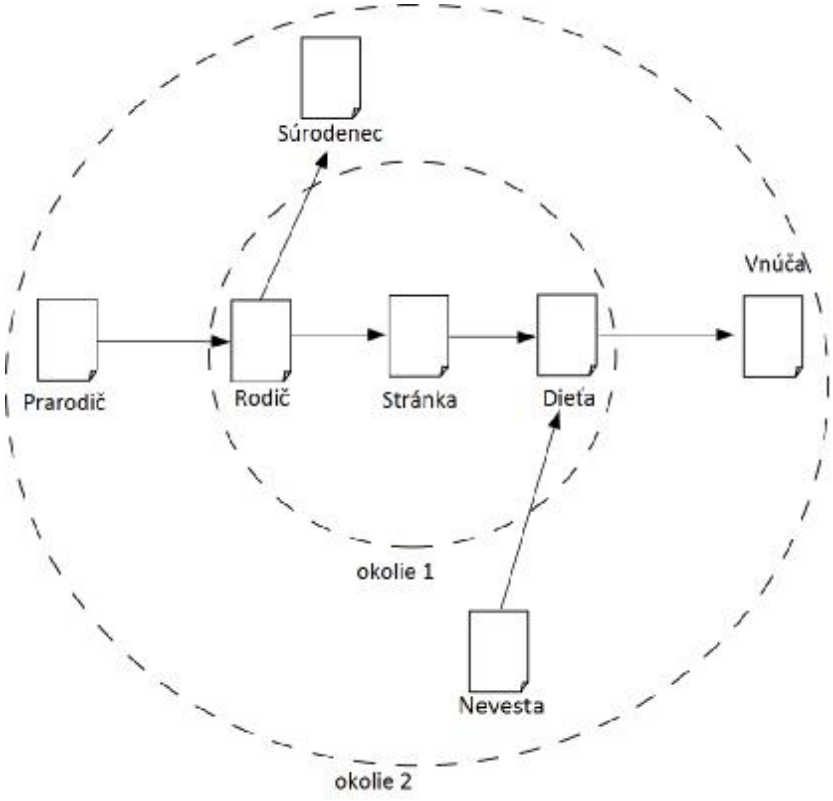
Webová stránka má väčšinou minimálne dve reprezentácie. Jedna je textová obsahujúca HTML značky a iné prvky, druhá verzia je samotná vizuálna reprezentácia – tak, ako ju vykresľuje webový prehliadač, ktorá môže byť rovnako užitočná. Keďže samotná vizuálna reprezentácia stránky je založená na HTML značkách, pričom rôzne značky môžu viesť k totožnej vizuálnej reprezentácii [27], analýza na základe tejto reprezentácie sa javí ako zaujímavá a výhodná, keďže na jej základe môže byť stránka reprezentovaná ako hierarchický multigraf, prípadne môžu byť identifikované logické časti stránok.

2.3.3. „Susedné“ stránky

Často sa stáva, že všetky požadované informácie sa nenachádzajú priamo na danej stránke, resp. stránka obsahuje málo textu a napríklad veľa obrázkov, prípadne video a podobne. Tu môže nastať problém so samotnou klasifikáciou. Ako jedno z riešení možno využiť prístup susedných stránok. Tento prístup predpokladá, že pokiaľ sú dve stránky z rovnakej kategórie a existuje stránka, ktorá je ich „sused“ v grafovej reprezentácii (na základe odkazov, obsahovej podobnosti a pod.), patrí tiež s veľkou pravdepodobnosťou do rovnakej kategórie [37]. Tento predpoklad

môžeme aplikovať rovnako na klasifikáciu predmetu stránky, ale aj na klasifikáciu funkcie [27]. V doméne klasifikácie sa používa aj zosilnené tvrdenie, kedy pravdepodobnosť prislúchania stránky do určitej kategórie narastá s počtom susediacich stránok z tejto kategórie. Toto tvrdenie však bolo dokázané pri klasifikácii predmetu stránok.

Výber susedných stránok, ktoré sa použijú pri klasifikácii, je ďalšou závažnou otázkou. Vo všeobecnosti sa využívajú stránky do hĺbky (vzdialenosti) nie väčšej ako 2. Pri tejto hĺbke dostávame 6 typov susedných stránok (Obr. 6).



Obr. 6 - Susedia stránky pri hĺbke 2 [27].

Základné štyri typy susedov (rodič, dieťa, súrodenec, nevesta) boli preštudované a bežne sa používajú. Na druhej strane využitie „prarodiča“ a „vnúčaťa“ je otázne a doposiaľ nebolo preskúmané.

Pri klasifikácii s využitím susedov sa využívajú aj tagy, čiastkový obsah ako nadpisy, ukotvenia textu, ich okolie a podobne, ako aj úplný obsah. Na druhej strane sa často v špecifických prístupoch využívajú aj tagy pridané používateľom. Problémom však je, že takéto tagy sú dostupné len pre veľmi malý okruh webu. Problematickým ostáva aj zjednotenie názvov značiek pre jednotlivé dokumenty, resp. ich akceptácia rôznymi komunitami používateľov.

Vo všeobecnosti bolo dokázané, že využívanie prístupu susedných stránok pri klasifikácii dokáže zvýšiť presnosť o približne 12% [27], avšak ide o „najnákladnejší“ prístup (z časového a výpočtového hľadiska).

2.3.4. Dolovanie v používaní webu

Dolovanie používania webu sa zakladá na dolovaní webových záznamoch – logoch. Toto dolovanie prebieha v princípe z dvoch strán [16]. Logy môžeme skúmať:

1. Zo strany servera – odkrývanie informácií o stránkach serverom, používané na zlepšenia návrhu stránok.
2. Zo strany klienta – informácie o klientovi (používateľovi) systému, ktoré sa využívajú na personalizáciu alebo zlepšenie predspracovania stránok.

Pri zovšeobecnení úlohy sa najčastejšie jedná o dolovanie sekvenčných vzorov a asociačných pravidiel. Najprv je potrebné vykonať predspracovanie dát ako v klasickej úlohe dolovania v dátach, a to čistenie, identifikácia používateľa a identifikácia „sedení" (session). Sedenie je usporiadaný zoznam stránok, ku ktorým používateľ pristupoval v „rozumnej" (minúty až hodiny) časovej postupnosti. Následne môžeme časovú postupnosť zanedbať, prípadne ich využiť na identifikáciu preferencií používateľa.

Pri tomto prístupe vznikajú problémy spojené s identifikáciou používateľa, v prípade, že jeden počítač využíva viacero používateľov (proxy server a pod.). Iným problémom je využívanie vyrovnávacej (cache) pamäte, kedy sa stránky nezobrazujú zo servera, ale z lokálnej pamäte, čím sa naruší sledovaná postupnosť, čo priamo vedie na posledný problém chýbajúcej cesty, kedy používateľ navštívi stránku A , následne stránku B, pričom priame spojenie odkazom neexistuje.

Asociačné pravidlá

Asociačné pravidlo predstavuje vzor správania. Používajú sa najmä v doméne obchodov, kde v dátach objavíme závislosti medzi výrobkami, ktoré si ľudia kupujú, napr. kúpa masla spolu s pečivom a podobne. V našej doméne môžu byť asociačné pravidlá použité na doplnenie odkazov, resp. priame odporúčanie stránok, ktoré si užívatelia prečítajú a považujú za kvalitné.

Apriori algoritmus [34] je jedným z najznámejších algoritmov pre vytváranie asociačných pravidiel a používa sa vo väčšine komerčných produktov. Algoritmus sa riadi pravidlom, že každá podmnožina veľkej množiny musí byť veľká. Inými slovami, pokiaľ o nejakej množine vieme, že je malá, nepotrebujeme z nej generovať žiadnych kandidátov – nové množiny, pretože aj tieto množiny budú malé.

Základnou myšlienkou algoritmu je vygenerovať množiny kandidátov (Tab. 2) určenej veľkosti a následne skontrolovať databázu, či sú títo kandidáti dostatočne „veľkí". Iba takíto kandidáti sú použití na generovanie kandidátov v ďalšom kroku.

Tab. 2 - Príklad tvorby asociačných pravidiel.

Sedenie	**Množina položiek**
T1	A,B,C
T2	A,C
T3	A,D,C
T4	E,A,
T5	E,D

Podpora S („support“) pre pravidlo X=>Y je percentuálne vyjadrenie transakcie v databáze, ktorá obsahuje prienik X a Y. Sila α („confidence or strength“) pre pravidlo X=>Y je pomer počtu transakcií, ktoré obsahujú prienik X a Y k počtu transakcií, ktoré obsahujú X.

Podpora nám hovorí o tom, ako často sa vyskytuje dané pravidlo v databáze. Napríklad sa pozrime na A=>C. Sila pravidla je 75 percent, čo znamená, že ¾ krát, kedy sa vyskytlo A, rovnako sa v jednom sedení vyskytlo aj C. Podpora pre B=>C je iba 20 percent, čo znamená, že toto asociačné pravidlo existuje iba v pätine sedení.

2.4. Hľadanie podobnosti dokumentov založené na analýze textu

Zisťovanie podobnosti dvoch textov je klasická úloha v oblasti vyhľadávania informácií. Rozsiahly výskum sa uskutočňuje v oblasti detekcie plagiátorstva, kde sa využívajú podobné techniky [21]. My sa budeme zaoberať podobnosťou článkov, ktoré majú isté odlišnosti v porovnaní s bežným textom.

V prvom rade sa jedná o informácie, ktoré môžu byť z povahy článku identifikované – názov, kategória, autor, dátum vloženia a podobne. Nezanedbateľným faktorom je dĺžka samotných článkov, ktoré sú vo všeobecnosti rádovo kratšie ako štandardné dokumenty.

Základné metódy pre detekciu podobnosti dvoch textov môžeme rozdeliť na dve skupiny:

- štatistické hľadanie podobnosti,
- hľadanie so zahrnutím sémantiky [37].

2.4.1. Hľadanie podobnosti založené na štatistike

Jedná sa o najrozšírenejšie metódy detekcie podobnosti dvoch textov, kedy sa text reprezentuje pomocou priestoru vektorov. Tento algebrický model reprezentuje dokument písaný v prirodzenom jazyku ako vektor ohodnotený reálnymi hodnotami. Tieto hodnoty sú väčšinou vypočítané ako TF-IDF („Term frequency – inverse document frequency“). Následne sú tieto vektory spojené a vytvárajú tak maticu dokument – slovo – hodnota. Podobnosť sa následne vypočíta jednou zo štandardných metrík (kosínusová podobnosť, Jaccard index [13], Dice koeficient [5], euklidovská vzdialenosť a pod.)

$$Jaccard\ index(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

$$Dice\ koeficient\ (A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

$$Euklidovská\ vzdialenost\ (A,B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

Problémom pri využití TF-IDF je nutnosť zahrnúť veľkú množinu slov obsiahnutých v texte pre dosiahnutie dostatočnej presnosti. Na druhej strane tento prístup ignoruje akékoľvek informácie o štruktúre, ktorú v sebe text nesie a rovnako zanedbáva akúkoľvek informáciu o sémantike [37].

Iným štatistickým prístupom je využitie textových atribútov a barycentrických koordinátov (nie sú unikátne), kedy sa vytvorí textový barycentrický model v karteziánskej sústave.

Využitie Hammingovej vzdialenosti, ktorá hovorí o počte zmien v jednotlivých reťazcoch, nie je tak rozšírené, nakoľko nie všetky rozdiely majú rovnakú dôležitosť [37].

2.4.2. Hľadanie podobnosti so zahrnutím sémantiky

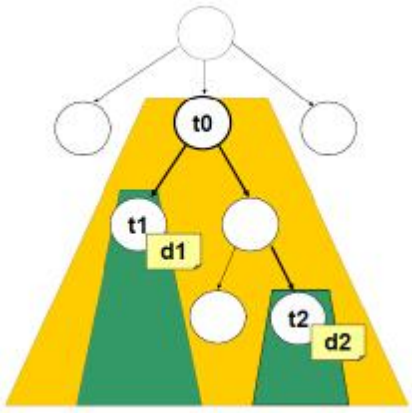
Princípy zahŕňajúce meranie podobnosti dvoch slov využívajú porovnávanie vzdialenosti významu založené na slovníkových prístupoch (pre anglický jazyk) ako WordNet, HowNet a pod. Potom v prípade dvoch slov s_1 a s_2, ktoré sú zviazané s konceptami c_{11}, $c_{12}...c_{1n}$ a c_{21}, $c_{22}...c_{2m}$ sa podobnosť vypočíta ako:

$$Podobnost\ (s_1, s_2) = \max_{i=1..n, j=1..m} Sim(c_{1i}, c_{2j})$$

Často sa za podobnosť neberú celé slová, ale používa sa metóda N-gramov. Kedy sa vezme *N* znakov textu, ktoré sú považované za ucelenú jednotku a následne sa s ňou pracuje ako v prípade slova. Potom sa nastaví posunutie v spracovávanom texte o 1 a opäť sa vezme *N* znakov atď. (pre slovo auto budú „bigramy“: „au,ut,to“) Tento prístup však nemusí byť použitý na úrovni znakov, ale aj slov, čím sa metóda stáva použiteľná aj na rozsiahlejšie texty.

Využitie sémantiky ako takej častokrát zahŕňa informácie o slovných druhoch, pozícii vo vete a pod. V tomto prípade hovoríme o metrike podobnosti viet [37]. Vtedy je text reprezentovaný vektorom 3 dimenzií – doména, situácia, pozadie.

Iný prístup pri hľadaní podobnosti so zohľadnením sémantiky je využitie stromov. Sémantická podobnosť medzi dvoma témami v určitej taxonómii je definovaná ako funkcia významu zdieľaného témami a významom každej z jednotlivých tém [22] (Obr. 7). V taxonómii je možné zistiť význam zdieľaný dvoma témami ako najnižší spoločný predok. Rozšírením tohto prístupu je zavedenie podobnosti na základe grafu, ktorá zohľadňuje nielen hierarchické, ale aj nehierarchické väzby.



Obr. 7 - Hľadanie podobnosti v stromovej reprezentácii [22].

Celkovo môžeme konštatovať, že v oblasti personalizovaného odporúčania existuje niekoľko problémov, ktoré sú predmetom aktuálneho výskumu, či už sa jedná o výpočtovú zložitosť pri odporúčaní založenom na obsahu, alebo problematickú konštrukciu modelu používateľa, prípadne problém s počtom hodnotení obsahu od používateľov pri kolaboratívnom odporúčaní. Zásadný problém pri personalizácii ako takej tvorí nový používateľ, ktorý sa v systéme predtým nepohyboval [25]. Vzhľadom na výpočtovú zložitosť a to najmä pri odporúčaní založenom na obsahu pri veľkom počte odporúčaného obsahu a dynamicky sa meniacom prostredí, ostáva personalizácia založená na obsahu v úzadí. Tento problém by sa mohol dať zmierniť efektívnou reprezentáciou obsahu.

3 EXISTUJÚCE SYSTÉMY PRE ODPORÚČANIE V DOMÉNE SPRAVODAJSTVA

V súčasnosti existuje pomerne veľa systémov pre odporúčanie. My sa zameriame na tie, ktoré realizujú personalizované odporúčanie v doméne spravodajských portálov, v rámci ktorej je nami navrhnutá metóda overená. Doména spravodajských portálov je zaujímavá z hľadiska počtu a dynamiky informácií, ktoré sú na portál pridávané. Toto so sebou prináša viaceré problémy (reprezentácia obsahu, výpočtová zložitosť a pod.), ktoré sú umocnené o rýchlosť, s ktorou tieto informácie strácajú na hodnote.

Doména internetového spravodajského portálu, resp. doména masovokomunikačných prostriedkov je vo všeobecnosti špecifická vo viacerých aspektoch. Uvedieme niektoré z nich, ktoré platia vo všeobecnosti a rovnako aj také, ktoré sú špecifické pre portál ***www.sme.sk*** (hlavný zdroj dát v tejto práci). Prvým aspektom je dynamika informácií. Správy pridávané na spravodajský portál, rovnako ako správy publikované v tlačenej podobe, dramaticky strácajú v čase svoju hodnotu. Samozrejme, že čas ako taký je v tomto prípade vysoko subjektívny, a teda jedna informácia (napr. tipy na dovolenku) bude strácať na hodnote rozdielnym tempom ako iná informácia (napr. správa o zemetrasení).

Ďalšou špecifickou črtou domény je povaha samotných informácií. Väčšina dokumentov, resp. textov obsiahnutých v spravodajskom portáli, má formu textu s rozsahom približne 150-300 slov. V obsahu môžeme nájsť aj obsah multimediálny, ako rôzne videá, fotografie a podobne, kedy je text vo väčšine prípadov obmedzený na minimum, prípadne chýba úplne. Na druhej strane počet unikátnych slov je často veľmi veľký, nakoľko sa jedná o správy z celého sveta, a teda sa v nich napr. odrážajú aj rozličné lokálne názvy a pomenovania.

Jednou zo zaujímavých vlastností spravodajského portálu je jeho návštevnosť a správanie používateľov, ktoré sa mení napríklad v závislosti od dňa v týždni.

V našej práci sa zameriame na jeden konkrétny portál, ktorý má tiež svoje špecifiká, avšak navrhované postupy sú použiteľné aj na iné spravodajské portály so zohľadnením jazykovo-závislých častí metódy. Na ilustráciu vyššie uvedených čŕt uvádzame niekoľko údajov vypovedajúcich o návštevnosti a správaní používateľov spravodajského portálu SME.SK (Tab. 3).

Tab. 3 - Štatistiky návštevnosti spravodajského portálu www.sme.sk[1].

	Počet užívateľov (cookies)	Počet návštev	Priemerná dĺžka návštevy	Priemerný čas strávený používateľom
Deň (Po)	272 913	2 457 307	8min 06s	16min 34s
Deň (So)	176 050	287 583	7min 53s	15min 44s
Týždeň (45)	993 369	2 959 648	7min 56s	37min 7s
Mesiac (8)	2 425 112	10 943 203	8min 9s	1h 20min 33s
Mesiac (10)	2 786 366	12 539 590	8min 9s	1h 22min 30s

Ako je zrejmé z vyššie uvedených údajov, portál má rôznu návštevnosť nielen na základe dňa v týždni, ale aj z dlhšieho časového hľadiska, kedy je návštevnosť počas prázdninového mesiaca nižšia (2007-2010). Rovnako je zrejmé, že čas strávený používateľom pri jednej návšteve je približne polovica z času, ktorý strávi na portáli za jeden deň, čo naznačuje, že priemerný používateľ sa na portál vráti približne 2x za jeden deň. Dĺžka jednej takejto návštevy je približne 8 minút.

Je zrejmé, že za tento čas používateľ nemá možnosť hlbšie prechádzať štruktúru sídla a hľadať konkrétne články, ale skôr uprednostní články zobrazené na titulnej stránke portálu. Keďže sa na titulnej stránke zobrazuje maximálne približne 20 článkov, môžeme nadnesene povedať, že v najlepšom prípade má čitateľ možnosť vidieť 40 rôznych nefiltrovaných článkov (pri 2 návštevách denne).

Takéto správanie bude pravdepodobne badateľné aj pre ostatné spravodajské portály, keďže sa ich štruktúra vo všeobecnosti podobá. Vo väčšine prípadov vieme ku každému článku priradiť konkrétneho autora, názov a dátum [33]. Niektoré portály[2] ku obsahu článku ešte rozlišujú aj mená osôb, ktoré sa v ňom nachádzajú, miesta, organizácie, dátumy udalostí a podobne. Rovnako nezriedkavý je aj zoznam kľúčových slov pre daný článok zväčša na rôznom stupni abstrakcie pre rôzne portály.

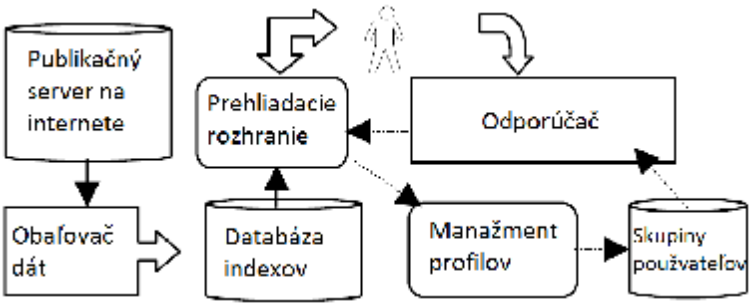
3.1. OTS

Cieľ systému OTS [36] je poskytovať personalizované vyhľadávanie na internete. Systém je navrhnutý tak, aby poskytoval odporúčanie založené na obsahu, ale aj kolaboratívne odporúčanie. Pre jednoduchosť však vyhľadávanie neprebieha nad webom ako takým, ale nad publikačným serverom. Obr. 8 znázorňuje architektúru systému. Pred samotným spustením systému sa inicializuje obaľovač dát, ktorý zozbiera bibliografické dáta z publikačného servera a vytvorí sa databáza indexov. Následne prehliadacie rozhranie pomáha používateľovi prezerať databázu podľa kategórií. V prípade, že používateľ stiahne daný článok, systém pridá informácie do profilu používateľa, kde sa od nej odvodia preferencie a správanie používateľa.

Používatelia sú rozdelení do kategórií na základe preddefinovaných typov a klasifikačnej metódy používanej v OTS. Vždy, keď používateľ požiada o odporučenie článku, systém vyberie kandidátov, ktorí sa najviac podobajú aktuálnemu profilu. Systém si udržuje zoznam prečítaných článkov, a tak neodporúča používateľovi prečítať článok, ktorý už raz videl (Obr. 8).

[1] Zdroj www.aimmonitor.sk – Asociácia internetových médií.

[2] reuters.com, nytimes.com



Obr. 8 - Architektúra systému OTS [36].

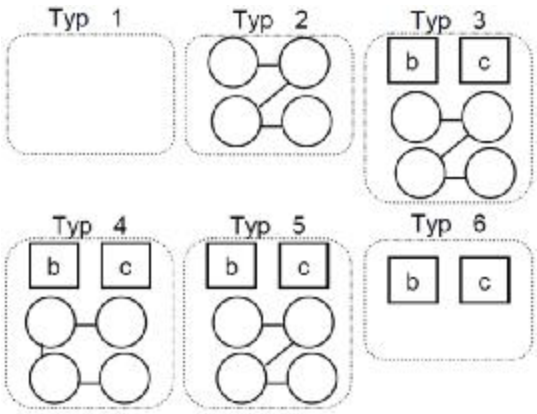
Systém pracuje na princípe asociačných pravidiel. Jadrom systému je takzvaná tabuľka záujmov, ktorá je vytvorená pre každého používateľa (Tab. 44) a následne sa podpora („Support“) vypočíta podľa vzorca:

$$podpora(c) = \frac{Počet(c)}{T - Prvý(c) + 1}$$

Tab. 4 – Príklad tabuľky záujmov v systéme OTS [36].

Transakcie	Kategória	Prvý	Posledný	Počet	Podpora
T1 {a, c, e}	A	T1	T1	1	-
T2 {b, c, e, f}	B	T2	T4	2	67%
T3 {d, e, f}	C	T1	T4	3	75%
T4 {b, c, d}	D	T3	T4	2	100%
	E	T1	T3	3	75%
	F	T2	T3	2	67%

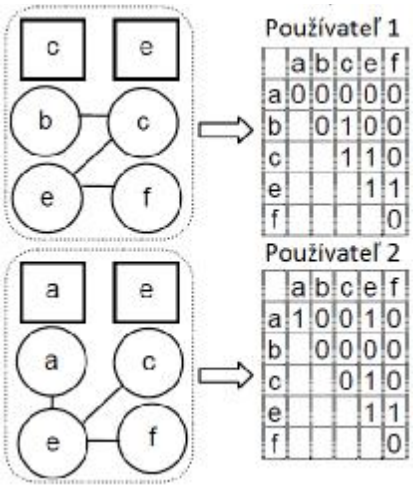
Systém uvažuje aj časové hľadisko, kedy kategórie navštívené v blízkej minulosti majú väčšiu váhu ako kategórie navštívené dávno. Systém pracuje pri jednom používateľovi s dvoma profilmi – profilom správania a profilom záujmov. Na základe týchto dvoch skupín je odvodených 6 typov profilov používateľov (Obr. 9).



Obr. 9 - Typy používateľských profilov v systéme OTS [36].

Typ 1 predstavuje používateľov, ktorí čítajú články bez žiadnej pravidelnosti alebo prvýkrát. Typ 2 zahŕňa používateľov, ktorí nepreferujú žiadnu kategóriu, ale prezerajú si články v rovnakom poradí. Typ 3 sú používatelia, ktorí sa zaujímajú o špecifické kategórie (b,c) a obyčajne si prezerajú články v rovnakom poradí, avšak občas si prezrú aj články z iných kategórií. Typ 4 je podobný s typom 3, ale používatelia v aktuálnej transakcii neprezerajú všetky obľúbené kategórie. Typ 5 je opäť podobný typu 3 a 4 s tým rozdielom, že žiadna kategória z profilu správania sa nezhoduje s kategóriou v profile záujmov. Posledný typ 6 predstavuje používateľov, ktorí preferujú jednoznačné kategórie, avšak neexistuje postupnosť, s ktorou si články prezerajú.

Na meranie vzdialeností medzi dvoma používateľmi sa používa euklidovská vzdialenosť dvoch vektorov, ktoré sa získajú „sčítaním“ vektorov z vyššie spomínaných typov (Obr. 10).



Obr. 10 - Matica "Záujem - Správanie" dvoch používateľov [36].

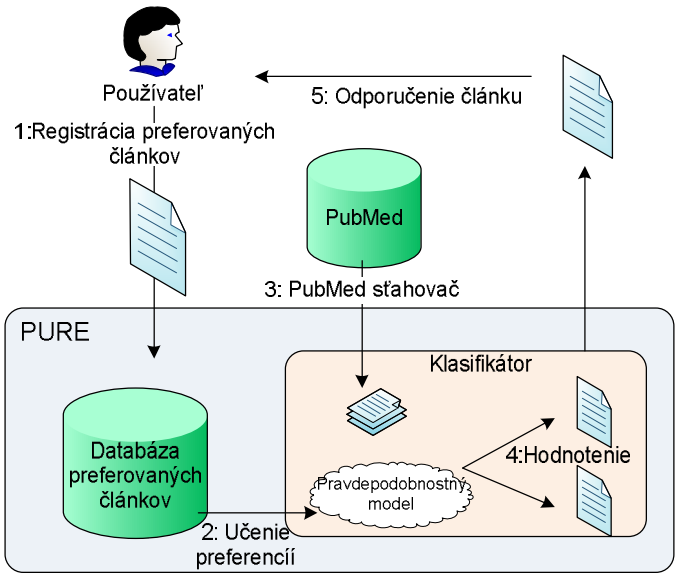
3.2. PURE

Systém PURE [39] je systém pre odporúčanie článkov z domény medicíny na základe obsahu. Systém ma webové rozhranie, pomocou ktorého používatelia pridávajú, prípadne mažú ich preferované články. V prípade, že sú články registrované, systém PURE spustí klasifikáciu založenú na modeli preferovaných článkov a odporučí články s najvyšším skóre, ktoré získali v predikcii prostredníctvom e-mailu spravidla raz za deň (Obr. 11).

Systém pracuje nad databázou PureMed, ktorá je jedna z najväčších verejných databáz v doméne biológie a medicíny. Denne tu pribúda približne tisíc nových príspevkov. Systém je založený na klient-server architektúre.

Samotný proces odporúčania pozostáva z piatich krokov :

1. *Používateľ si registruje obľúbené články.*
   Pre registráciu nových článkov existujú dva rôzne spôsoby. Prvým je, že používateľ priamo do systému zadá identifikačné číslo článku v databáze PubMed. Prostredníctvom tohto istého rozhrania je možné registrované články aj vymazať. Druhou možnosťou je výber článkov z množiny odporúčaných článkov ku konkrétnemu dňu. Článok, pokiaľ ešte nebol v systéme načítaný, sa následne stiahne z databázy PubMed.



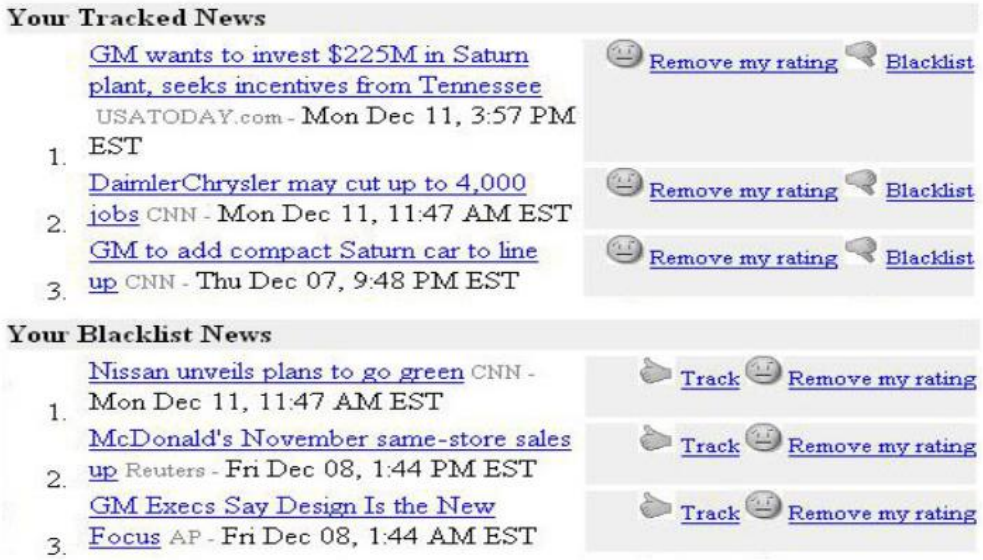
Obr. 11 - Odporúčacia schéma systému PURE [39].

2. *Natrénovanie pravdepodobnostného modelu.*
   Natrénovanie je rozdelené do dvoch krokov, ktoré sa vykonajú v nasledujúcom poradí. Najskôr sa vyberú slová a priradia sa počiatočné hodnoty. Keďže systém pracuje na báze odporúčania založeného na obsahu, je nutné vytvoriť zoznam kľúčových slov pre daný článok. Na túto činnosť sú využité klasické vyššie spomínané metódy frekvencie výskytu slov v spojení s inverznou frekvenciou, s využitím odstránenia stop slov.
   Druhým krokom je samotné naučenie pravdepodobnostného modelu pomocou algoritmu EM (Expectation – Maximization) [7], ktorý opakuje E a M kroky pokiaľ nie je splnená finálna podmienka.
3. *Denné sťahovanie nových článkov z databázy PubMed.*
   Články sa sťahujú vo formáte MEDLINE v nočných hodinách. Ukladané sú v databáze MySQL spolu s dodatočnými informáciami ako ID používateľa, dátumu registrácie a pod.
4. *Predikcia vykonaná na základe natrénovaného modelu.*
   Stiahnuté články sú ohodnotené na základe pravdepodobnostného modelu. Pre každý článok a identifikované kľúčové slová sú následne vypočítané pravdepodobnosti s použitím Z-score normalizácie.
5. *Prezentácia najlepšie hodnotených článkov používateľovi.*
   Výsledky sú používateľom prezentované prostredníctvom webového rozhrania, prípadne sú používateľom zasielané v podobe e-mailu. Počet odporúčaných článkov si používateľ nastavuje v svojom profile.

3.3. NewsMe

Systém NewsMe [35] je adaptívny systém odporúčania článkov, založený na otvorenom modeli používateľa, ktorý umožňuje používateľom robiť zásahy do svojich profilov. Systém je realizovaný pomocou webového rozhrania. Používatelia systému poskytujú spätnú väzbu prostredníctvom tlačidiel „+“, „-“. Systém preberá správy z 81 RSS kanálov od 21 rôznych

zdrojov. Správy sú na základe obsahu kategorizované do 8 kategórií, ktoré sú prezentované používateľom (Obr. 12).



Obr. 12 - Systém NewsMe [35].

Systém si uchováva samostatný model používateľa pre každú z týchto kategórií, čím sa zabraňuje miešaniu preferencií používateľa pri rôznych oblastiach záujmu. Personalizácia je v systéme transparentná – to znamená, že po prečítaní používateľ pridá článok do zoznamu obľúbených článkov alebo do zoznamu blokovaných článkov. V prípade, že sa k článku nevyjadrí, predpokladá sa, že nemá na článok nijaký špecifický názor.

Ako sme už spomínali, systém umožňuje používateľom manipulovať so svojim profilom. Po zvolení tejto možnosti sa používateľovi zobrazí zoznam obľúbených a zoznam blokovaných článkov. Používateľ môže následne ľubovoľne zoznamy editovať, presúvať jednotlivé články, mazať alebo pridávať nové články.

Samotné učenie systému prebieha na základe metódy „Najbližší Sused“ (Nearest Neighbor – NN). Model konvertuje články pomocou frekvencie výskytu slova a následne hľadá podobné články pomocou NN, alebo k-N (k najbližších susedov) metódy. Rovnako systém narába aj s dvoma hranicami – *min*, kedy sa články považujú za veľmi vzdialené a *max*, kedy naopak články sú považované za príliš podobné a neodporučia sa.

Ako vyplýva z výsledkov testovania uskutočneného na systéme, explicitná spätná väzba neprináša výrazné zlepšenie oproti explicitnej spätnej väzbe. Tento záver si môžeme vysvetliť tak, že používatelia jednoducho nečítajú a ani neklikajú články, ktoré ich nezaujímajú. Druhý a rovnako zaujímavý záver je, že systém mal porovnateľné výsledky pri žiadnych zmenách profilov používateľmi a s malým počtom zmien. Naproti tomu v prípade viacerých zmien v používateľskom profile sa výkonnosť systému značne znížila.

3.4. NewsBrief

Jedným z veľmi zaujímavých systémov nie tak pre personalizované odporúčanie, ako pre hľadanie podobných článkov, je systém NewsBrief[3]. Nepodarilo sa získať bližšie informácie o tom, ako systém konkrétne pracuje, ale z hľadiska jeho funkcionality sa nám zdal ako veľmi zaujímavý, a preto bližšie opíšeme niektoré jeho funkcie.

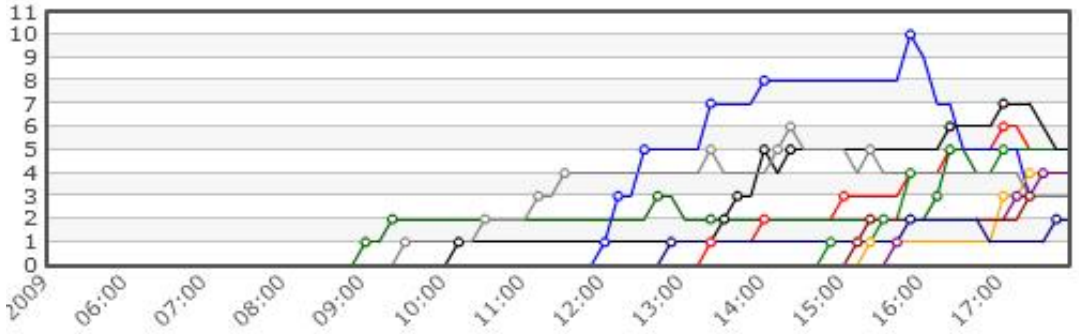
Jednou z najzaujímavejších čŕt systému je vytváranie zhlukov článkov. Systém sa automaticky snaží detegovať témy udalostí z jednotlivých jazykov (viacero spravodajských portálov v rámci jedného jazyka). Následne tak pre konkrétny jazyk systém zobrazí aktuálny zoznam tém, o ktorých sa v danom jazyku najviac píše. Systém momentálne pokrýva 27 jazykov vrátane slovenčiny. Každá téma spravidla neobsahuje viac ako 6 konkrétnych článkov (Obr. 13).

Obr. 13 - Ukážka vytvorených zhlukov článkov [www.newsbrief.eu].

Úspešnosť zhlukovacieho algoritmu je rôzna, zatiaľ čo pri článku „SDKÚ-DS chce vidieť zmluvu o nákupe vakcín" je všetkých 6 článkov v zhluku relevantných. Pri inom článku sú relevantné len 2 z celej skupiny.

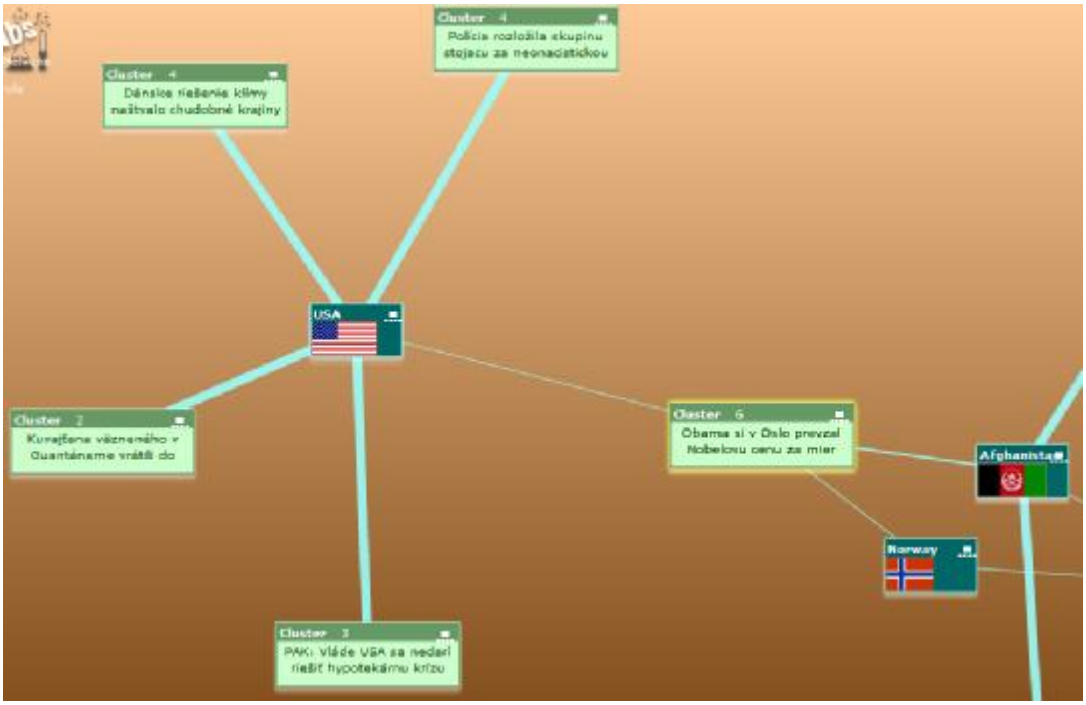
Inou pomerne zaujímavou funkcionalitou je vizualizácia detegovaných tém a ich intenzity (Obr. 14).

[3] www.newsbrief.eu

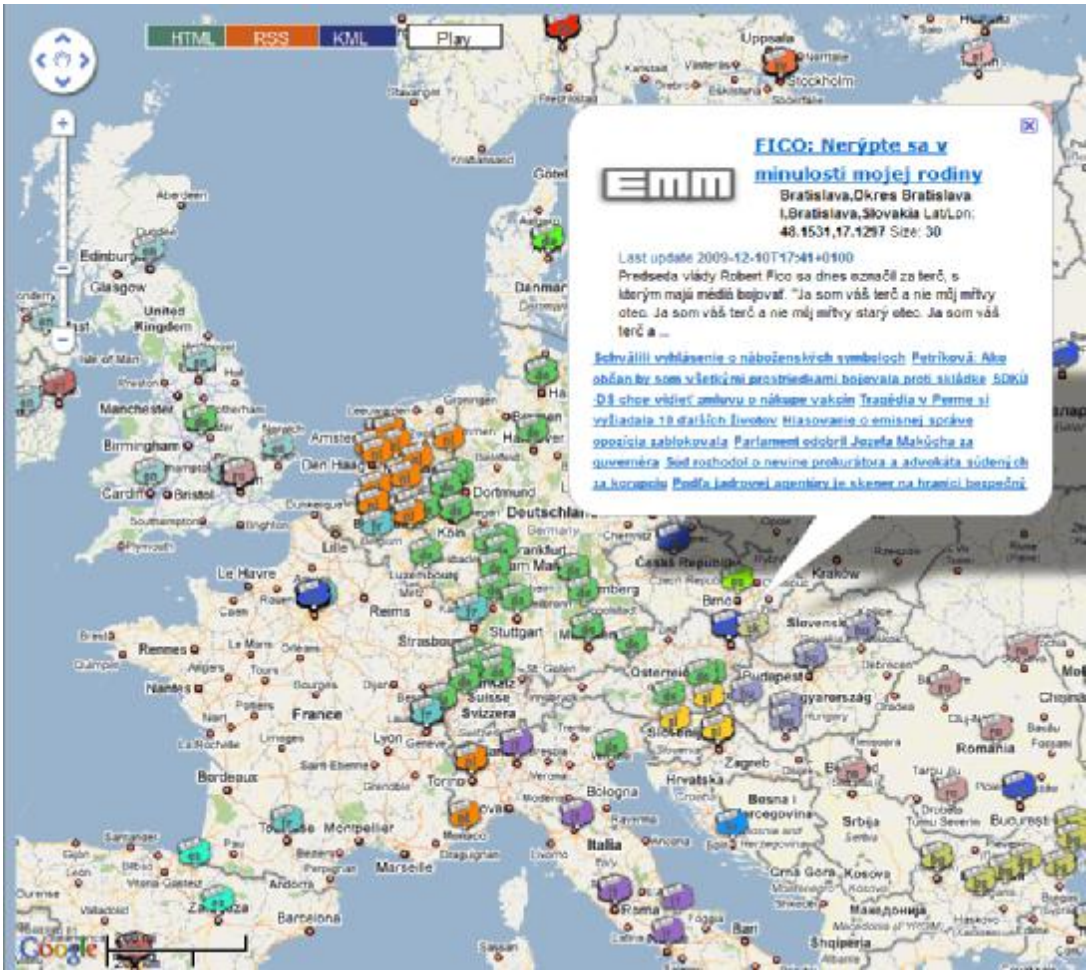


Obr. 14 - Vizualizácia tém a ich intenzity [www.newsbrief.com].

Systém podporuje odobrenie správ prostredníctvom kanálu RSS, umožňuje zasielať správy na e-mail. Medzi ďalšie zaujímavé vizualizačné možnosti systému patrí grafická reprezentácia zhlukov a ich prepojení (Obr. 15). Systém takisto poskytuje možnosť identifikovať správy na mape sveta, pričom je primárne zobrazený jazyk, v ktorom je správa napísaná (Obr. 16).



Obr. 15 - Ukážka vizualizácie vzťahov medzi jednotlivými zhlukmi [www.newsbrief.com].



Obr. 16 - Geografická reprezentácia jednotlivých správ [www.newsbrief.com].

Personalizácia ako taká zatiaľ v systéme nie je na vysokej úrovni, zakladá sa len na jednoduchých nastaveniach používateľa, kedy si vyberie, aké správy mu majú byť posielané na e-mail, prípadne prostredníctvom kanála RSS. Každopádne sa jedná o zaujímavý systém, ktorý sa neustále rozširuje a vyvíja.

Opísané systémy pre personalizované odporúčanie v doméne spravodajstva takmer vôbec neriešia otázku úspornej reprezentácie odporúčaného obsahu, a tým zefektívnenia celého procesu, čo by v ďalšom viedlo k možnosti vytvorenia personalizovaného odporúčania obsahu na základe špecifických preferencií používateľa v reálnom čase (výpočet samotnej podobnosti by prebiehal v reálnom čase). Pritom nepracujú v reálnom čase a podobnosť realizujú „off-line“. Rovnako sa tieto riešenia nevenujú problematike slovenského jazyka, kedy sa vo všeobecnosti nedajú využiť známe prístupy na detegovanie kľúčových slov, sémantiky či predspracovania, bežne používané vo svetových jazykoch.

Taktiež vyššie opísané systémy nedokázali pracovať bez explicitnej spätnej väzby, kedy používatelia museli pridávať obľúbené alebo naopak nežiaduce články do svojich profilov (modelov).

Na druhej strane tieto systémy dokázali pomerne úspešne naplniť očakávaný cieľ, a to odporúčať prehliadaný obsah, čo používateľom do určitej miery uľahčilo prácu, či už v prostredí rozsiahlej knižnice PubMed, alebo v prostredí novinových článkov.

4 CIELE PRÁCE

Personalizované odporúčanie umožňuje získavanie špecifických informácií z vysoko dynamického prostredia rýchlo a efektívne. Mení pomer času stráveného používateľom na danom spravodajskom portáli v prospech „čítania“ informácie namiesto jej hľadania. Jednou z možností personalizovaného odporúčania je odporúčanie založené na odporúčanom obsahu. Tento prístup priamo závisí od kvality a rýchlosti spracovania a reprezentácie daného obsahu, ktoré sa priamo odzrkadľujú v samotnej výkonnosti a presnosti celého odporúčania.

Jednou z hlavných úloh pri odporúčaní založenom na obsahu je hľadanie podobnosti dvoch zdrojov. V doméne spravodajského portálu sa často táto úloha presúva na autorov obsahu, ktorí po napísaní príspevku manuálne prehliadajú a označujú podobné zdroje. Je dôležité tento proces automatizovať – urýchliť a spresniť.

Vyhodnotenie metód je v doméne odporúčania netriviálny problém. Viacero prístupov pracuje bez nutnosti priameho zapojenia používateľov. Tieto však nevypovedajú o kvalite metódy tak, ako pri reálnom použití a odporúčaní konkrétnym používateľom. Nevyhnutnosť zabezpečiť dostatočnú vzorku používateľov na dostatočne dlhý čas je tiež značným problémom.

Ako sme spomínali v úvode, na účel personalizácie sa používajú dva základné prístupy. Avšak výskum napovedá, že najlepšie výsledky možno dosiahnuť skĺbením týchto dvoch prístupov. V tejto práci sa zaoberáme odporúčaním založenom na obsahu, ktoré je možné následne integrovať do iných (napr. kombinovaných) riešení.

Cieľom práce je :

- Navrhnúť metódu personalizovaného odporúčania v doméne spravodajstva založeného na obsahu v reálnom čase. Tu bude vhodné:
    - zohľadniť aktivitu používateľov
    - využiť sémantiku prehliadaných informačných zdrojov
    - ktorá bude:
        - minimalizovať výpočtovú náročnosť procesu odporúčania a hľadania podobných zdrojov
        - minimalizovať nutnosť zapojenia používateľa do zberu dát pri vytváraní modelu používateľa
- Navrhnutú metódu a odporúčania overiť v doméne spravodajského portálu (SME.SK)

Ako základný prístup pre odporúčanie využijeme zisťovanie podobnosti. Vzhľadom na rozsah domény a dynamické zmeny je nevyhnutné uvažovať o efektívnosti zisťovania podobnosti, čo predpokladá návrh úspornej reprezentácie a efektívneho porovnávania.

5 METÓDA ZISŤOVANIA PODOBNOSTI ČLÁNKOV

Podobnosť článkov je z pohľadu odporúčania založeného na obsahu jednou z kľúčových zložiek. Proces hľadania podobných článkov sme rozdelili do štyroch základných krokov:

1. Extrakcia dát
2. Predspracovanie článkov
3. Reprezentácie článkov
4. Zistenie podobnosti

Extrakcia dát a predspracovanie článkov značne závisí od konkrétnej domény a informačných zdrojov jazyka. Podrobnejšie ich opíšeme v kapitole 7-Realizácia metód pre SME.SK.

5.1. Reprezentácia článkov

Pre výpočet podobnosti je nevyhnutné reprezentovať každý spracovávaný text tak, aby boli operácie v procese výpočtu podobnosti čo najefektívnejšie. Jednou zo štandardných a široko používaných reprezentácií je vektorová reprezentácia textu [20, 9, 3]. Jednotlivé zložky vektora predstavujú samotné slová z reprezentovaného článku. Problém nastáva, keď dané články po predspracovaní stále obsahujú rádovo stovky až tisíce slov a počet dokumentov, ktoré majú byť spracované, sa pohybuje v tisícoch.

Väčšina systémov založených na odporúčaní obsahu preto nepracuje v reálnom čase. Našou snahou je zaviesť takú reprezentáciu, ktorá bude čo možno najefektívnejšia a pritom si zachová reprezentatívnu výpovednú hodnotu o danom článku. S cieľom efektívnej reprezentácie sme navrhli vektor konštruovaný ku každému článku, ktorý pozostáva zo šiestich zložiek (Tab. 3).

Tab. 3 - Vektor článku.

Názov článku	TF slov z názvu v obsahu	Kategória článku	Mená, Názvy	Kľúčové slová	Index čitateľnosti

Jednotlivé slová v každej časti vektora sú reprezentované ako množina slov („Bag of words“), čo znamená, že nezáleží na poradí, v akom sa nachádzajú.

Názov článku

Obsahuje predspracované slová, ktoré sa nachádzajú v názve článku. Váha jednotlivých slov sa vypočíta na základe frekvencie výskytu slova v názve:

$$tf_i = \frac{n_i}{\sum_k n_k},$$

kde sa frekvencia daného slova tf_i vypočíta ako podiel počtu výskytov daného slova n_i a počtu všetkých slov v dokumente .

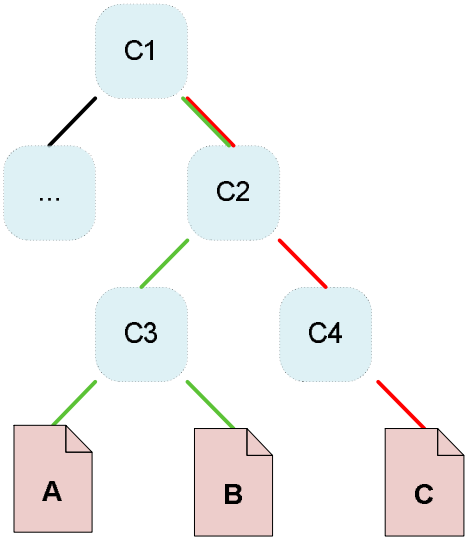
TF slov z názvu v obsahu

Keďže primárnou zložkou vektora je názov článku, musíme zobrať do úvahy, že nie všetky články zodpovedajú svojím obsahom nadpisu. Preto sme zaviedli druhú časť vektora, ktorá podobne ako prvá časť obsahuje len slová z názvu, avšak ich frekvencia výskytu („Term Frequency“) sa vypočíta na základe výskytu v obsahu článku. Tak sa nám podarí vyfiltrovať tie články, v ktorých sa slová z nadpisu nenachádzajú aj v tele textu. Pri článkoch musíme uvažovať aj to, že vo všeobecnosti článok na spravodajskom portáli nemusí obsahovať žiadny text, ale napríklad video. Keďže ide o pomerne malé percento výskytu, predpokladáme, že v takomto prípade názov článku zodpovedá aj obsahu daného článku.

Kategória článku

Vo väčšine prípadov je každý článok na spravodajskom portáli zaradený do určitej kategórie. Už samotné priradenie do kategórie vypovedá do istej miery o podobnosti dvoch článkov. Preto sme *kategóriu* článku zaradili ako jednu z častí reprezentatívneho vektora (v prípade, že portál takúto informáciu obsahuje).

Kategórie nie sú lineárne, preto sa pre každý článok vytvorí z hľadiska „hierarchie“ váha pre každú kategóriu a podkategóriu konkrétneho článku. Majme tri články *A, B, C*. Každý z týchto článkov patrí do podkategórie *C1,C2,C3,C4*. Každá z týchto kategórií je sama o sebe ešte podkategóriu inej kategórie (Obr. 17).



Obr. 17 - Hierarchia kategórií.

Následne sú váhy priradené jednotlivým zložkám tejto časti vektora vypočítané takto:

```
n=1
For i=|Kategórie| downto 0 do
váha_i=1/n
   n=n*2
end
```

Takýmto spôsobom označíme články z rovnakej kategórie za podobnejšie ako články z rôznych kategórií. Vyššie opísaným spôsobom teda dostaneme ohodnotenie jednotlivých subkategórií a kategórií pre konkrétny článok (Tab. 4).

Tab. 4 - Vektorová reprezentácia kategórií článku.

	C1	**C2**	**C3**	**C4**
A	1/4	1/2	1	-
B	1/4	1/2	1	-
C	1/4	1/2	-	1

Mená, Názvy

V článkoch sa väčšinou vyskytujú mená osôb, názvy miest alebo organizácií. Keďže ide o jazykovo závislé slová, slovníkové metódy na ich identifikáciu sú vo väčšine prípadov neúspešné. Na identifikáciu Mien a Názvov využívame pomerne jednoduchý prístup.

Pri predspracovaní odstraňujeme interpunkčné znamienka s výnimkou bodky. Za meno alebo názov v našom prístupe označujeme slovo, ktoré začína veľkým začiatočným písmenom a pred ním sa nenachádza bodka. Takáto identifikácia dokáže odhaliť približne 85-95% názvov v článkoch, nakoľko sa názvy vyskytujú ako prvé slovo vety len minimálne (napr. prvé slovo článku – geografická lokalita).

Kľúčové slová

Sú najpoužívanejším charakteristickým znakom pre reprezentáciu článku [18]. V našej metóde používame statický zoznam kľúčových slov, ktorý bol vytvorený na základe metódy TF-IDF a identifikácie podstatných mien (slovnik.juls.savba.sk). Následne si pre každý článok zistíme *n* najrelevantnejších kľúčových slov, ktoré sú pridané do vektora článku. Presný počet kľúčových slov, ktoré sú potrebné pre reprezentáciu daného informačného zdroja určíme na základe experimentálneho overenia s ohľadom na výpočtovú náročnosť.

Existujú viaceré prístupy pre identifikáciu kľúčových slov. My sme sa pre jednoduchosť rozhodli použiť vyššie opísaný prístup, ktorý je výhodný z časového hľadiska (niekedy na úkor presnosti).

Index čitateľnosti

Táto časť vektora nie je až tak významná z pohľadu sémantickej stránky podobnosti dvoch textov, ale je postavená na predpoklade, že používateľ môže mať „rád“ články s podobnou úrovňou čitateľnosti. Index čitateľnosti vo všeobecnosti vypovedá o náročnosti čítania daného textu, resp. o potrebnom vzdelaní na porozumenie textu. Preto zaradenie tejto zložky, ktorá zväčší dĺžku vektora článku len o 1, prispeje k pre usporiadaniu výsledkov, ktoré by boli získané bez neho. Naša metóda využíva Coleman – Liau index čitateľnosti [6], ktorý sa vypočíta ako:

$$CLI = 5.89 \times \left(\frac{znaky}{slová}\right) - 29.5 \times \left(\frac{vety}{slová}\right) - 15.8$$

Existuje niekoľko indexov čitateľnosti. My sme sa rozhodli pre Coleman-Liau index, nakoľko nevyžaduje výpočet slabík jednotlivých slov, a tým je proces výpočtu pomerne rýchly.

5.2. Výpočet podobnosti

Ako základ pre výpočet podobnosti využívame vektory opísané v predchádzajúcej časti. Pre nájdenie najpodobnejších článkov k referenčnému článku využívame kosínusovú podobnosť, ktorá vyjadruje veľkosť uhla zvieraného dvoma vektormi a je široko využívaná v doméne spracovania textu a hľadania podobnosti [11].

Keďže sa vektor, ktorý reprezentuje konkrétny článok, skladá z viacerých zložiek, je možné, aby každá z týchto zložiek mala svoju vlastnú váhu, s ktorou sa zložka započítava do celkovej podobnosti [41]. Tento fakt umožňuje získať rôzne usporiadanie výslednej podobnosti aj so zohľadnením aktivity a preferencií používateľa, čiže modelu používateľa.

$$similarity = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} a_{ji} b_{ji}}{\sqrt{\sum_{j=1}^{m}\sum_{i=0}^{n} a_{ji}^2}\sqrt{\sum_{j=1}^{m}\sum_{i=0}^{n} b_{ji}^2}}$$

kde m je počet jednotlivých vektorov (v navrhnutej metóde 6 – pri využití všetkých zložiek) a n počet zložiek konkrétneho vektora. V celom procese predspracovania sa nejedná o časovo najnáročnejší problém, ako by sa na prvý pohľad mohlo zdať.

Druhý spôsob zisťovania podobnosti na základe vektorovej reprezentácie je Jaccard index. Táto metrika meria podobnosť dvoch množín (vektorov) ako:

$$similarity = \frac{|A \cap B|}{|A \cup B|}$$

kde A a B sú jednotlivé vektory. Ako je zrejmé, môžeme taktiež aplikovať váhy pre jednotlivé časti vektora reprezentujúceho článok. Metóda Jaccard index je z hľadiska výpočtovej zložitosti v porovnaní s kosínusovou podobnosťou rýchlejšia, čo sa odrazí pri spracovaní veľkého množstva dát na druhej strane neumožňuje využitie CLI. Preto sme sa rozhodli využiť obe metódy, prípadne ich kombináciu, kedy podobnosť pre zložku vektora CLI vypočítame na základe kosínusovej podobnosti a pre ostatné zložky sa využijeme metódu Jaccard index.

6 METÓDA ODPORÚČANIA NA ZÁKLADE PODOBNOSTI OBSAHU

Hlavným cieľom práce je návrh metódy pre personalizované odporúčanie založené na obsahu. Tento typ odporúčania stavia na zistenej podobnosti dvoch článkov, ktorú určíme na základe obsahu článku, chápaného ako postupnosť slov, rozšírenú o informácie, ako napr. nadpis alebo oblasť. Iným druhom podobnosti (napr. tematická) sa zaoberať nebudeme, nakoľko hlavnou zložkou je výpočet podobnosti a reprezentácia článkov, na ktorú sa zameriame.

Vstupom pre metódu odporúčania (Obr. 8) sú dva zoznamy:

- zoznam podobných článkov (získaný metódou opísanou v predchádzajúcej kapitole) – pre každý článok zoznam jemu podobných,
- aktivita používateľa (časovo zoradená) – aké články používateľ čítal.

V prípade zoznamu aktivity používateľa potrebujeme jednoznačne rozlíšiť konkrétnych používateľov a články, ktoré si zobrazili. Rovnako je nevyhnuté rozlíšiť články, ktoré danému používateľovi už odporučené boli a ktoré nie.

6.1. Model používateľa

Model používateľa vytvárame na základe aktivity používateľa, a teda sa jedná o model používateľa identifikovaný automaticky [29]. Uvažujeme len o zozname navštívených článkov jedného používateľa za isté časové obdobie. Na základe tejto aktivity používateľa odhadneme okruhy momentálneho (krátkodobého) záujmu používateľa. Keďže každý používateľ má svoju vlastnú inštanciu modelu, jedná sa o prekryvný model. Odporúčanie sa prispôsobuje každému používateľovi individuálne [1,2].

Metóda si uchováva poslednú návštevu používateľa, pre rozlíšenie dvoch „sedení“, pričom za rozdielne „sedenia“ považujeme návštevy, ktoré boli uskutočnené aspoň s hodinovým rozdielom. Jednotlivé údaje o používateľovi sa neevidujú explicitne, ale odvodzujeme ich na základe informácií o aktivite používateľa obsiahnutých v záznamoch servera a jednoznačne identifikovaných identifikátorom „cookie“.

Automatické sledovanie správania sa používateľa je výhodné najmä z hľadiska nutnosti interakcie používateľa pri vypĺňaní a získavaní modelu používateľa. V takomto prípade používateľ vykonáva bežné činnosti ako prehliadanie zaujímavých článkov, pričom táto aktivita sa zaznamená na strane servera. Následne vieme získať údaje o tom, aký článok používateľ prezeral, kedy tento článok prezeral, či sa jedná o článok odporúčaný navrhovanou metódou, alebo ho používateľ našiel iným spôsobom. Server zaznamenáva všetky kliknutia používateľa na portáli. Teda naše záznamy obsahujú záznam „používateľ - zoznam prehliadaných (kliknutých) článkov“.

Neevidujeme explicitnú spätnú väzbu o prehliadanom obsahu. V našom prístupe zjednodušene predpokladáme, že používateľ si klikne a zobrazí len také články, ktoré ho zaujímajú (podľa nadpisu, zhrnutia článku a pod.). Existujúce prístupy sledujú aktivitu používateľa po zobrazení článku ako pohyb kurzora, využívanie posuvníkov, časový úsek, ktorý používateľ strávi čítaním a pod. Pri časovom úseku je zohľadnená dĺžka textu, ktorú má používateľ prečítať. Zohľadňuje sa aj aktivita používateľa – pohyb kurzorom a iné. Zaujímavou metódou je skúmanie pohybu očí používateľa, kedy sa na základe kamier a polohy zreničiek určuje, kam sa používateľ momentálne pozerá [5]. My predpokladáme, že v prípade, ak používateľa daný článok nezaujme (rýchly odchod z daného článku), ide buď o zavádzajúci nadpis, alebo obsah nie je na požadovanej úrovni, ktorú používateľ očakával. Samotný fakt, že si používateľ už na tento článok klikol na základe nadpisu a zhrnutia, hovorí, že používateľ sa o danú problematiku zaujíma.

6.2. Získanie odporúčaných článkov

Prvým krokom je stanovenie počtu článkov, ktoré chceme používateľovi odporučiť. Zoznam článkov, ktoré sa používateľovi odporučia, pozostáva z dvoch častí:

- zoznam podobných článkov k článkom odporučeným a navštíveným (*S*)
- zoznam podobných článkov k článkom predtým neodporučeným, ale navštíveným (*N*-*S*)

Pomer medzi týmito časťami sa následne vypočíta ako:

$$S = N\left(1 - \frac{Nr}{V}\right)$$

kde *S* je počet podobných článkov k článkom navštíveným a odporúčaným, *N* je počet článkov, ktoré sa majú odporučiť, *Nr* reprezentuje počet článkov, ktoré neboli odporučené, ale používateľ ich navštívil a *V* je počet navštívených článkov celkom. Takýmto spôsobom vieme dynamicky meniť veľkosť jednotlivých zoznamov na základe aktuálnych preferencií používateľa.

Po vypočítaní pomeru pre jednotlivé pod-zoznamy nasleduje nájdenie samotných článkov, ktoré majú byť odporučené:

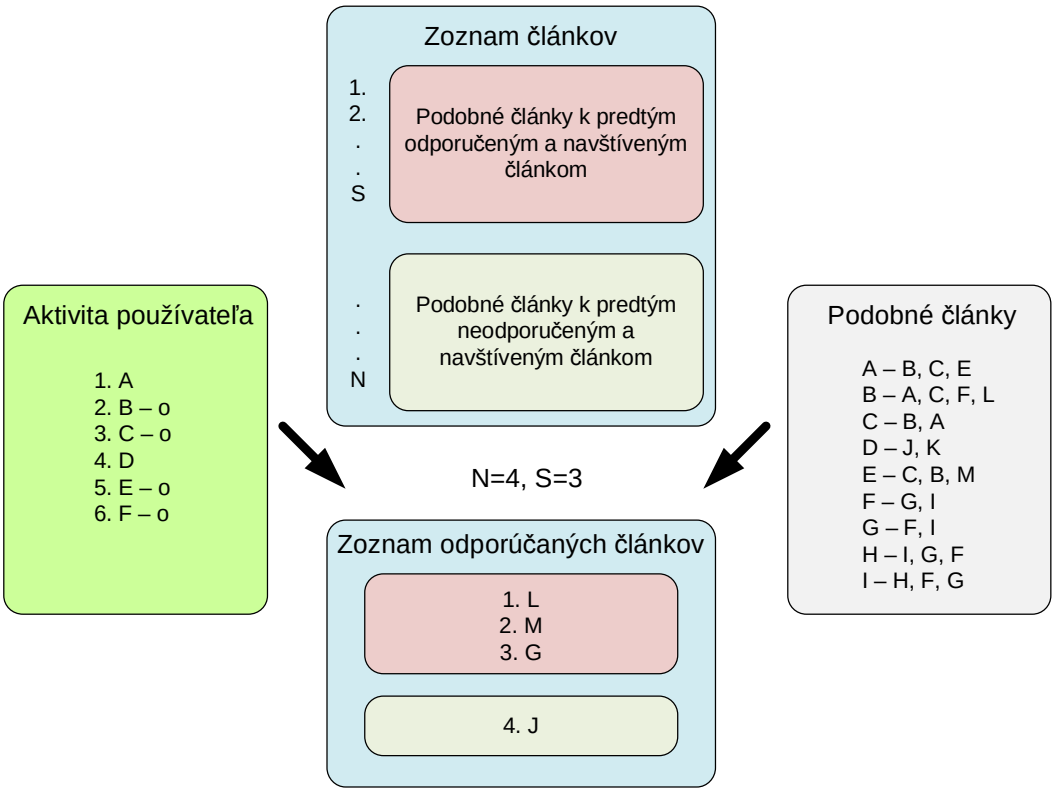
```
foreach user activity log do
 visited = get visited articles list
 visitedRec = get visited and recomommended articles list

 foreach visited do
   if randomNum > probability
     listPart1 = get first non visited article from computed
                 similarity list
   else
     listPart1 = get random non visited article
   end
 end

 foreach visitedRec do
   listPart2 = get first non visited article from computed
               similarity list
 end

 listToRecommend = listPart1[1..N] + listPart2[1..M]
end
```

Pre zoznam navštívených článkov nájdeme postupne pre každý z týchto článkov najpodobnejší a doteraz neprečítaný článok. S malou pravdepodobnosťou je však možné vložiť do zoznamu aj náhodný článok, aby sme sa vyhli tematickej monotónnosti odporúčaných článkov. Podobný prístup je aplikovaný aj pri druhom zozname navštívených, ale predtým odporučených článkov. Nakoniec dôjde ku spojeniu týchto dvoch zoznamov, čím vytvoríme jeden zoznam článkov, ktoré sa používateľovi zobrazia ako odporúčané (Obr. 18).

Takýmto spôsobom môžeme reagovať na aktuálne preferencie používateľa. V prípade, že zoznam aktivity používateľa neobsahuje dostatočné množstvo údajov, priradia sa články náhodne.

6.3. Príklad výpočtu

Príklad výpočtu je znázornený na Obr. 18. Atribút „-o“ pri aktivite používateľa znamená, že daný článok bol odporučený. V prípade, že chceme odporučiť 4 články (*N=4*), vyššie opísaným spôsobom získame pomer pre jednotlivé pod-zoznamy „3:1“ (tri podobné články k článkom odporučeným a navštíveným a jeden podobný ku článkom navštíveným, ale neodporučeným). V našom príklade máme 4 navštívené a odporučené články – *B, C, E, F*. Pre každý z týchto článkov sme získali metódou hľadania podobnosti zoznam podobných článkov. Pre každý z týchto článkov nájdeme zo zoznamu podobných také, ktoré ešte neboli používateľom navštívené. Napríklad pre článok *B* existuje len jeden nenavštívený článok *L*, ktorý sa pridá do zoznamu na odporučenie. Tento postup aplikujeme, pokiaľ nezískame potrebné množstvo článkov. V prípade, že neexistujú podobné a nenavštívené články ku konkrétnemu článku (napr. článok *C*), je tento článok preskočený, nakoľko používateľ už videl všetky relevantné články.



Obr. 18 – Metóda pre personalizované odporúčanie založené na podobnosti článkov.

V zozname aktivity používateľa sa nachádzajú dva neodporučené, ale navštívené články. Článok *A* preskočíme, pretože neexistujú k nemu podobné články, ktoré používateľ nenavštívil a bude sa pokračovať článkom *D*. Takýmto spôsobom získame kompletný zoznam 4 článkov, ktoré budú používateľovi odporučené.

Dynamický výpočet pomeru medzi jednotlivými pod-zoznamami umožňuje metóde prispôsobiť sa na aktuálne záujmy používateľa. V prípade, že používateľ nie je spokojný s odporúčanými článkami a využíva ostatnú navigáciu na danom sídle, bude sa „veľkosť“ prvého pod-zoznamu zmenšovať, zatiaľ čo veľkosť druhého bude narastať.

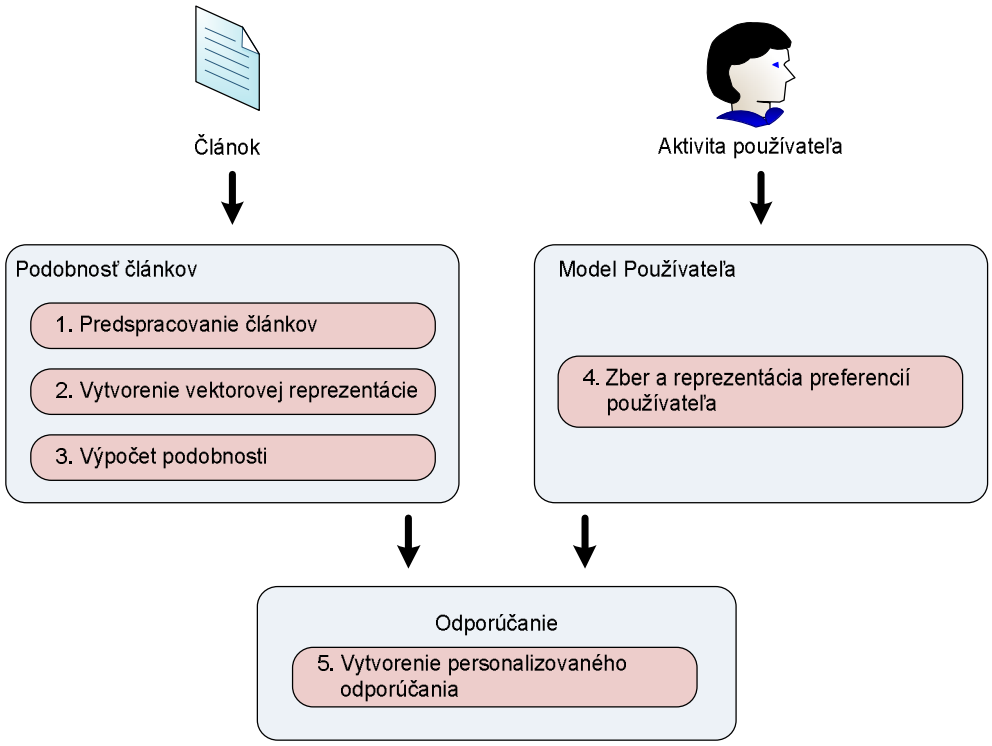
Pri navrhnutej metóde nie je potrebné zaznamenávať „vek článku“ - veličinu, ktorá udáva, ako dlho sa článok používateľovi zobrazuje medzi odporúčanými. V prípade, ak používateľ nenavštívi tento článok dostatočne dlhý čas (počet vygenerovaných odporúčaní), tento článok sa automaticky zo zoznamu odstráni.

7 REALIZÁCIA ODPORÚČANIA PRE SME.SK

V rámci portálu SME.SK je aktivita používateľov zaznamenávaná na strane servera. Jednotliví používatelia sú jednoznačne identifikovaní na základe dočasných záznamov prehliadača – „cookie“. Každý záznam nesie informáciu o čase, kedy bol daný článok zobrazený, jednoznačný identifikátor článku, identifikátor odporúčacej metódy, ktorou bol článok odporučený („0“-ak odporučený nebol) a url adresu z ktorej používateľ na daný článok prešiel.

Články sú jednoznačne identifikované na základe „id“. Pre každý článok potom rozoznávame názov, text článku, čas zverejnenia, vytvorenia a modifikácie článku, url adresu, krátke zhrnutie článku, sekciu a kategóriu.

Odporúčanie pozostáva z piatich krokov (Obr. 19). V prvom kroku je nevyhnutné novo pridané články predspracovať, vytvoriť im prislúchajúcu vektorovú reprezentáciu a vypočítať podobnosť s ostatnými (vybratými) článkami. Nezávisle od týchto činností, na základe aktivity používateľa, identifikujeme jeho preferencie, a tým upravujeme jeho model používateľa. Následne výstup z týchto častí využijeme pre vytvorenie personalizovaného zoznamu odporúčaných článkov, ktorý prezentujeme používateľovi.



Obr. 19 - Metóda personalizovaného odporúčania.

7.1. Extrakcia dát

Prvým krokom pre získanie dát, ktoré môžeme použiť pre predspracovanie, je ich získanie. V prípade pridania nového článku, je nevyhnutné tento článok stiahnuť (na základe „url“) a v danom HTML kóde extrahovať potrebné dáta. Z daného zdrojového kódu (Obr. 20) pomocou regulárnych výrazov extrahujeme Názov, Obsah, Autora, Dátum a Kategóriu daného článku.

```
<h1>Piráti sa neúspešne pokúsili druhýkrát uniesť loď Maersk Alabama</h1>
</div>
<div class="articlec col">
<div id="itext_content">
<p>NAIROBI. Somálski piráti dnes po druhý raz počas siedmich mesiacov zaútočili
na plavidlo Maersk Alabama. Strážna služba na palube nákladnej plaviacej sa pod
vlajkou Spojených štátov však pokus o únos odrazila. Informovala o tom námorná
flotila EÚ.</p>
<p>Piráti v apríli uniesli Maersk Alabama a kapitána Richarda Phillipsa päť dní
držali ako rukojemníka na záchrannom člne. Phillipsa oslobodili ostreloväči
špeciálnej námornej jednotky SEAL, ktorí pri zásahu zastrelili troch pirátov.</p>
<p>Somálski piráti zaútočili na loď automatickými zbraňami dnes ráno približne
350 námorných míl východne od somálskeho pobrežia, strážcovia na palube však
streľbu opätovali a únos odvrátili.</p>
<p>Podľa hovorcu námornej flotily EÚ Johna Harboura ide o "úplnú náhodu", že
piráti zaútočili <a href="http://www.sme.sk/c/4389042/snajperi-oslobodili-
uneseneho-kapitana.html">na loď Maersk Alabama už druhý raz.</a></p>
<p>Do vyšetrovania útoku sa zapojilo aj lietadlo flotily EÚ z Džibuti a
najbližšia loď námornej flotily EÚ je poverená pátraním po pirátoch, ktorí
zaútočili, uviedla flotila EÚ vo vyhlásení.</p>
</div>
<!-- eTarget ContextAd End -->

<p class="autor_line"><b>streda 18. 11. 2009 14:13</b> | Copyright &copy; TASR
2009<br /><span class="copyr"><a href="#" onClick="st_openWindow('/footer/',
'PetitPress','width=650,height=550'); return false;">&copy 2009 Petit Press.
Autorské práva sú vyhradené a vykonáva ich vydavateľ. Spravodajská licencia
vyhradená.</a></span></p>
```

Obr. 20 - Zdrojový kód stránky článku [sme.sk].

Niektoré spravodajské portály (reuters.com, nytimes.com) sprístupňujú rozhranie (API) pre získanie článkov, ktoré tento krok značne uľahčuje použitím niektorej zo štandardných notácií (XML, JSON a pod.)

7.2. Predspracovanie článkov

Hlavnú úlohu z pohľadu presnosti v procese hľadania podobného obsahu tvorí práve predspracovanie textu. Predspracovanie má vo všeobecnosti niekoľko krokov [15] v závislosti od požadovaného výstupného jazyka, keďže niektoré činnosti sú jazykovo závislé. Cieľom predspracovania textov je získanie postupnosti reťazcov z vstupného viac alebo menej štruktúrovaného textu – termov normalizovaných na slovné základy a rôzne objekty, odkazy [19] a pod.

7.2.1. Lexikálna analýza

V procese lexikálnej analýzy sa spracovávaný text podrobí samotnému rozdeleniu na základné značky. Samotné lexikálne analyzátory majú široké a tradičné využitie pri tvorbe kompilátorov programovacích jazykov [19]. Na úrovni lexikálneho analyzátora je nevyhnutné spracovať aj diakritiku, veľké písmená a pod.

Rozlišujeme základné značky ako „slovo“, resp. „číslo“. Samotnú lexikálnu analýzu riešime regulárnymi výrazmi. V tomto kroku rovnako eliminujeme všetky ostatné informácie zahrnuté v texte, ako napríklad url odkazy, HTML značky a podobne. Rovnako sú odstránené aj interpunkčné znamienka s výnimkou „.“. Bodka sa v ďalších krokoch využíva na zistenie jednoduchej sémantiky.

7.2.2. Stop slová

Stop slová sú slová špecifické pre konkrétny jazyk, ktoré nenesú významovú hodnotu („a“, „o“, „alebo“, „ale“ a pod.) a pritom môžu výrazne ovplyvniť nielen podobnosť medzi dvoma objektmi, ale aj veľkosť samotného priestoru unikátnych slov. Ich výskyt, resp. význam v texte je hlavne syntaktický [19].

Vo väčšine prístupov sa využíva statický zoznam slov [8], ktoré sa v procese predspracovania ignorujú. Niekedy sa však tento zoznam slov vytvorí, resp. nahradí zoznamom najčastejšie sa vyskytovaných slov v danej spracovávanej vzorke textov, ktorý môže byť vytvorený napríklad pomocou metódy TF-IDF („Term frequency-inverse document frequency“).

Pre účely spracovania článkov používame statický zoznam slov, ktorý obsahuje približne 200 takýchto slov pre slovenský jazyk. Zoznam slov vytvorených metódou TF-IDF sme sa rozhodli nevyužiť, nakoľko by sme odstránili aj slová, ktoré sa vyskytli zatiaľ len málo. To by znamenalo, že môžeme odstrániť aj mená osôb, názvy firiem a pod., ktoré sú často (napr. odhalenie škandálu) následne používané pri odporúčaní.

7.2.3. Lematizácia

Lematizácia ako taká patrí spolu so „stemmingom“ (redukcia na koreň slova) medzi metódy pre získanie základného tvaru slov [24]. Ako je zrejmé, slová sa vyskytujú v rôznych morfologických tvaroch. Rovnako ako pri stop slovách použitie takýchto slov by viedlo k neúmernému rozšíreniu priestoru unikátnych slov, nehovoriac o problémoch pri hľadaní podobnosti, kedy by slová z rozdielnym morfologickým tvarom boli považované za úplne ino-významové slová.

Lematizácia je metóda založená na slovníkovom princípe, kedy používaný slovník obsahuje možné morfologické tvary slov a ich prislúchajúce lemy. Lema sa v rámci morfológie považuje za kanonickú formu lexém slova. Lexéma je abstraktná jednotka reprezentujúca rôzne formy rovnakého slova [19].

Tento princíp zaručí pomerne presné kategorizovanie daných lém. Nevýhodou je však problém so slovami, ktoré sa v slovníku nenachádzajú a teda rovnako ako pri stop slovách budú takéto slová spracované samostatne. Treba však poznamenať, že takmer všetky metódy predspracovania čiastočne stierajú významové hodnoty daných slov, a tým pádom umelo zvyšujú podobnosť dvoch dokumentov (lema „mier“ je rovnaká pre „mier“, „miera“, „mieriť“ [19]).

Pre potreby lematizácie využívame slovníkový lematizátor JULS[4] [10]. Tento obsahuje približne 590 000 dvojíc slov a ich lém v slovenskom jazyku.

[4] Jazykovedný ústav Ľudovíta Štúra, Slovenská akadémia vied Bratislava

7.3. Reprezentácia a výpočet podobnosti

Po predspracovaní vytvoríme každému článku jeho vektorovú reprezentáciu, ktorá pozostáva zo 6 základných častí:

- názov článku
- TF (počet výskytu slov) slov z nadpisu v obsahu článku
- kategória článku (sekcia a kategória)
- mená a názvy
- kľúčové slová
- index čitateľnosti

V prípade kategórie článku portál SME.SK eviduje pre každý článok jeho sekciu a následne kategóriu v rámci danej sekcie. Pre extrakciu kľúčových slov, prípadne mien a názvov, sme sa rozhodli nevyužiť dostupné nástroje a služby, nakoľko vo väčšine prípadov neponúkajú podporu slovenského jazyka a priniesli by nárast procesu výpočtu z časového hľadiska. Takéto vektorové reprezentácie si následne uložíme do databázy, pretože sú potrebné pri výpočte podobnosti novo pridaných článkov.

Výpočet podobnosti pre nový článok realizujeme na základe kombinácie kosínusovej podobnosti a metódy Jaccard index. Podobnosť pre nový článok určujeme s poslednými 10 000 článkami, pričom za podobné články považujeme také, ktorých podobnosť nie je nulová. Následne pre každý článok získame zoznam maximálne 10 najpodobnejších článkov, ktoré uložíme do databázy. Pri každom takomto novo vygenerovanom zozname spätne upravíme zoznamy všetkým článkom, ktoré sa v ňom nachádzajú.

7.4. Preferencie používateľa a odporúčanie

Používateľ je v rámci portálu SME.SK identifikovaný na základe „cookie“. Po zobrazení článku je jeho aktivita zaznamenaná pričom pri zobrazení odporúčaného článku vieme tento jednoznačne odlíšiť od článku neodporúčaného. Takýmto spôsobom získavame zoznam používateľovej aktivity – prezreté odporučené a prezreté neodporučené články. Do úvahy berieme ohraničenú aktivitu - posledných maximálne 50 článkov.

Samotné odporúčanie pozostávajúce z 10 článkov potom vytvoríme na základe navrhnutej metódy, postupom opísaným v kapitole 6. Samotná konštrukcia zoznamu pre odporúčanie je pomerne rýchla, nakoľko vieme rýchlo vyhľadať podobné články na základe vopred vypočítanej podobnosti. V prípade, že nemáme o používateľovi dostatočné množstvo informácií (nedostatočná predchádzajúca aktivita), sú články doplnené o náhodne vybraté články z poslednej aktivity všetkých používateľov na portáli.

7.5. Prototyp

Pre potreby overenia navrhovanej metódy sme doteraz implementovali prototyp. Keďže predpokladáme, že metóda bude nasadená a integrovaná s už existujúcim riešením (projekt SMEFIIT), metóda je implementovaná v jazyku Ruby. Niektoré nezávislé časti, ktoré sa

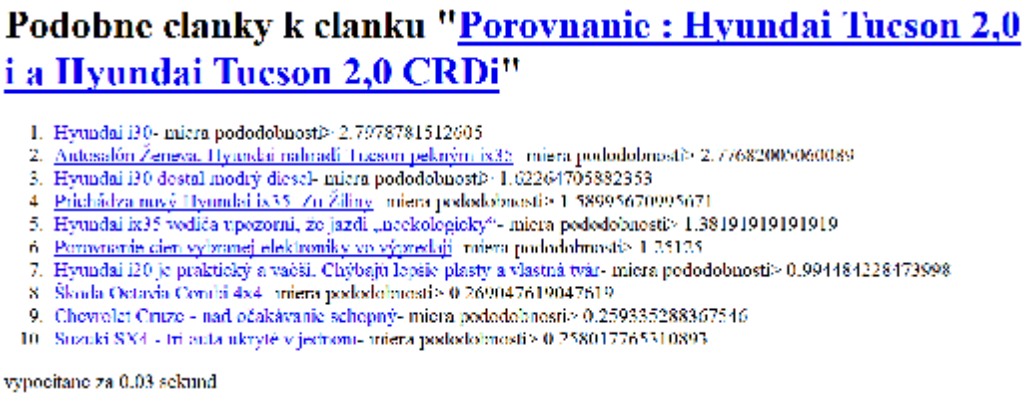
nespúšťajú pravidelne, boli implementované ako .NET/C# aplikácia, prípadne v oboch jazykoch pre porovnanie výkonnosti.

Prototyp zahŕňa:

1. Metódu pre extrakciu dát z HTML zdrojového kódu a XML (SME.SK)
   - Extrakcia Názvu, Obsahu, Kategórie článku
2. Metódu pre predspracovanie článkov v slovenskom jazyku
   - Odstránenie stop-slov, lematizácia, odstránenie interpunkčných znamienok
   - Extrakcia kľúčových slov
3. Metódu pre identifikovanie kľúčových slov pre daný korpus
   - Identifikácia slovných druhov[5]
4. Metódu pre reprezentovanie článku pomocou charakteristického vektora a následný výpočet kosínusovej podobnosti a Jaccard indexu
5. Metódu pre personalizované odporúčanie na základe aktivity používateľov
6. Nástroj pre vytvorenie vlastnej dátovej vzorky
7. Metódu pre spracovanie dát zo spravodajského portálu REUTERS.COM
   - Preklad textu do slovenského jazyka[6]

Samotná implementácia vyššie opísaných metód sa nachádza na priloženom médiu. V prílohe E-Technická dokumentácia uvádzame zdrojový kód metódy pre generovanie odporúčaní. Ukážku spracovávaných dát uvádzame v prílohe C-Ukážka vzorových dát SME.SK, ktorá obsahuje zdrojové dáta z portálu SME.SK. Príloha rovnako obsahuje aj ukážku reprezentácie týchto článkov navrhovanou metódou.

Jednotlivé metódy sú prístupné cez webové rozhranie (Obr. 21, Obr. 22), ktoré však bolo nahradené rozhraním na samotnej stránke portálu SME.SK prostredníctvom rozšírenia webového prehliadača (implementovaného v rámci projektu SMEFIIT), ktorý ponuku portálu SME.SK rozšíri o záložku odporúčané (Obr. 23).



Obr. 21 – Výpis podobných článkov prostredníctvom webového rozhrania.

[5] slovnik.juls.savba.sk – Jazykovedný ústav Ľudovíta Štrúra SAV

[6] translate.google.com

Odporucene clanky pre cookie "12609556664255691"

1. Jamkovyé nastrpí do štvrtka 25. februára
2. Mesto podpísalo zmluvu s Hospitale o prenájme nemocnice
3. Odbory, o ktoré je na trhu práce záujem, sa žiakom nechcú študovať
4. Trak volí parlament, čo zaujíma Američanov
5. Grécko sa pripravuje na ďalšie protesty
6. Bezmenní v Sudáne stále volajú: SOS Dárfúr!
7. Nech je zákon (ne)účinný
8. Mladý vodič nehodu neprežil
9. Trénera Kruegera Nemci zatiaľ neoslovili
10. Rekonštrukcia Župného domu sa má skončiť v apríli

vypocitane za 0.212001 sekund

Obr. 22 – Výpis odporúčaných článkov pre konkrétneho používateľa.

Obr. 23 – Integrácia odporúčania do portálu SME.SK.

8 OVERENIE RIEŠENIA - EXPERIMENTY

V rámci overenia navrhnutej metódy pre personalizované odporúčanie zaujímavých textov v doméne spravodajstva sme si stanovili nasledujúce ciele:

- experimentálne nastavenie váh jednotlivých častí vektora reprezentujúceho článok,
- vyhodnotenie úspešnosti hľadania podobných článkov,
- vyhodnotenie úspešnosti odporúčania celej metódy.

8.1. Testovacie dáta

Testovacie dáta pre overenie jednotlivých cieľov boli vybrané z domény spravodajského portálu SME.SK. Následne sme vytvorili referenčnú vzorku podobných článkov, ktoré sme manuálne označili jedným zo stupňov podobnosti (100 článkov – 10 000 dvojíc):

1. Sú veľmi podobné (rovnaká „kauza“, téma, udalosť a pod.) – celý článok
2. Sú čiastočne podobné (rovnaká „kauza“, téma, udalosť a pod.) – časť článku
3. Sú trochu podobné (články sú podobné v menej ako 1/3 obsahu)
4. Sú veľmi málo podobné (v článku sa spomínajú rovnaké osoby, firmy, oblasti a pod.)
5. Nie sú podobné

Dáta pre vyhodnotenie úspešnosti odporúčania celej metódy priamo získavame z portálu SME.SK, následne priamo využívame pri odporúčaní používateľom, od ktorých sa získava spätná väzba o samotnom odporúčaní.

Vo vzorke sú rovnomerne zastúpené články z jednotlivých kategórií (momentálne 52), v ktorých sú články zaradené s ohľadom na početnosť jednotlivých skupín. Túto referenčnú vzorku využili pre overenie metódy hľadania podobných článkov.

Ďalšou dátovou vzorkou použitou v rámci overenia riešenia je vzorka 1000 článkov, ku ktorým sú priradené podobné články. Táto vzorka je automaticky získaná z portálu SME.SK, pričom podobné články sú extrahované z päty konkrétnych článkov, kde sa uvádzajú súvisiace články, pričom však pre metodiku pridávania takýchto článkov nemôžeme tvrdiť, že sa jedná o najrelevantnejšie články.

Pre overenie samotnej metódy personalizovaného odporúčania využívame aktuálne dáta na portáli SME.SK, kedy budeme používateľom (účastníkom experimentu) priamo personalizovať obsah tohto spravodajského portálu, či už priamo prostredníctvom stránky SME.SK, prípadne prostredníctvom iného rozhrania. Následne budeme prostredníctvom záznamov servera vyhodnocovať úspešnosť jednotlivých odporúčaní, a to tak, že v záznamoch servera sa po kliknutí na odporúčaný obsah objaví pri danom zázname príznak, ktorý samotné odporúčanie jednoznačne identifikuje.

8.2. Návrh experimentov

Pri overovaní využijeme známe a široko používané metriky presnosť a úplnosť („precission“ a „recall“). V doméne získavania znalostí sú tieto metriky definované ako:

- Presnosť - počet relevantných dokumentov („relevant documetns“) vrátených metódou predelený počtom dokumentov celkovo vrátených metódou („retrieved documents“)

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

- Úplnosť – počet relevantných dokumentov vrátených metódou („relevant documents“) predelený počtom všetkých relevantných dokumentov vo vzorke („relevant documents“)

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

Vyššie uvedené metriky aplikujeme do nášho problému takto - presnosť vyjadruje pomer správne nájdených podobných článkov ku všetkým podobným článkom. Úplnosť vyjadruje pomer správne nájdených podobných článkov ku relevantným podobným článkom v množine.

Metrika, ktorá kombinuje metriku presnosť a úplnosť do harmonického priemeru, sa nazýva F-metrika („F-measure“), tiež známa ako F_1 metrika, pretože presnosť a úplnosť majú rovnaké váhy. Je definovaná ako:

$$F = 2.\frac{precision\ .recall}{precision + recall}$$

Nadobúda hodnoty z intervalu $\langle 0,1 \rangle$ a čím je vyššia, tým je hodnotený systém úspešnejší. Keďže F- metrika je kombináciou presnosti a pokrytia, korešponduje dosiahnutie vysokej F-metriky nájdeniu kompromisu medzi presnosťou a pokrytím [19].

V rámci overenia metódy pre hľadanie podobných článkov sme využili vytvorenú referenčnú vzorku, ktorej sme manuálne označili mieru podobnosti daných článkov. Následne sme vyššie opísaným postupom vyhodnotili úspešnosť metódy. Výsledky umožnili aj nastavenie jednotlivých parametrov (váh zložiek vektora), keďže práve prostredníctvom daných parametrov môžeme výrazne ovplyvniť zoznam nájdených podobných článkov. Nastavenie jednotlivých váh prebiehalo na základe evolučného algoritmu, pričom za „fitness funkciu“ bola považovaná F-metrika.

Rovnako sme experimentovali s rôznymi nastaveniami predspracovania článkov a ich vplyvom na výpočtovú náročnosť, ale aj presnosť celej metódy.

Overenie celej metódy personalizovaného odporúčania prebiehalo prostredníctvom syntetických testov. Na základe záznamov servera SME.SK o návštevách používateľov sme vygenerovali odporúčanie v čase *t-1*. Vygenerované odporúčanie sme následne porovnali aplikovaním vyššie opísaných metrík so skutočným správaním používateľov v čase *t*.

Hypotézy:

A. Navrhnutá reprezentácia v dostatočnej miere reprezentuje článok, pričom z časového hľadiska je rýchlejšia ako štandardný prístup.
B. Navrhnutá reprezentácia a spôsob hľadania podobnosti prináša presnejšie výsledky hľadania podobných článkov ako bežne používané metódy (TF-IDF)

C. Metóda personalizovaného odporúčania vygeneruje články, ktoré si používatelia následne skutočne prezrú.

8.3. Overenie určovania podobnosti

Východiskovou metódou pre navrhnuté personalizované odporúčanie je samotné určenie podobnosti. Pre overenie navrhnutého prístupu sme realizovali niekoľko experimentov.

V rámci experimentovania s nastavením jednotlivých váh a overením metódy pre hľadanie podobných článkov sme realizovali experiment na dátovej vzorke REUTERS, ktorý, ako sa neskôr ukázalo, nebol vhodný pre overenie tak špecifickej podobnosti, akou sa zoberáme my. Bližšie experiment popisuje príloha - Experiment REUTERS.

V rámci ďalšieho overovania sme manuálne vytvorili dátovú vzorku pozostávajúcu z 100 článkov, ktoré boli navzájom označené stupňom podobnosti tak, ako je to opísané v časti testovacie dáta. Rovnako sme pri tomto experimente využili súvisiace články, ktoré sa nachádzajú v päte jednotlivých článkov.

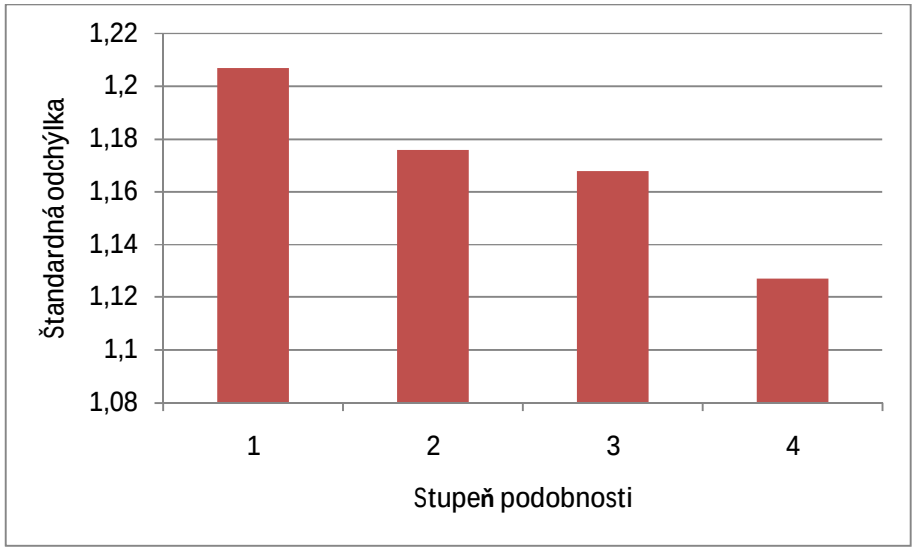
Navrhnutou metódou sme následne vypočítali zoznam podobných článkov pre obe množiny, ktorý sme porovnali voči metóde TF-IDF aplikovanej taktiež na obe dátové vzorky. Pre manuálne vytvorenú vzorku sme pri výpočte metrík zohľadnili aj poradie, v ktorom sa podobné články nachádzali. Výsledky štandardných metrík a navrhovanej metódy uvádzame v Tab. 7.

Tab. 7 – Vyhodnotenie hľadania podobnosti.

Vzorka	**SME.SK**		**Manuálne vytvorená**		
Metóda	***Navrhnutá metóda***	***TF-IDF***	***Navrhnutá metóda***		***TF-IDF***
			Kosínusová podobnosť	*Jaccard index*	
Presnosť	0.165	0.091	0.700	0.843	0.511
Pokrytie	0.202	0.117	0.816	0.818	0.587
F-Metrika	**0.182**	0.102	**0.753**	**0.870**	0.546

Ako vidíme, navrhnutá metóda značne zvýšila pokrytie a presnosť oproti klasickej metóde TF-IDF, pričom výpočtový proces sa zrýchlil 2.46 - krát (kosínusová podobnosť) a takmer 2 - krát (Jaccard index oproti kosínusovej podobnosti). Nízke hodnoty F-Metriky pri vzorke získanej z päty článkov na SME.SK môžeme pripísať už spomínanej neúplnosti (manuálne neboli označené všetky prípadne najrelevantnejšie články), kedy naša metóda a metóda TF-IDF pravdepodobne našli aj podobnejšie články.

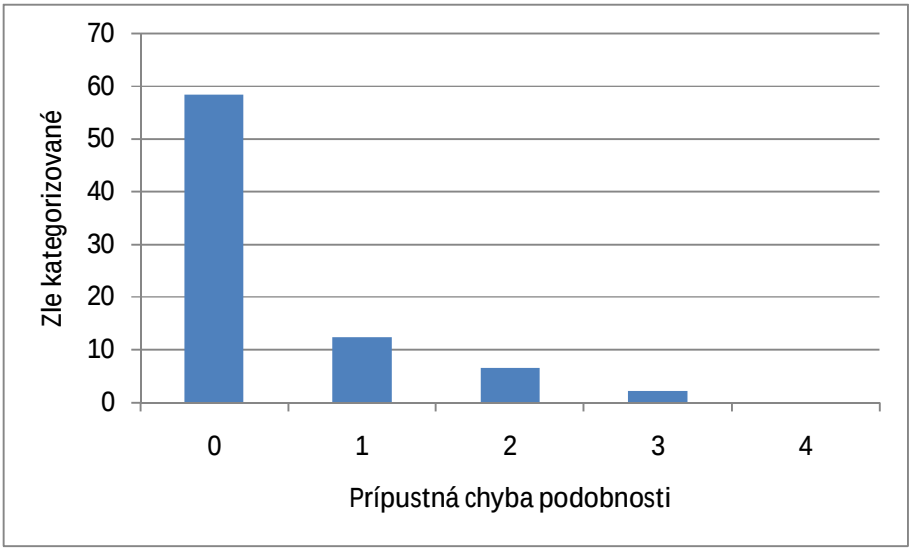
Na základe označených stupňov podobnosti sme vypočítali aj štandardnú odchýlku pre manuálne označenú vzorku. Výsledky navrhnutej metódy „<0,1>“ sme namapovali na päť stupňov podobnosti, ktoré boli použité pri tvorbe vlastnej vzorky. Následne sme pre jednotlivé stupne vypočítali štandardné odchýlky (Obr. 24)



Obr. 24 – Štandardná odchýlka pre jednotlivé stupne podobnosti.

Najväčšia odchýlka sa vyskytla pri stupni podobnosti 1 a predstavovala hodnotu 1,207, čo je pre potreby odporúčania článkov prijateľné.

Na Obr. 25 je znázornená priemerná dovolená chyba podobnosti a priemerný počet zle kategorizovaných článkov. V prípade, že povolíme chybu o veľkosti jedného stupňa podobnosti, získame priemerne len 12,5% zle kategorizovaných článkov pre jednotlivé stupne podobnosti.



Obr. 25 – Počet zle kategorizovaných článkov v závislosti od prípustnej chyby.

Váhy jednotlivých vektorov sme našli pomocou evolučného algoritmu, kedy sme ako hodnotiacu „fitness“ funkciu zvolili F-Metriku v porovnaní s manuálne vytvorenou vzorkou. Ako príklad uvádzame, že navrhnutá reprezentácia priniesla 4 krát lepšie výsledky ako pri využití len názvu článku a najmenej 1,4 - krát lepšie výsledky ako len pri využití kľúčových slov. Využitie časti

kategória prinieslo zlepšenie len 1,15 krát. Avšak predpokladáme, že môže byť užitočné v prípade malého počtu článkov ku konkrétnej téme.

8.4. Overenie personalizovaného odporúčania

Personalizované odporúčanie sme overili prostredníctvom syntetických testov. Na základe záznamov servera počas 3 dní sme pre používateľov (identifikovaných na základe „cookie“) vytvorili odporúčanie navrhnutou metódou. Následne sme porovnali články odporúčané so skutočnými článkami, ktoré si používatelia aj skutočne prezreli.

Problémom pri tomto type experimentu je, že v podstate predikujeme správanie používateľov, ktorí momentálne vo väčšine prípadov používajú podobný model. Na portáli SME.SK sa momentálne používateľom zobrazujú najčítanejšie články pre rôzne časové obdobia, rovnako sa aktuálne články zobrazujú na titulnej stránke. Väčšina používateľov preto automaticky prechádza len tieto odkazy a vzhľadom na priemerný čas, ktorý strávia pri jednej návšteve, nemajú možnosť vyhľadať aj iné články.

Iným aspektom, ktorý je nevyhnutné zohľadniť pri hodnotení odporúčania založeného na obsahu, je jeho samotná povaha. Pri odporúčaní založenom na obsahu odporúčame „do hĺbky“ takýto typ odporúčania je vhodný napríklad ako náhrada statických zoznamov podobných článkov, ktoré sa nachádzajú za konkrétnym článkom. Iný prístup je kombinácia viacerých typov odporúčania (napr. kolaboratívne), kde by bolo zohľadnené aj prehliadanie „do šírky“.

Samotné syntetické testy prebehli s rôznou granularitou vyhodnotenia, kedy sme sa zamerali na konkrétne články alebo všeobecnejšie kategórie. Pri konkrétnych článkoch sme skúmali, či si používatelia prezreli konkrétne články, ktoré sme im odporučili. V prípade kategórií sme kontrolovali odporúčanú kombináciu sekcie a kategórie konkrétneho odporučeného článku s článkami, ktoré si používatelia skutočne prezreli. Rovnako sme menili aj časové okná, z ktorých sa odporúčalo a overovalo so zohľadnením rôznych počtov návštev pre jednotlivé okná.

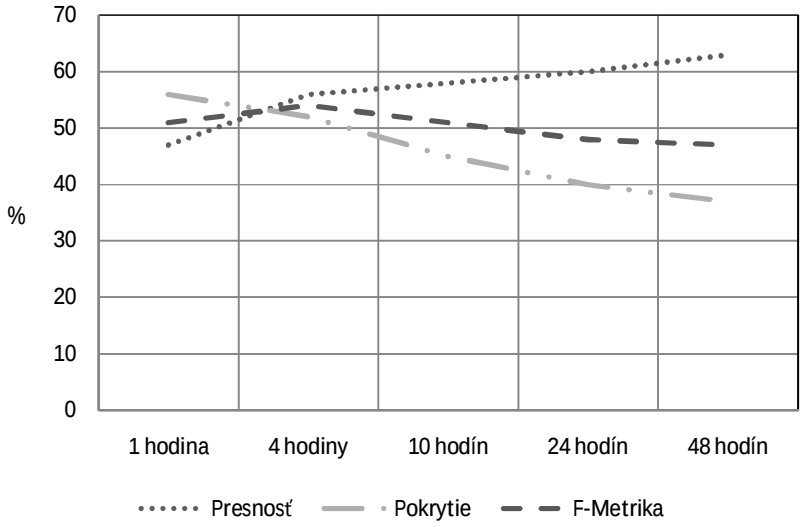
Výsledky pre jednotlivé nastavenia uvádzame v Tab. 8.

Tab. 8 – Vyhodnotenie syntetických testov.

	Odporúčacie okno [h]	**Overovacie okno [h]**	**Presnosť**		**Pokrytie**		**F1-metrika**	
			Kos.	Jacc.	Kos.	Jacc.	Kos.	Jacc.
Kategórie	9	63	43.23	64.05	50.28	36.26	46.49	46.31
	24	48	40.26	63.26	50.94	37.44	44.97	47.04
	33	39	39.73	62.12	51.36	39.92	44.80	48.63
	48	24	38.02	59.91	59.95	40.23	46.53	48.14
Články	9	63	1.43	1.83	0.84	0.77	1.06	1.08
	24	48	0.76	1.81	0.47	0.80	0.58	1.11
	33	39	0.67	1.68	0.49	0.85	0.57	1.13
	48	24	0.5	1.53	0.64	1.34	0.56	1.43

Ako je zrejmé, najlepšie výsledky (presnosť) sme získali pri overovacom okne 63 hodín. Rovnako pri článkoch tak aj pri kategóriách si používatelia s väčším časovým oknom pozreli články odporúčané, prípadne kategórie článkov totožné s článkami odporúčanými. Pokrytie bolo pri

dlhšom overovacom okne nižšie, nakoľko si používatelia prezreli úmerne viac článkov ako pri kratších overovacích oknách (Obr. 26).



Obr. 26 – Synteticé testy – presnosť, pokrytie a F-metrika.

Nízke hodnoty pri porovnaní samotných článkov môžeme vysvetliť už vyššie spomínaným fenoménom, kedy návštevníci portálu SME.SK vo väčšine prípadov klikajú na najčítanejšie články priamo zobrazované na hlavnej stránke. Na druhej strane pomerne vysoké skóre pre porovnanie kategórií naznačuje, že používatelia sa zameriavajú na kategórie, ktoré im boli odporučené, pričom pri samotnom odporúčaní sa kategória vôbec nezohľadňuje.

Navrhnutú metódu sme porovnali s metódou „TRecom“ (metóda pre odporúčanie založené na obsahu so stromovou reprezentáciou) [40]. Navrhnutá metóda získala lepšie výsledky (Tab. 9) pre dlhšie odporúčacie okná (F-Metrika). Pri kratších oknách vykazovala naša metóda nižšie pokrytie, pričom presnosť bola vyššia pre všetky porovnávané časové okná.

Tab. 9 – Porovnanie navrhnutej metódy a metódy TRrecom [40].

Overovacie okno [h]		**1 h**	**4 h**	**10 h**	**24 h**	**48 h**
TRecom	**Precision**	40	49	56	58	59
	Recall	71	60	44	32	25
	F1-Mesure	51	54	49	41	35
Navrhnutá metóda	**Precision**	47	56	58	60	63
	Recall	56	52	45	40	37
	F1-Measure	51	54	51	48	47

Rovnako sme experimentovali s potrebnou veľkosťou predchádzajúcej aktivity používateľa pre dosiahnutie čo najlepšieho odporúčania. Výsledky korelujú s veľkosťou odporúčacieho okna, od ktorého priamo závisí počet klikov na danom sídle. Ukázalo sa, že pri 15 a viac kliknutiach dokázala metóda získať najlepšie výsledky.

9 ZÁVER

Problematika personalizovaného odporúčania je v súčasnosti široko skúmaná. Prístup odporúčania nie je zaujímavý len z pohľadu sprístupňovania informácií používateľom, ale zaujímajú sa o ňu aj prevádzkovatelia rôznych internetových obchodov, kedy sa používateľom odporúčajú pri nákupe podobné výrobky.

Jedným zo základných prístupov pre personalizované odporúčanie je odporúčanie založené na obsahu. Tento prístup je náročný z hľadiska výpočtovej zložitosti, pri veľkom počte spracovávaného obsahu a nutnosti rýchlej odozvy systému.

V práci sme navrhli metódu, ktorou riešime tento problém úspornou reprezentáciou odporúčaného obsahu, čím sa zabezpečí rýchle hľadanie podobnosti nad danými objektmi. Výhodou navrhovanej reprezentácie je, že každá zložka navrhnutého vektora má vlastnú váhu, ktorá sa môže pre konkrétneho používateľa dynamicky meniť, a tak jednou metódou dostaneme rôzne výsledky podobného obsahu.

Navrhnutá reprezentácia obsahuje aj index čitateľnosti daných článkov, čo by malo priniesť vyššiu mieru spokojnosti používateľov, nakoľko za podobnejšie články označíme také, ktoré sú si podobnejšie z hľadiska náročnosti čítania a pochopenia. Vychádzame pritom z predpokladu, že napr. človek s vyšším vzdelaním uprednostní rovnaké, zložitejšie články na rovnakú tému, ako len nejaké triviálne „bulvárne“ články a podobne. Samozrejme, že sa jedná o vysoko individuálne preferencie, ktoré môžu byť aj obrátené, prípadne človek číta rôzne články, na čo však navrhovaná metóda reaguje a prispôsobuje sa.

Odporúčanie založené na obsahu je široko používané v kombinácii s kolaboratívnym odporúčaním, kedy vo všeobecnosti prinášajú lepšie výsledky.

Motiváciou tejto práce je zefektívniť a zjednodušiť prístup používateľov k obsahu spravodajského portálu prostredníctvom metódy personalizovaného odporúčania spravodajských článkov. Analyzovali sme dostupné a široko používané prístupy pre personalizované odporúčanie, uviedli sme niektoré existujúce riešenia, ktoré sa problematikou personalizácie v danej doméne zaoberajú.

Ďalej sme dokumentovali špecifiká doménovej oblasti spravodajského portálu vo všeobecnosti, resp. na príklade konkrétneho portálu. Jadro práce tvorí návrh metódy personalizovaného odporúčania založeného na obsahu s dôrazom na reprezentáciu článkov a hľadanie podobnosti, ktoré prebieha v piatich základných krokoch:

1. Predspracovanie článkov.
2. Vytvorenie vektorovej reprezentácie.
3. Výpočet podobnosti.
4. Vytvorenie a úprava modelu používateľa.
5. Odporučenie personalizovaného obsahu.

Články reprezentujeme viac-zložkovým vektorom, ktorý obsahuje váhu pre jednotlivé zložky, a tak môže byť dynamicky upravovaný pre konkrétnych používateľov. Vektor je navrhnutý tak, že dokáže reprezentovať aj články, ktoré nesú iný ako textový obsah.

V kapitole Overenie riešenia opisujeme navrhované experimenty a vyhodnotenie úspešnosti predkladanej metódy. Pri overovaní odporúčania je nevyhnutné zohľadniť momentálne správanie používateľov na danom spravodajskom portáli. Samotné odporúčanie založené na obsahu je vhodné využiť v prípade, kedy má používateľ záujem hlbšie sa oboznámiť s určitou problematikou, prípadne ho využiť v kombinácií s inými metódami. Aktuálne správanie používateľov na danom portáli je silne ovplyvnené dostupnými navigačnými pomôckami (najčítanejšie články za určité časové obdobie). To vysvetľuje pomerne malé skóre získané pri syntetických testoch a zameraní na konkrétne články.

Hlavným prínosom práce je navrhnutie úspornej reprezentácie článkov a jej následné využitie pri hľadaní podobnosti článkov, ktoré skrátilo výpočtový proces oproti metóde TF-IDF približne 2,46 krát (pri počte článkov 10 000 to znamená úsporu viac ako 8h výpočtového času). Podobne aj pomocou navrhnutej metódy personalizovaného odporúčania dokážeme generovať odporúčané články v reálnom čase, čo je pre dynamicky sa meniace domény kľúčový faktor.

Navrhnuté metódy nájdu uplatnenie ako náhrada väčšinou staticky vytváraných odporúčaní – podobných článkov, ktoré sa nachádzajú na konci článku. Pri kombinácií s inými spôsobmi odporúčania môžu byť zahrnuté aj do „štandardného“ odporúčania, kedy pokrývajú prehľadávanie „do hĺbky“. Metódy sú navrhnuté ako jazykovo nezávislé, po náhrade jazyka lematizátora a zoznamu stop-slov, môžu byť použité na iné jazyky. Rovnako je možné metódy využiť aj na iné spravodajské portály, keďže sa vo väčšine prípadov dá jednoznačne identifikovať názov, obsah, prípadne kategória článku.

LITERATÚRA

1. Barla, M., 2006. *Zachytávanie záujmov používateľa na webe*. Diplomová práca. Vedúci : Mária Bieliková. Slovenská technická univerzita.
2. Barla, M., Tvarožek, M., Bieliková, M., 2009. *Rule-based user characteristics acquisition from logs with semantics for personalized web-based systems.* In Computing and Informatics, Vol. 28, No. 4.
3. Bennett, P. N., Carbonell, J. G., 2007. *Combining Probability-Based Rankers for Action-Item Detection.* In Proc. of the Human Language Technologies — North American ACL 2007 Conf., Rochester, New York.
4. Buscher, G., van Elst, L., Dengel, A., 2009. *Segment-level display time as implicit feedback: a comparison to eye tracking.* In Proc. of the 32nd international ACM SIGIR Conf. on Research and Development in information. SIGIR '09. ACM, New York, NY, pp. 67-74.
5. Carrico, J. A., Pinto,F. R., Simas, C., 2005. *Assessment of band-based similarity coefficients for automatic type and subtype classification of microbial isolates analyzed by pulsed-field gel electrophoresis*. Journal of Clinical Microbiolgy, Vol. 43, pp. 5438-5490.
6. Coleman, M., Liau, T. L., 1975. *A computer readability formula designed for machine scoring*. Journal of Applied Psychology, Vol. 60, pp. 283-284.
7. College, F. D., Dellaert, F., 2002. *The expectation maximization algorithm.*College of Computing, Georgia Institute of Technology.
8. Češka, Z., 2009. *Porovnání technik předspracování textu pro detekci plagiátu*. Proc. of the 8th Annual Conference ZNALOSTI 2009, pp. 293-296.
9. Debnath, S., Ganguly, N., Mitra, P., 2008. *Feature weighting in content based recommendation system using social network analysis.* In: WWW '08: Proceeding of the 17th international conference on World Wide Web. ACM, New York, NY, USA, pp. 1041-1042.
10. Garabik, R. 2006. *Slovak morphology analyzer based on levenshtein edit operations.* In Proc. of 1st Workshop on Inteligent and Knowledge Oriented Technologies, WIKT 2006, Bratislava, Slovakia, pp. 2-5.
11. Garcia, E., 2006. *Cosine similarity and Term weight tutorial.* Dostupné z: http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html. (10.11.2009)
12. Goldberg, D., Nichols, D., Oki, B. M., Terry, D., 1992. *Using collaborative filtering to weave an information tapestry*. Communications of the ACM 35 (12), pp. 61–70.
13. Jaccard, P., 1901. *Étude comparative de la distribution florale dans une portion des Alpes et des Jura*. Bulletin del la Société Vaudoise des Sciences Naturelles 37, pp. 547–579.
14. Jacob, E. K., 2004. *Library trends: Classification and categorization: a difference that makes a difference*. Library Trends.

15. Jurgen, A., Teahan, W., 2005. *Universal Text Preprocessing for Data Compression*. IEEE Trans. Comput, pp. 497-507.
16. Kosková, G. 2009. *Dolovanie na webe*. Objavovanie znalostí. Dostupné z: http://www2.fiit.stuba.sk/~polcicova/ZZ/prednasky/09_web_mining.pdf.(5.4.2009)
17. Kosková, G. 2009. *Predspracovanie a transformácia dát*. Objavovanie znalostí. Dostupné z: http://www2.fiit.stuba.sk/~polcicova/ZZ/prednasky/02_preprocessing.ppt. (5.4.2009)
18. Kou, H., Gardarin, G., 2009. *Keywords Extraction, Document Similarity and Categorization*, technique reports No.2002/22, PRiSM Laboratory of Versailles University.
19. Laclavík, M., Šeleng, M., 2008. *Vyhľadávanie informácií*. Dostupné z: http://ikt.ui.sav.sk/vi/vi_laclavik.pdf. (10.10.2009)
20. Lijuan , Zheng, et al., 2007. *Research and Improvement of Personalized Recommendation Algorithm Based on Collaborative Filtering*. In JCSNS International Journal of Computer Science and Network Security . Vol.7, No.7., pp. 134-138.
21. Lukashenko, R., Graudina, V., Grundspenkis, J., 2007. *Computer-based plagiarism detection methods and tools: an overview*. In Proc. of the 2007 international Conference on Computer Systems and Technologies. CompSysTech '07, vol. 285. ACM, New York, NY, pp. 1-6.
22. Maguitman, A. G., Menczer, F., Roinestad, H., and Vespignani, A., 2005. *Algorithmic detection of semantic similarity*. In Proc. of the 14th international Conference on World Wide Web.WWW '05. ACM, New York, NY, pp. 107-116.
23. Màrquez, L., Carreras, X., Litkowski, K. C., and Stevenson, S., 2008. *Semantic role labeling: an introduction to the special issue*. Comput. Linguist. 34, pp. 145-159.
24. Martinez, J. L., Garcia-Serrano, A., Martinez, P., Villena, J., 2003. *Automatic Keyword Extraction for News finder*. LNCS, 2004, vol. 3094.
25. Melville, P., Mooney, R. J., Nagarajan, 2002. *Content-boosted collaborative filtering for improved recommendations*. In Eighteenth National Conference on Artificial intelligence (Edmonton, Alberta, Canada). R. Dechter, M. Kearns, and R. Sutton, Eds. American Association for Artificial Intelligence, Menlo Park, CA, pp. 187-192.
26. Pazzani, M., Billsus, D., 2007. *Content-based recommendation systems*. pp. 325-341.
27. Qi, X., Davison, B. D., 2009. *Web page classification: Features and algorithms*. ACM Comput. Surv. 41, pp. 1-31.
28. Sarabjot, A., Bamshad, M., 2005. *Intelligent techniques for web personalization*, pp. 1-36.
29. Shen, X., Tan, B., Zhai, C., 2005. *Implicit user modeling for personalized search*. In Proc. of the 14th ACM international Conference on information and Knowledge Management. CIKM '05. ACM, New York, NY, pp. 824-831.
30. Soller, A., 2004. *Computational Modeling and Analysis of Knowledge Sharing in Collaborative Distance Learning*. *User* Modeling and User-Adapted Interaction 14, 4, pp. 351-381.
31. Towle B., Quinn, C., 2000. *Knowledge based recommender systems using explicit user models*. AAAI Technical Report WS-00-04.
32. Vojtek. P, Bieliková M., 2010. *Homophily of Neighborhood in Graph Relational Classifier*. SOFSEM 2010.

33. Watters, C., Wang, H., 2000. *Rating news documents for similarity*. J. Am. Soc. Inf. Sci. 51, pp. 793-804.
34. Witten, I. H., Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition.* Morgan Kaufmann series in data management systems. Morgan Kaufmann.
35. Wongchokprasitti, C., Brusilovsky, P., 2007. *Newsme: A case study for adaptive news systems with open user model*. In: Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference, pp. 69.
36. Wu, Y., Chen, Y., Chen, A. L., 2001. *Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors.* In Proc. of the 11th international Workshop on Research Issues in Data Engineering. RIDE. IEEE Computer Society, Washington, DC, 17.
37. Xiuhong, W., Shiguang, J., Shengli. W., 2008. *Challenges in Chinese Text Similarity Research.* International Symposiums on Information Processing, pp. 297-302.
38. Xu, S., Jiang, H., Lau, F. C., 2009. *User-oriented document summarization through vision-based eye-tracking*. In Proc. of the 13th international Conference on intelligent User interfaces. IUI '09. ACM, New York, NY, pp. 7-16.
39. Yoneya, T., Mamitsuka, H., 2007. *Pure: a pubmed article recommendation system based on content-based filtering*. Genome informatics. International Conference on Genome Informatics 18,pp. 267-276.
40. Zeleník, D., 2010. Tvorba odporúčaní využitím reprezentácie vzťahov podobnosti. Diplomová práca. FIIT STU.
41. Ziegler, C., McNee, S. M., Konstan, J. A., Lausen, G., 2005. *Improving recommendation lists through topic diversification.* In Proc. of the 14th international Conference on World Wide Web. WWW '05. ACM, New York, NY, pp. 22-32.

PRÍLOHA A – ČLÁNOK NA KONFERENCIU ECWEB

Content-based News Recommendation

Michal Kompan and Mária Bieliková,

Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3,
842 16 Bratislava 4, Slovakia
kompan05@student.fiit.stuba.sk, bielik@fiit.stuba.sk

Abstract. The information overloading is one of the biggest problems nowadays. We can see it in various domains, including business, especially in the news. This is more significant in connection to news portals, where the quality of the news portal is commonly measured by amount of news added to the site. Then the most renowned news portals add hundreds of new articles daily. The classical solution usually used to solve information overloading is a recommendation. In this paper we present an approach for fast content-based news recommendation, based on cosine-similarity search.

Keywords: news, recommendation, vector representation, user model.

1 Introduction

There are plenty of news portals over the web. Renowned and influential portal contains hundreds of new articles from whole the world added daily. These articles cannot be easily accessed. For example users of the biggest Slovak news portal SME.SK spend daily approximately 16 min 34 sec on the site in usually two visits per day[1]. The amount of words on the websites increased two times since year 2003 and we can see this effect applied to links, pictures, tables, advertisements etc. More than 60% respondents participating in IDC research said, that they face up the information overloading in more than half of the time (see Fig. 1).

One of the quality criteria for a good news portal is time spending by reading considering the amount of useful information acquisition. It is extremely important to access new information as quick as possible. Importance of fresh news can be easily seen on various non news portals, where can be found various shorten top news.

We proposed a method for content-based news recommendation, which uses our devised effective article representation. This representation is important when similar articles are computed. Finally we use these similar articles to create recommended content based in implicit user model.

[1] Source www.aimmonitor.sk – Association of Internet Media

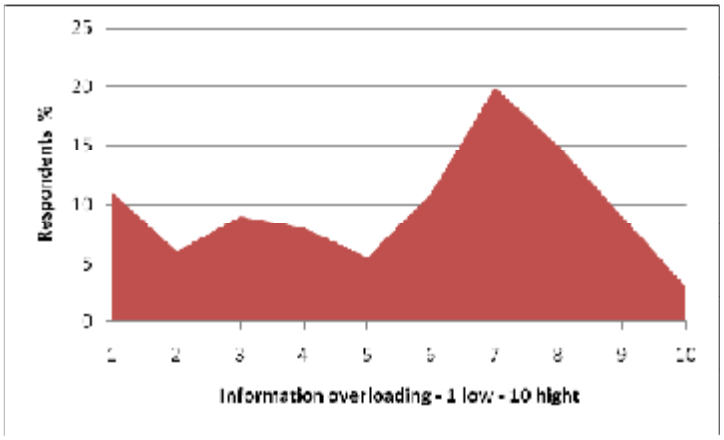


Fig. 1. Frequency of information overloading [IDC, autumn 2008, U.S., set of 500 respondents].

Our content-based method for recommendation is based on three steps – computing article similarity, creating a user model and the recommendation based on the first two steps (see Fig. 2).

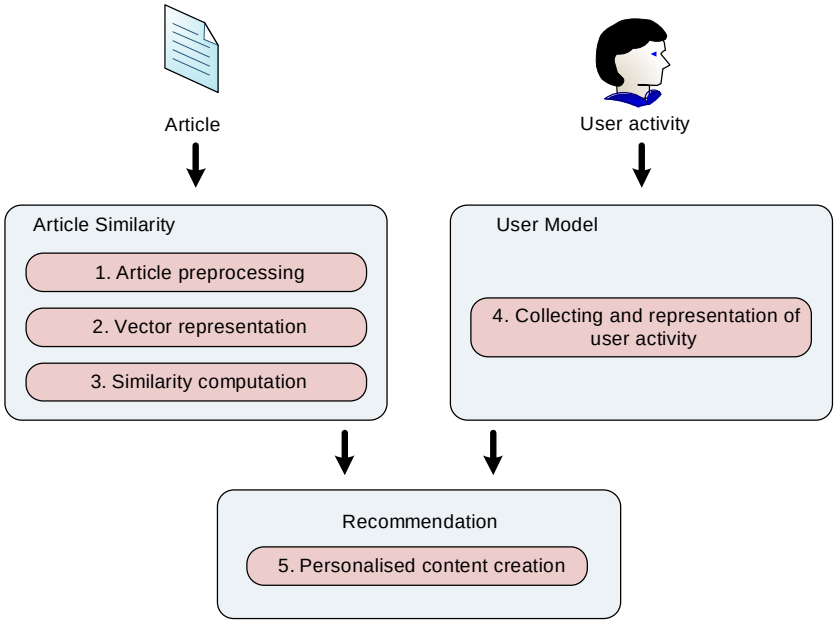


Fig. 2. Proposed news recommendation method.

In the article similarity step it is necessary to preprocess every article to reduce word space. Then the article is represented in an effective vector representation, which is used in similarity computation. As a result of article similarity step we obtain a list of similar articles for every article in the dataset.

The user model is created based on implicit feedback extracted from server logs by identification of visited and recommended article for unique cookie.

Finally is the recommended content from both similar articles and user model created. We will give deeply description for every step.

The paper is structured as follows. Section 2 describes state of work in the recommendation domain. In section 3 we provide overview of proposed vector structure representation. Section 4 describes our recommendation method. The evaluation of proposed method is described in section 5.

2 Related work

The recommendation is one of the actual research topics nowadays. There exist two basic approaches of the recommendation [8]. Traditional collaborative filtering accounts social element. Users are grouped into clusters based on their preferences, habits or content ranking. The problem of personalization is reduced to finding similar users and recommending new items to the users, which were visited and high ranked by other users in the same cluster.

Second approach of recommenders is based on the content-based filtering. The main goal is to identify two similar items-create “clusters” of sites instead of users. It is necessary to map user profiles (user models) to specific site clusters. This type of filtering is successful in well structured domains like movies, news [9].

These two approaches are widely used and mixed together, which usually brings better results [7], [2]. For example, we can find similar sites and then estimate user rank prediction for sites, which were not visited. The main problem in the content-based filtering is effective and enough expressive representation of items (or articles). This is often done by means of text summarization [3], keywords extraction [5] or by various categorization models [5]. These techniques are commonly used in English based systems and cannot be easily applied on other languages. Keywords extraction and summarization brings better results as other methods but are more time consuming. These methods cannot represent non-text documents without modification.

There are several recommender systems in the news domain. The problem within this domain which is rather similar to other business domains is extremely large amount of dynamically changed data. This causes that the recommendation is not provided directly over the whole data, when content-based recommendation is used [13], [14]. OTS system [13] use association rules to create “preference table” for every user. When there are a lot of new documents added daily, there is usual to not compute recommendation lists real-time [14]. Brusilovsky [12] has shown that explicit filled and open user model in the news domain brings usually worse results. Some systems have involved user location into recommendation systems, where recommendation list is created depending on user location [4].

3 Article similarity computation

For fast similarity estimation we propose effective vector article representation. This representation consists of six basic parts:

- *Title.* Lemmatized words from article title (aprox. 5 words – 150 000 Slovak article corpus.) This should be good describing attribute in the most occurrences.
- *TF of Title words in the article content.* We use term frequency to compute article relevance. If the article name is abstract and do not correspond to article content, we can reveal this situation. Term frequency is computes as follows:

$$tf_i = \frac{n_i}{\sum_k n_k} \quad \textbf{(1)}$$

  where tf_i is term frequency for term *i* (term from article title) and n_i is number of occurrences of term i in the document (article content) and $\sum_k n_k$ is the sum of numbers of all terms in document.
- *Names and Places.* We extract names and places from article content. There exists several names or places extractor for English language. We use simple approach to detect these items. As name or place is marked word starting with an upper letter and there is no sentence end before (dot, question mark etc.).
- *Keywords.* We store 10 more relevant keywords. Several news portals define list of keywords for every article. These keywords are unfortunately on various abstract levels for various news portals. We introduced our own keywords list based in TF-IDF computation (150 000 Slovak news articles from news portal SME.SK). We also removed any words except nouns and names.
- *Category.* Consists of "tree-based" category vector with weights. This vector is constructed based on specific news portal structure hierarchy (optional). This is useful, when not enough similar articles are found. The weight for every category is computed as:

```
n=1
For i=|Category| downto 0 do
        weighti=1/n
        n=n*2
end
```

- *CLI.* Coleman-Liau Index provides information of understandability of the text. This vector part is not important for standard similarity computation, but it is important in the results rearrangement. Our hypothesis is that the user wants to read articles of one similar level of understanding. This method is able to distinguish between two articles with similar title and different content ("Jaguar" – animal vs. car). CLI can be easily computed based on this formula [6]:

$$\mathrm{CLI} = 5.89 \times \left(\frac{\text{characters}}{\text{words}}\right) - 29.5\left(\frac{\text{sentences}}{\text{words}}\right) - 15.8 \quad \textbf{(2)}$$

When using this article representation, we can store article in the vector no longer than 30 items in most of occurrences. Example of proposed representation is given in Table 1.

Table 1. The example of vector article representation.

Vector part	Weights
Title	transplantácia_0.5 tvár_0.5
TF of title words in the content	transplantácia_0.0178571428571429 tvár_0.0714285714285714
Category	Sme.sk_0.5 PRESS_FOTO_1.0
Keywords	klinika_0.0357142857142857 povrch_0.0178571428571429 nos_0.0178571428571429 zub_0.0178571428571429 nerv_0.0178571428571429 svalstvo_0.0178571428571429 pacientka_0.0178571428571429 rozsah_0.0178571428571429
Names/Places	Cleveland_1
CLI	0.2543

For the purpose of similarity computation, we use cosine similarity [11], which is widely used in the information retrieval tasks. Our vector consists of 6 sub-vectors with weights so there is need to extend standard cosine similarity as:

$$similarity = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} a_{ji} b_{ji}}{\sqrt{\sum_{j=1}^{m}\sum_{i=0}^{n} a_{ji}^2}\sqrt{\sum_{j=1}^{m}\sum_{i=0}^{n} b_{ji}^2}} \quad \textbf{(3)}$$

where m is number of vector parts (6 in our method) and n is number of vector items.

3.1 News Preprocessing

Text pre-processing holds important role in the process of similarity search, because can significantly reduce word space. This part of the process is high language depending. Our experiments are provided in the Slovak language, which is one of the most complicated languages (declension of nouns, verbs etc.). The architecture of the system is variable, so pre-processing for Slovak language can be easily replaced by other languages and their methods (e.g. Porter algorithm[2]). For the speed of next computations, plays pre-processing a critical role. There is need to maximum reduction of article words dimensions.

[2] The Porter Stemming Algorithm page maintained by Martin Porter.
www.tartarus.org/~martin/PorterStemmer

The first task is to remove stop-words. We used static list, which can be replaced by TF-IDF output [10]. This method can identify commonly repeated words over the dataset.

As the main part of the pre-processing of Slovak language articles we used lemmatizing of the text. There is problem with algorithmic solution for this process, which can be solved by using dictionary of lemmas. The result we received is lemmatized (basic form) bag of words for every article.

It is necessary to note that we removed any punctuation except sentences ends. We use dots as a fast name or place indicator – when we check if there is a dot before an uppercase letter, and if not, it is probably personal name, or place etc. Name or Place extractor is one problem in information retrieval and is not main part of this article.

After keywords extraction we do not need whole article content anymore. We can safely delete all words except Title words obtained in the article content. Then for every processed article we have this list of words:

- Lemmatized article Title
- Lemmatized words from Content (which were included in the Title)
- 10 most relevant keywords
- List of Names and Places

Pre-processing methods we described above can significantly reduce number of words stored for every article up to 80%.

4 Recommendation

The most important part of proposed method is recommendation step (Fig. 3). For recommendation creation we need two lists as an input. First is list of 10 most similar articles for every article computed as we described above. Second list is list of visited articles for every user based on cookie. In this list we need to distinguish between articles visited but not recommended to users and articles visited and recommended before which can be easily done by extending article URL with special attribute.

Firstly we have to define number of articles to recommend (length of list to recommend). As we can see in Fig. 3 list of articles to recommend consists of two sub-lists.

- List of similar articles for visited and not recommended (S)
- Similar articles for visited and before recommended (N-S)

The ratio of this list is dynamically computed as:

$$S = number\ to\ recommend\ N\left(1 - \frac{Nr}{V}\right) \quad \textbf{(4)}$$

where S is number of similar articles for visited and not recommended articles, N is number of articles to recommend. Nr represents number of visited not recommended articles during the last session and V is number of visited articles together. For the proposed method, two sessions are distinguished as 1 hour break between visits.

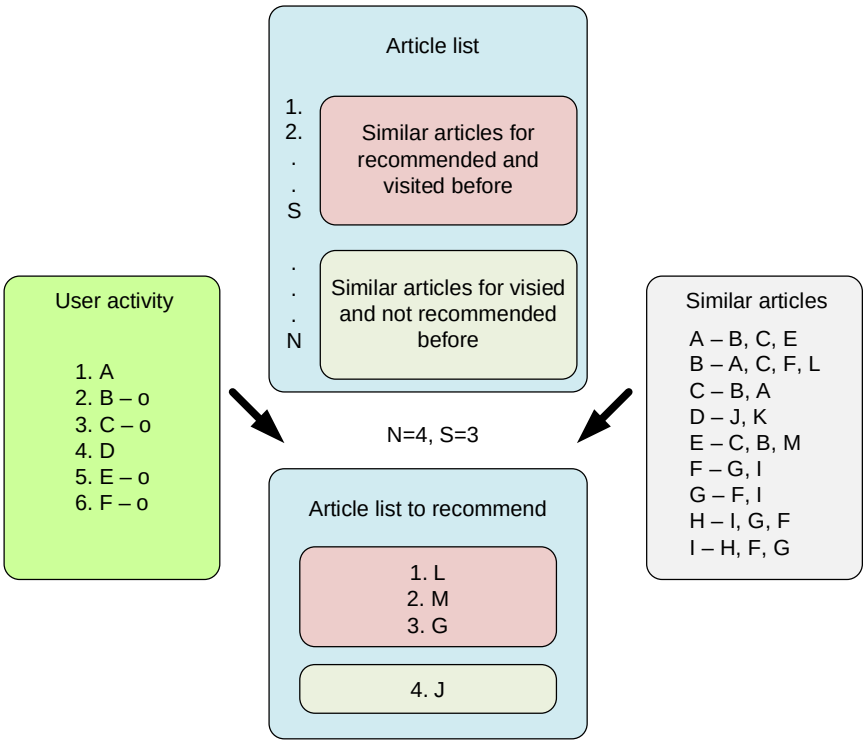


Fig. 3. Recommendation method steps.

Recommendation list is then computed for every part separately as follows:

```
foreach cookie do
 visited = get visited articles list
 visitedRec = get visited and recom. articles list
 foreach visited do
  if randomNum > 0.02
   listPart1 = get first non visited article from computed
               similarity list
  else
   listPart1 = get random non visited article
  end
 end

 foreach visitedRec do
  listPart2 = get first non visited article from computed
              similarity list
 end
 listToRecommend = listPart1[1..N] + +listPart2[1..M]
end
```

When there is not enough user activity (does not mean “cold start”) we use random article assignment. In this manner we can easily react to user’s most recent

preferences. For the list of recommended and visited articles we also introduced a coincidence – where user obtains a random article to the recommendation list.

Fig. 3 presents an example of recommended content creation. We have a list of user activities where "-o" attribute indicates whether the article was or was not recommended. Based on this we obtain list of visited articles and list of visited and before recommended articles. Then when we want to recommend 4 articles ($N=4$), we will obtain ratio 3:1 for sub-lists (similar articles for visited and recommended, similar articles for visited and not recommended before).

As we can see in our example, we have 4 visited and recommended articles *B, C, E, F.* We have found not visited article from the list of similar articles for every of these 4 articles. There is only one non visited similar article *L* for article *B.* This is repeated until the "before recommended" list is full. In the case when there do not exist non visited article in the similar list (article *C*), we skip this article, because the user saw all relevant articles for this "topic" already.

In our example there are 2 non recommended but visited articles, but there are no non visited articles for *A* – method will skip this article and will recommend first non visited for article *D.* In this manner we obtain a full list of 4 articles to recommend.

Dynamical computation of the ratio between sublist allows us to adapt for actual user activity and preferences. If the user does not like recommended articles and he uses other portal navigation, the size of first sub-list is decreasing while second part will increase respectively.

Our method stores "article age" for every recommended article. This number represents how long have been article recommended. If user does not visit this article for a defined time (number of recommendations) is this article deleted from the recommendation list as not interesting.

User activity list consists of pair cookie – visited article. We use implicit user model representation, where there no need to involve users into various forms completing or need of logging etc.

5 Experimental results

Proposed method was implemented within news recommendation system within the research project SME-FIIT [1].We evaluated the similarity computation over 10 000 articles from the Slovak news portal SME.SK, which is equivalent to one week time period. For this window we are able to estimate the similarity in 2-3 seconds (2,6 MHz Pentium, 4Gb RAM). The preprocessing takes approximately 20 seconds for the whole dataset. Then for the new article, when preprocessing is necessary, the whole computation process takes approximately 22s. When we need only re-estimate similarity with changed vectors parts weights is this process really fast as we mentioned above.

The accuracy of the similarity computation method was computed based on two datasets. The first one consists of 1 000 articles from news portal SME.SK. Every article from the dataset has assigned at least one similar article. These similar articles were obtained from the news portal, where there are mostly one or two similar articles quoted in the article footer. These similar articles are obviously chosen by the article author, which does not mean that there are not more similar articles.

The second dataset was the manually annotated dataset, which consist of 100 articles in 5 levels of similarity, so we obtained 10 000 article pairs with similarity level. Our method computed the list of similar articles for every article in the dataset. We compared these datasets to our method – the list of similar articles computed by our method and the list of similar obtained from one of two datasets with respect to order (more similar articles first). We calculated precision and recall and F-Score for every dataset and the method. Results were compared to standard text mining method TF-IDF as shown of Table 2.

Table 2. Similarity computation evaluation.

Dataset	**SME.SK**		**Manually annotated**	
Method	Our method	TFIDF	Our method	TFIDF
Precision	0.165	0.091	0.700	0.511
Recall	0.202	0.117	0.816	0.587
F-score	0.182	0.102	0.753	0.546

The dataset SME.SK is created based on "similar article" data (none, one or two) in the articles footers. These similarities are assigned by article's authors intuitively and often this choice does not mean not the only possibility but also one of the best matching articles. This is reflected in the results as we obtained only 0.182 F-score. Providing manual check we found out that our method in most cases founded more similar (and relevant) articles as the authors assigned. This indicates that manual similarity articles list creation by the article authors can be improved by our method.

We also computed standard deviation based on similarity levels. We mapped cosine similarity range <0, 1> to five similarity levels used in our manually annotated dataset. The biggest standard deviation was 1.21 "similarity level", which is an acceptable rate in the field of news recommendation.

6 Conclusion

In this paper we provided overview of short and high representative article vectors, which can be used for similarity search and real content-based recommendation in large and dynamically changing datasets and domains. A key future of this method is short article representing vector. Based on these vectors can be computed similarity between articles (text or non-text content) in a fast way. Every article vector consists of 6 sub-vectors based on article part used for their construction. Every part has its own weight, which can be dynamically changed to rearrange similar articles list to enable fast personalization.

The weight was found using evolution algorithms for every vector part, to obtain the best result. As an example, using proposed representation brings 4 times better precision than using only article title, and at least 1.4 time better results as using only keywords. The category part improves precision only 1.15 time, but on the other hand it can be useful when "no similar" article is in the dataset.

Based on this computed similarity further recommendation is created. User preferences are collected implicitly via server's logs. A recommended list consists of

two sub lists, where the first one represents similar articles to the visited and already recommended. The second sublist is based on similar articles for visited but not recommended before. In this way we can easily adapt to user preferences. The ratio between these two sub-lists is dynamically computed.

Proposed vector representation is a promising method for the fast news similarity computation to allow real time recommendation. We plan to make improvements on the precision and the recall, for example by using more sophisticated keywords extraction methods etc. and evaluate whole recommendation method by its implementation to existing news portal.

References

1. Barla, M., Kompan, M., Suchal, J., Vojtek, P., Zeleník, D., Bieliková, M., 2010. News recommendation. In Proc. of the 9th Znalosti.
2. Bouras, C., Tsogkas, V., 2009. Personalization Mechanism for Delivering News Articles on the User's Desktop. In Proc. of the 2009 Fourth int. Conf. on internet and Web Applications and Services (May 24 - 28, 2009). ICIW. IEEE Computer Society, Washington, DC, pp. 157-162.
3. Dakka, W., Gravano, L. 2007. Efficient summarization-aware search for online news articles. In Proc. of the 7th ACM/IEEE-CS Joint Conf. on Digital Libraries. JCDL '07. ACM, New York, NY, pp. 63-72.
4. Chen, Ch., Hong, Ch., Chen, S., 2009. Intelligent Location-Based Mobile News Service System with Automatic News Summarization. International Conference on Environmental Science and Information Application Technology, pp. 527-530.
5. Kou, H., Gardarin, G., 2009. Keywords Extraction, Document Similarity and Categorization. Tech.rep. No.2002/22, PRiSM Laboratory of Versailles Univ.
6. McCallum, D. R, Peterson, J. L., 1982. Computer-based readability indexes. In Proc. of the ACM '82 Conf. ACM 82. ACM, New York, NY, pp. 44-48.
7. Melville, P., Mooney, R. J., Nagarajan, 2002. Content-boosted collaborative filtering for improved recommendations. In Proc. of 18th National Conf. on Artificial intelligence (Edmonton, Alberta, Canada). AAAI, Menlo Park, CA, pp. 187-192.
8. Mobasher, B., Anand, S., S., 2005 Intelligent Techniques for Web Personalization: IJCAI 2003 Workshop, ITWP 2003. (Lecture Notes .. / Lecture Notes in Artificial Intelligence). Springer-Verlag New York, Inc.
9. Pazzani, M., Billsus, D., 2007. Content-based recommendation systems, pp. 325-341.
10. Ramos, J., 2000. Using TF-IDF to Determine Word Relevance in Document Queries. Tech. rep., Departament of Computer science. Rutgers University.
11. Tata, S., Patel, J.M., 2007. Estimating the Selectivity of tf-idf based Cosine Similarity Predicates. SIGMOD Record, Vol. 36, pp. 7-12.
12. Wongchokprasitti, C., Brusilovsky, P., 2007. Newsme: A case study for adaptive news systems with open user model. In: Autonomic and Autonomous Systems, 2007. ICAS07. Third Int. Conference, pp. 69.
13. Wu, Y., Chen, Y., Chen, A., L., 2001. Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors. In Proc. of the 11th int.l Workshop on Research Issues in Data Engineering. RIDE. IEEE Computer Society, Washington, DC, 17.
14. Yoneya, T., Mamitsuka, H., 2007. Pure: a pubmed article recommendation system based on content-based filtering. Genome informatics. International Conference on Genome Informatics 18, pp. 267-276.

PRÍLOHA B – NÁVRH ČLÁNKU DO ČASOPISU

Similarity Search and Personalized Recommendation in Dynamic Domains

x

Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3,
842 16 Bratislava 4, Slovakia
kompan05@student.fiit.stuba.sk, bielik@fiit.stuba.sk

Abstract. Nowadays we face up enormous informational overloading. The amount of information obtained in the weekend edition of a newspaper is similar to the amount which gained common man before hundred years for his whole life. Effective search should be scalable thus requires effective and accurate analysis of the content. We present a method for similarity search in highly changing domain as news articles are. It is based on proposed effective representation of the article vector to allow real time computations over the dataset. In this paper we present also an overview of data extraction and vector construction method, similarity computation. This representation and similarity search is used in described content-based recommendation method.

Keywords: content-based recommendation, similarity search, vector representation

1 Introduction

The data amount on the web is serious problem for the common user. The existence of information is not so relevant, when there is no one who can access or find this information in acceptable time. One of the most relevant sources of information over the web is presented by news portals (nytimes.com, reuters.com, etc.). Most users prefer large renowned news metaportals. They include thousands of daily added news from the whole world and there is no chance to access them in a fast and comfortable way for every user. The only way to help the user is to personalize large amount of information and reduce it to an acceptable amount. There are several personalization systems in this domain nowadays [22], [20].

The main problem in the content-based personalization is effective and enough expressive representation of items (or articles). This is often done by means of text summarization [4] or keywords extraction [9]. These techniques are commonly used in English based systems and cannot be easily applied to other languages. Keywords extraction and summarization brings better results as the other methods but are more time consuming. These methods cannot represent non-text documents without modification.

Our method for similarity computation compresses article information value to short vectors, which are used for fast similarity computation over the specific articles time-window. This vector represents article in an effective way, so there is no need to store whole articles. Proposed method expects pre-processed article as an input and produces vector representation usually no longer than 30 words. Then these vectors can be easily used for similarity computations or we can use them in special structures for recommendation e.g. binary trees [24].

Fast similarity estimation plays the critical role in the high changing domains as news portals are. It is necessary to process new article as fast as possible and start to this article recommendation, because of the high information value degradation.

The paper is structured as follows. Section

2 Related work

There is lot of research in the field of text similarity. Most of method are used or developed for the plagiarism detection (MOSS, Ferret, Glatt etc.). Basically there are two types of the word similarity algorithms: statistical measure based on corpus and semantic distance based on hierarchical organization [18]. This can be extended by paraphrases identification, mainly for English language (MNLPG – Microsoft Natural Language Processing Group: Automatic Paraphrase identification) or by vector approximation [17] etc. Standard methods and their extensions are widely used like n-grams, longest common subsequence, measuring shared syntax or text "fingerprints" [10], [13].

A lot of systems used semantic nearness of documents [12], often based on WordNet dictionary [19], what is a significant problem when non English content is processed and fast computation is needed.

The text similarity focused on the news article is not so developed area. There are various projects in the field of text summarization [5] or news classification [15], latent semantic analysis or SOM [19] etc. Vector representation for text is a widely used representation for various techniques or systems [8], where it is not used in hierarchical or weighting purpose.

The similarity definition in recommendation systems is difficult task. We can define similarity based on news content (like plagiarism task), or based on "topic" or "affair". This is extremely important when recommendation list is created [25]. Our method respects every of these types. We can easily redefine our similarity with simple changing the weights for vectors parts and adjust it for various recommender methods.

Because of information overloading the recommendation is one of the actual research topics nowadays. We can find two basic concepts which are often mixed together to bring better results [2], [11]. Collaborative personalization accounts social element, where users are grouped into clusters, based on their activity (preferences, habits, etc.). Then we recommend items (e.g. news) which were read by other users from the cluster.

The main goal of content-based based personalization is to identify similar items – create "clusters" of items instead of users. Then we map user profiles to these

clusters. This type of recommendation is successful in well structured domains like movies, news etc. [14]. One of the problems in content-based personalization is effective and enough expressive items' representation.

There are several content-based news or articles recommendation systems. OTS [21] provides content-based and collaborative personalization based on association rules and users interest table. System works off-line because of large data amount. Users choose interesting articles and based on user profiles, recommended articles are found.

PURE [22] is designed to recommend medicine articles. There are aprox. 1000 new articles added daily. The user have to create own profile by defining interesting articles. Then system recommend new articles based on classification and EM algorithm, one time per day.

NewsMe [20] is adaptive recommendation system based on open user model. System monitors 81 RSS channels from 21 sources. The core of recommendation method represents Nearest Neighbor algorithm. Brusilovsky [20] has shown that explicit filled and open user model in the news domain brings usually worse results. Some systems have involved user location into recommendation systems, where recommendation list is created depending on user location [7].

3 Similarity search

The similarity search is one of the most important tasks in the content-based recommendation process. There are 3 basic steps necessary for similarity computation:

- Data preprocessing
- Representative vector construction
- Similarity estimation

3.1 Representative article vector construction

The main goal for fast similarity estimation is compact and high precision article vector representation. We propose a vector, which consists of 6 basic parts described in Table 1. We give short description for every part.

Table 1. Article Vector Representation

Title	TF of Title words in the content	Keywords	Category	Names/ Places	CLI

Title
Article vector comprises lemmatized words from article title. It consists of approximately 5 words (150 000 Slovak article dataset). We suggest that article title should be in most occurrences good describing attribute.

TF of title words in the content

We used TF – term frequency to estimate the article name confidence. If the article name is abstract and do not correspond to article content, we can easily discover this situation. Term frequency is computed as:

$$tf_i = \frac{n_i}{\sum_k n_k}$$

where tf_i is term frequency for term i (term from article title) and n_i is number of occurrences of term i in the document (article content) and $\sum_k n_k$ is the sum of numbers of all terms in document.

Keywords

The keywords part consists of 10 most relevant keywords. Many news portals store a list of keywords for every article. These are unfortunately usually at different abstraction level over various portals. This disadvantage can be solved by introducing own keywords list, which can be obtained using TF-IDF list calculated over the dataset (100 000 Slovak news articles from news portal SME.SK). We can reduce height dimension by removing any words except nouns and names.

Category

We include "tree-based" category vector with weights. This vector is constructed based on the portal specific category hierarchy (optional). The category is important for the similarity search, when articles from one category are marked as more similar. The weight for every category is estimated as:

```
n=1
For i=|Category| downto 0 do
  weight_i=1/n
  n=n*2
end
```

For example, we have 3 articles *A, B, C* and 4 categories *C1, C2, C3,* and *C4,* where the article *A* and *B* are stored in categories *C1, C2, C3* and the article *C* in categories *C1, C2, C4.* Then the vector for every article is represented in Table 2 and Fig. 1. As we can see, articles *A* and *B* would be more similar then the article *C.*

Table 2. Article Category Vector Representation

	C1	C2	C3	C4
A	1/4	1/2	1	-
B	1/4	1/2	1	-
C	1/4	1/2	-	1

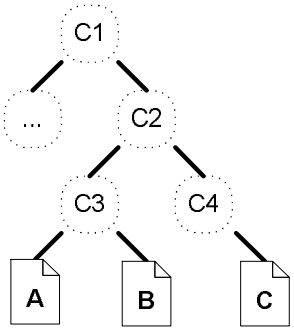


Fig. 1. Tree based article category organization.

Names/Places
We include names and places extracted from the article content. We purpose simply method for the name or place extraction. During the preprocessing process our method does not remove dots or uppercase letters. Then we can find names or places as the word starting with an upper letter and there is no dot before in the text stream. There are various name extractor systems for English language [6].

CLI
Coleman-Liau readability index (CLI) [3] provides information of the understandability of the text. This vector part is not important for standard similarity computation, but it is important in the results rearrangement. Our hypothesis is that the user wants to read articles of one similar level of understanding. This method is able to distinguish between two articles with similar title and different content ("Jaguar" – animal vs. car). CLI can be easily computed based on this formula:

$$CLI = 5.89 \times \left(\frac{characters}{words}\right) - 29.5 \times \left(\frac{sentences}{words}\right) - 15.8$$

Using proposed article representation, we can easily represent each article with length of vector no more as 30 items in most occurrences. Table 3 shows an example of the article represented in the proposed vector representation.

3.2 Similarity computation

For the similarity calculation we propose the cosine similarity [16] and Jaccard index computation, which is widely used in information retrieval tasks. The similarity of two articles ***A,B*** is computed as:

$$cosine\ similarity = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

$$Jaccard\ index = \frac{|A \cap B|}{|A \cup B|}$$

Table 3. The example vector representation of an article.

Vector part	Weights
Title	transplantácia_0.5 tvár_0.5
TF of title words in the content	transplantácia_0.0178571428571429 tvár_0.0714285714285714
Category	Sme.sk_0.5 PRESS_FOTO_1.0
Keywords	klinika_0.0357142857142857 povrch_0.0178571428571429 nos_0.0178571428571429 zub_0.0178571428571429 nerv_0.0178571428571429 svalstvo_0.0178571428571429 pacientka_0.0178571428571429 rozsah_0.0178571428571429
Names/Places	Cleveland_1
CLI	0.2543

Then, if the article vector length is approximately 30 items and the second article has the length 30, we make 60 comparisons in the worst case. Our method considers weight for every vector part, so there is a need to extend standard similarity computation:

$$cosine\ similarity = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} a_{ji}b_{ji}}{\sqrt{\sum_{j=1}^{m}\sum_{i=0}^{n} a_{ji}^2}\sqrt{\sum_{j=1}^{m}\sum_{i=0}^{n} b_{ji}^2}}$$

$$Jaccard\ index = \sum_{i=1}^{m} w_i \frac{|A_i \cap B_i|}{|A_i \cup B_i|}$$

where m is number of vector parts (6 in our method) and n is number of vector items.

Every part of the article vector has its own global weight. By changing this weight we can adjust calculated similarity and its precision. When we can calculate article similarity in a fast way, we can dynamically change these weights to obtain new results. For example if user reads all proposed similar articles, we can change the category part weight to zero and recalculate the similarity. As a result we obtain new set of similar articles from different categories. Fast similarity calculation is the advantage when new articles came too. We can find similar articles and start to recommend this new article in less than 2 seconds for cosine similarity and in led than 1 second for Jaccard index (window of 10 000 articles).

4 Recommendation

The input for recommendation method represents two lists:

- List of similar articles (obtained with described similarity search)
- User activity (time ordered)

We need clearly to identify users and articles they read. There is need to distinguish between before not recommended and recommended articles.

The first step of proposed recommendation method is to define number of articles to recommend (*N*). Based on this is list of recommended articles constructed from two sub lists:

- List of similar articles for before recommended and visited (*S*)
- List of similar articles for before not recommended and visited (*N*-*S*)

The ratio of these lists is dynamically computed as:

$$S = N\left(1 - \frac{Nr}{V}\right)$$

where *N* is number of articles to recommend, *Nr* represents not recommended but visited articles and *V* is number of visited articles together.

The second step represents the core of the recommendation method. During this step both sub lists are constructed:

```
foreach user activity log do
   visited = get visited articles list
   visitedRec = get visited and recomommended articles list

   foreach visited do
     if randomNum > probability
       listPart1 = get first non visited article from computed
                   similarity list
     else
       listPart1 = get random non visited article
     end
   end

   foreach visitedRec do
     listPart2 = get first non visited article from computed
                 similarity list
   end

   listToRecommend = listPart1[1..N] + listPart2[1..M]
end
```

When there is not enough user activity (does not mean “cold start”) we use random article assignment. In this manner we can easily react to user’s most recent

preferences. For the list of recommended and visited articles we also introduced a coincidence – where user obtains a random article to the recommendation list.

Fig. 2 presents an example of recommended content creation. We have a list of user activities where "-o" attribute indicates whether the article was or was not recommended. Based on this we obtain list of visited articles and list of visited and before recommended articles. Then when we want to recommend 4 articles (*N=4*), we will obtain ratio 3:1 for sub-lists (similar articles for visited and recommended, similar articles for visited and not recommended before).

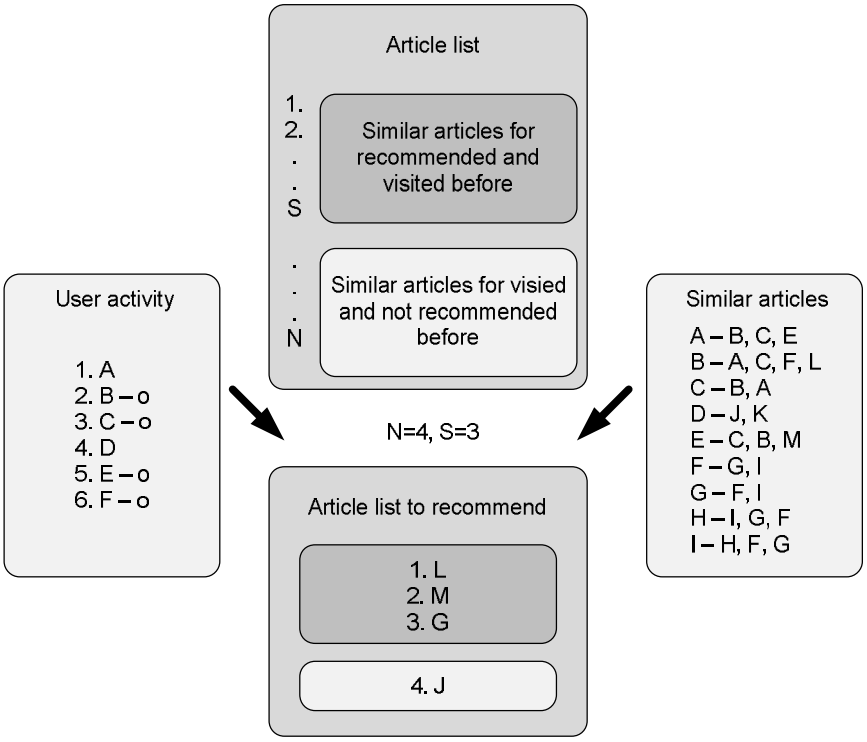


Fig. 2. Computation steps of proposed content-based recommendation method.

As we can see in our example, we have 4 visited and recommended articles *B, C, E, F.* We have found not visited article from the list of similar articles for every of these 4 articles. There is only one non visited similar article *L* for article *B.* This is repeated until the "before recommended" list is full. In the case when there do not exist non visited article in the similar list (article *C*), we skip this article, because the user saw all relevant articles for this "topic" already.

In our example there are 2 non recommended but visited articles, but there are no non visited articles for *A* – method will skip this article and will recommend first non visited for article *D.* In this manner we obtain a full list of 4 articles to recommend.

Dynamical computation of the ratio between sub list allows us to adapt for actual user activity and preferences. If the user does not like recommended articles and he uses other portal navigation, the size of first sub-list is decreasing while second part will increase respectively.

Our method doesn't need to stores "article age" for every recommended article. This number represents how long have been article recommended. If user does not visit this article for a defined time (number of recommendations) is this article automatically deleted from the recommendation list as not interesting.

User activity list consists of pair user identification – visited article. We use implicit user model representation, where there no need to involve users into various forms completing or need of logging etc.

5 Evaluation

Proposed methods were implemented within news recommendation system in project SMEFIIT [1]. We evaluated the similarity computation over 10 000 articles from the news portal SME.SK. For this window we are able to estimate the similarity in 0.8-1.5 seconds (2,6MHz Pentium, 4Gb RAM). The preprocessing takes approximately 10 minutes for the whole dataset. Then for the new article, when preprocessing is necessary, the whole computation process takes approximately 1-2 s.

5.1 Data preprocessing

Text pre-processing holds important role in the process of similarity search, because it can significantly reduce word space. This part of the process is high language depending. Our experiments are provided in the Slovak language, which is one of the most complicated languages (declension of nouns, verbs etc.). The architecture of the system is variable, so pre-processing for Slovak language can be easily replaced by other languages and their methods (e.g. Porter algorithm). For the speed of next computations, plays pre-processing a critical role. There is need to maximum reduction of article words dimensions.

The first task is to remove stop-words. We used static list, which can be replaced by TF-IDF output [16]. This method can identify commonly repeated words over the dataset.

As the main part of the pre-processing of Slovak language articles we used lemmatizing of the text. There is problem with algorithmic solution for this process, which can be solved by using dictionary of lemmas. The result we received is lemmatized (basic form) bag of words for every article.

It is necessary to note that we removed any punctuation except sentences ends. We use dots as a fast name or place indicator – when we check if there is a dot before an uppercase letter, and if not, it is probably personal name, or place etc. Name or Place extractor is one problem in information retrieval and is not main part of this article.

After keywords extraction we do not need whole article content anymore. We can safely delete all words except Title words obtained in the article content. Then for every processed article we have this list of words:

- Lemmatized article Title
- Lemmatized words from Content (which were included in the Title)
- 10 most relevant keywords
- List of Names and Places

Pre-processing methods we described above can significantly reduce number of words stored for every article up to 80%

5.2 Experimental results

The accuracy of our method was computed based on two datasets. The first one consists of 1 000 articles from news portal SME.SK. Every article from dataset has assigned at least one similar article. These similar articles were obtained from the portal, where there are usual two similar articles quoted in the article footer. These similar articles are chosen by the article author and this does not mean that there do not exist more similar articles.

The second dataset was manually annotated dataset, which consist of 100 articles marked in 5 levels of similarity, so we obtain 10 000 article pairs with similarity level. Our method computed list of similar articles for every article in the dataset. We compared these datasets to our method – list of similar articles computed by our method and list of similar articles obtained from one of two datasets. We calculated precision and recall and F-Score for every dataset and the method. Results were compared to standard text mining method TF-IDF as shown of Table 4.

Table 4. The Similarity Evaluation

Dataset	**SME.SK**		**Manually annotated dataset**		
Method	***Proposed method***	***TF-IDF***	***Proposed method***		***TF-IDF***
			Cosine similarity	***Jaccard index***	
Precision	0.165	0.091	0.700	0.843	0.511
Recall	0.202	0.117	0.816	0.818	0.587
F-Measure	**0.182**	0.102	**0.753**	**0.870**	0.546

As we can see there it is huge increase of precision and recall using proposed method, while computation process of our method was 2.46 times faster (cosine similarity) as when using TF-IDF method and 2 time faster (Jaccard index) as when using cosine similarity. Small score on the real “similar” dataset obtained directly SME.SK can be explained by not accurate and complete information on the websites. An author of a new article chooses none, one or two similar articles intuitively nowadays, and our method found probably more similar articles.

For the manually annotated dataset, we computed precision and recall with respect to the order of similar articles.

We also computed standard deviation based on similarity levels. We mapped computed similarity to five similarity levels used in our manually annotated dataset. The biggest standard deviation was 1.21 "similarity level", which is an acceptable rate in the field of news recommendation.

The verification of the recommendation method is based on synthetic tests (Table 5). The dataset – 3 days of user activity logs (05.03.2010 - 08.03.2010) from news portal SME.SK was divided into train and test period. We create recommendation list based on train period, then was the recommendation compared to real user activity from test period. This comparison was done for combination of section and category (Table 5 - Categories) and for articles too (Table 5 - Articles).

Table 5. Experimental results – synthetic tests.

	Train period [h]	Test period [h]	Precision		Recall		F1-Measure	
			Cos.	*Jacc.*	*Cos.*	*Jacc.*	*Cos.*	*Jacc.*
Categories	9	63	43.23	64.05	50.28	36.27	46.49	46.31
	24	48	40.26	63.26	50.94	37.44	44.97	47.04
	33	39	39.73	62.12	51.36	39.92	44.80	48.63
	48	24	38.02	59.91	59.95	40.23	46.53	48.14
Articles	9	63	1.43	1.83	0.84	0.77	1.06	1.08
	24	48	0.76	1.81	0.47	0.80	0.58	1.11
	33	39	0.67	1.68	0.49	0.85	0.57	1.13
	48	24	0.5	1.53	0.64	1.34	0.56	1.43

In our experiments brings cosine similarity better recall as Jaccard index, but in the other hand the precision was worse as Jaccard index. When we compare F1-Measure, Jaccard index bring better results for all test periods. When comparing articles, we can see very low score for every metrics. This we can explain by user stereotypes on news portal SME.SK. Nowadays on the title page there are list of most visited articles for various time periods. Most of users then visit these most visited articles.

Fig. 3. describes results of synthetic tests for proposed method in various test periods. As we can see precision of the proposed method increases over the time, because there are more articles visited by the users over the time period. This is the reason why recall decreases respectively.

We compared proposed method with TRecom [23] method (Table 6.). This method uses binary tree representation to allow content-based recommendation. Our method obtained better results for longer test periods (F-Measure), while we obtain better precision score for every period.

Table 6. Comparison of proposed method and TRecom method [23].

Test period		1 hour	4 hours	10 hours	24 hours	48 hours
TRecom	***Precision***	40	49	56	58	59
	Recall	71	60	44	32	25
	F1-Mesure	51	54	49	41	35
Proposed method	***Precision***	47	56	58	60	63
	Recall	56	52	45	40	37
	F1-Measure	51	54	51	48	47

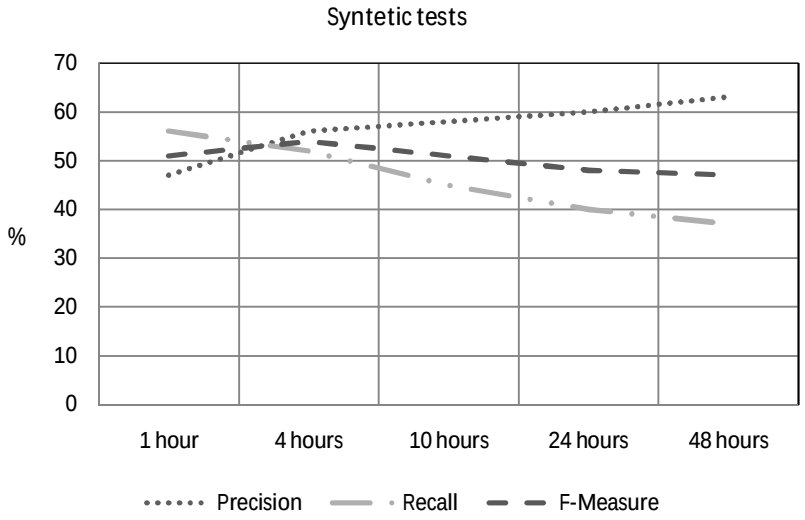


Fig. 3. Synthetic test – Precision, Recall, F-measure.

Weights of sub vectors for the similarity search were obtained based on evolutionary algorithm, where the fitness function was represented by F-Measure.

6 Conclusion

In this paper we provided overview of short and high representative article vectors, which can be used for similarity search and real content-based recommendation in large and dynamically changing datasets and domains. A key future of this method is short article representing vector. Based on these vectors can be computed similarity between articles (text or non-text content) in a fast way. Every article vector consists of 6 sub-vectors based on article part used for their construction. Every part has its own weight, which can be dynamically changed to rearrange similar articles list to enable fast personalization.

The weight was found using evolutionary algorithms for every vector part, to obtain the best result. As an example, using proposed representation brings 4 times better precision than using only article title, and at least 1.4 time better results as using only keywords. The category part improves precision only 1.15 time, but on the other hand it can be useful when “no similar” article is in the dataset.

Based on this computed similarity further recommendation is created. User preferences are collected implicitly via server’s logs. A recommended list consists of two sub lists, where the first one represents similar articles to the visited and already recommended. The second sub list is based on similar articles for visited but not recommended before. In this way we can easily adapt to user preferences. The ratio between these two sub-lists is dynamically computed.

We plan to make improvements on the precision and the recall, for example by using more sophisticated keywords extraction methods etc. Proposed vector representation is a promising method for the fast news similarity computation to allow real time recommendation.

Acknowledgment

This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0508/09.

References

[1] Barla, M., Kompan, M., Suchal, J., Vojtek, P., Zeleník, D., Bieliková, M., 2010. News recommendation. In Proc. of the 9th Znalosti.

[2] Bouras, C., Tsogkas, V., 2009. Personalization Mechanism for Delivering News Articles on the User's Desktop. In Proc. of the 2009 Fourth int. Conf. on internet and Web Applications and Services (May 24 - 28, 2009). ICIW. IEEE Computer Society, Washington, DC, pp. 157-162.

[3] Coleman, M., Liau, T. L., 1975. A computer readability formula designed for machine scoring, Journal of Applied Psychology, Vol. 60, pp. 283-284.

[4] Dakka, W., Gravano, L. 2007. Efficient summarization-aware search for online news articles. In Proc. of the 7th ACM/IEEE-CS Joint Conf. on Digital Libraries. JCDL '07. ACM, New York, NY, pp. 63-72.

[5] Díaz, A., Gervás, P., 2005. Personalisation in news delivery systems: Item summarization and multi-tier item selection using relevance feedback. Web Intelli. and Agent Sys. 3, 3 (Jul. 2005), 135-154.

[6] Finkel, J.,R., Grenager, T., Manning, Ch., 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proc. of the 43nd Annual Meeting of the Assoc. for Comp. Ling. (ACL), pp. 363-370.

[7] Chen, Ch., Hong, Ch., Chen, S., 2009. Intelligent Location-Based Mobile News Service System with Automatic News Summarization. International Conference on Environmental Science and Information Application Technology, pp. 527-530.

[8] Ingvaldsen, J. E., Gulla, J. A., Laegreid, T., Sandal, P. C., 2006. Financial News Mining: Monitoring Continuous Streams of Text. In Proc. of the 2006 IEEE/WIC/ACM int. Conf. on Web intelligence. Washington, DC, 321-324.

[9] Kou, H., Gardarin, G., 2009. Keywords Extraction, Document Similarity and Categorization, Tech.rep. No.2002/22, PRiSM Laboratory of Versailles Univ.

[10] Lukashenko, R., Graudina, V., Grundspenkis, J. 2007. Computer-based plagiarism detection methods and tools: an overview. In Proc. of the 2007 int. Conf. on Computer Systems and Technologies (Bulgaria, June 14 - 15, 2007). CompSysTech '07, vol. 285. ACM, New York, NY, 1-6.

[11] Melville, P., Mooney, R. J., Nagarajan, 2002. Content-boosted collaborative filtering for improved recommendations. In Proc. of 18th National Conf. on Artificial intelligence (Edmonton, Alberta, Canada). AAAI, Menlo Park, CA, pp. 187-192.

[12] Pandya, A., Bhattacharyya, P., 2005. Text Similarity Measurement Using Concept Representation of Texts. LNCS 2776, pp. 678-683.

[13] Parapar, J., Barreiro, Á., 2009. Evaluation of Text Clustering Algorithms with N-Gram-Based Document Fingerprints. In Proc. of the 31th European Conf. on IR Research on Advances in information Retrieval (Toulouse, France, April 06 - 09, 2009). Lecture Notes In Computer Science, vol. 5478. Springer-Verlag, Berlin, Heidelberg, 645-653.

[14] Pazzani, M., Billsus, D., 2007. Content-based recommendation systems, pp. 325-341.

[15] Ramos, J., 2000. Using TF-IDF to Determine Word Relevance in Document Queries. Tech. rep., Departament of Computer science. Rutgers University.

[16] Tata, S., Patel, J.M., 2007. Estimating the Selectivity of tf-idf based Cosine Similarity Predicates. SIGMOD Record, Vol. 36, pp. 7-12.

[17] Wang, Q., You, S., 2006. Fast Similarity Search for High-Dimensional Dataset. In Proc. of the Eighth IEEE int. Symposium on Multimedia (December 11 - 13, 2006). ISM. IEEE Computer Society, Washington, DC, 799-804.

[18] Wang, X., Ju, S., and Wu, S., 2008. Challenges in Chinese Text Similarity Research. In Proc. of the 2008 int. Symposiums on information Processing (May 23 - 25, 2008). ISIP. IEEE Computer Society, Washington, DC, 297-302.

[19] Wermter, S., Hung, C., 2002. Selforganizing classification on the Reuters news corpus. In Proc. of the 19th int. Conf. on Computer Ling. - Vol 1 (Taipei), pp. 1-7.

[20] Wongchokprasitti, C., Brusilovsky, P., 2007. Newsme: A case study for adaptive news systems with open user model. In: Proc. of Int. Conf. on Autonomic and Autonomus Systems, 2007. ICAS07. pp. 69.

[21] Wu, Y., Chen, Y., Chen, A. L., 2001. Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors. In Proc. of the 11th international Workshop on Research Issues in Data Engineering. RIDE. IEEE Computer Society, Washington, DC, 17.

[22] Yoneya, T., Mamitsuka, H., 2007. Pure: a pubmed article recommendation system based on content-based filtering. In Proc. of Int. Conf. on Genome Inf. 18,pp. 267-276.

[23] Zeleník, D., 2010. Relationships discovery and dynamic groups creation. Diplomová práca. FIIT STU.

[24] Zelenik, D., Bielikova, M., 2009. Dynamics in hierarchical classification of news. In Proc. of the 4th Work. on Intel. and Knowledge oriented Technologies (WIKT 2009), pp. 83-87.

[25] Ziegler, C., McNee, S. M., Konstan, J. A., Lausen, G., 2005. Improving recommendation lists through topic diversification. In Proc. of the 14th int. Conf. on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM, New York, NY, 22-32.

PRÍLOHA C – UKÁŽKA VZOROVÝCH DÁT

SME.SK

Vzorové dáta z portálu SME.SK

Časť zdrojového kódu HTML

```
<div id="contenth" class="art">
<div class="options"><a
href="http://www.sme.sk/diskusie/reaction_show.php?id_extern_theme=4832496&extern_type=sme
-clanok"><img src="/storm/imgs/toolbar/koment.gif" alt="diskutujte" border="0"
/><span>diskutujte</span></a> <a href="/clanok_tlac.asp?cl=4832496"><img
src="/storm/imgs/toolbar/tlac.gif" alt="vytlačiť" border="0" /><span>VYTLAČIŤ</span></a>
<a href="#" onclick="send_err_info('art', 4832496);"><img
src="/storm/imgs/toolbar/chyba.gif" alt="Upozornite na chybu" border="0"
/><span>UPOZORNITE NA CHYBU</span></a> <span class="sharespan">Pošlite:</span>
<a href="/clanok_tool.asp?t=odp&cl=4832496" class="sharelink"><img
src="/storm/imgs/toolbar/email.gif" title="pošlite e-mailom" border="0" /><span>E-
MAILOM</span></a> <a
href="javascript:location.href='http://www.facebook.com/share.php?u='+encodeURIComponent(l
ocation.href);" onclick="return art_poslite_click('fb')" class="sharelink"><img
src="/storm/imgs/toolbar/fb.gif" border="0" title="pridať na facebook" ><span>na
facebook</span></a>
<a
href="javascript:location.href='http://vybrali.sme.sk/sub.php?url='+encodeURIComponent(loc
ation.href);" class="sharelink" onclick="return art_poslite_click('vbsme');"><img
src="/storm/imgs/toolbar/doasdf_c.gif" title="pridať na vybrali.sme.sk" border="0"
/><span>VYBRALI.SME</span></a> | <a href="#" class="sharelink"
onmouseover="sharelinkOver()" onmouseout="sharelinkOut()"><span>ďalšie</span><img
src="/storm/imgs/toolbar/share_arrow.gif" alt="+" border="0"></a>
<div class="share_more" id="share_more" onmouseover="sharemoreOver();"
onmouseout="sharemoreOut();">
<ul>
<li><a href="http://delicious.com/save" onclick="return art_poslite_click('del')"
class="sharelink"><img src="/storm/imgs/toolbar/delicious.gif" title="pridať na delicious"
border="0"><span>delicious.com</span></a></li>
<li><a href="http://www.google.com/bookmarks" onclick="return art_poslite_click('g')"
class="sharelink"><img src="/storm/imgs/toolbar/google.gif" title="pridať na google"
border="0"><span>Google Bookmarks</span></a></li>
<li><a href="http://www.myspace.com/Modules/PostTo/Pages/" onclick="return
art_poslite_click('myspc')" class="sharelink"><img src="/storm/imgs/toolbar/myspc.gif"
title="pridať na myspace" border="0"><span>MySpace</span></a></li>
<li><a href="http://www.twitter.com/home?status=" onclick="return
art_poslite_click('twit')" class="sharelink"><img src="/storm/imgs/toolbar/twit.gif"
title="pridať na twitter" border="0"><span>Twitter</span></a></li>
</ul>
</div>
</div>
<h1>Transplantácia tváre</h1>
</div>
<div class="articlec col">
<div id="itext_content">
<div class="cl-fotorub">
<p><img alt="transplantacia-tvare_reuters.jpg"
src="http://i.sme.sk/cdata/6/48/4832496/transplantacia-tvare_reuters.jpg" /></p>
<p>Koláž fotografií Connie Culpovej, ktorá prežila výstrel do tváre. V Clevelandskej
klinike jej tím lekárov v decembri 2008 počas 22-hodinovej operácie transplantoval 80%
tváre. Včera klinika zverejnila výsledok operácií po zahojení (vpravo).</p>
<p>Lekárom sa podarilo zrekonštruovať nielen povrch tváre, ale aj jej funkčné časti: nos,
```

```
hornú sánku, zuby, nervy a svalstvo. Donorkou bola pacientka s mozgovou smrťou. Ide o prvú
operáciu takéhoto rozsahu na svete a o štvrtý prípad úspešnej transplantácie
tváre.</p></div>

</div>
<!-- eTarget ContextAd End -->

<p class="autor_line"><b>štvrtok 7. 5. 2009 9:00</b> | Reuters - Cleveland Clinic, trn<br
/><span class="copyr"><a href="#" onClick="st_openWindow('/footer/',
'PetitPress','width=650,height=550'); return false;">&copy 2009 Petit Press. Autorské
práva sú vyhradené a vykonáva ich vydavateľ. Spravodajská licencia
vyhradená.</a></span></p>
```


XML pre jeden článok


```
<clanok>
<id>4832496</id>
–<link>
http://s.sme.sk/r-odp/0/4832496/www.sme.sk/transplantacia-tvare.html
</link>
–<clanok_link>
http://www.sme.sk/c/4832496/transplantacia-tvare.html
</clanok_link>
<sekcia>SME Online</sekcia>
<rubrika>PRESS FOTO</rubrika>
<nadpis>Transplantácia tváre</nadpis>
<datum>7.5.2009 9:00:00</datum>
–<perex>
<img align="left" src="http://i.sme.sk/cdata/6/48/4832496/tn_transplantacia-
tvare_reu.jpg">
Koláž fotografií Connie Culpovej, ktorá prežila výstrel do tváre. V Clevelandskej klinike
jej tím lekárov v decembri 2008 počas 22-hodinovej operácie transplantovala 80% tváre.
</perex>
–<clanok>
<p><img alt="transplantacia-tvare_reuters.jpg"
src="http://i.sme.sk/cdata/6/48/4832496/transplantacia-tvare_reuters.jpg" /></p>
<p>Koláž fotografií Connie Culpovej, ktorá prežila výstrel do tváre. V Clevelandskej
klinike jej tím lekárov v decembri 2008 počas 22-hodinovej operácie transplantoval 80%
tváre. Včera klinika zverejnila výsledok operácií po zahojení (vpravo).</p>
<p>Lekárom sa podarilo zrekonštruovať nielen povrch tváre, ale aj jej funkčné časti: nos,
hornú sánku, zuby, nervy a svalstvo. Donorkou bola pacientka s mozgovou smrťou. Ide o prvú
operáciu takéhoto rozsahu na svete a o štvrtý prípad úspešnej transplantácie tváre.</p>
</clanok>
<autor>Reuters - Cleveland Clinic, trn</autor>
</clanok>
```


Predspracovaný článok

Nadpis	transplantácia tvár
Obsah	koláž fotografia Connie Culpovej prežiť výstrel tváre. Clevelandskej klinika tím lekár december 2008 počas 22 hodinový operácia transplantovať 80 tváre. včera klinika zverejniť výsledok operácia zahojení vpravo .Lekáro. podariť zrekonštruovať nielen povrch tvár funkčný časť nos horný sánka zub nerv svalstvo. Donorkou pacientka mozgový smrťou. ísť prvý operácia takýto rozsah svet štvrtý prípad úspešný transplantácia tváre.
Kategória	Sme.sk;PRESS FOTO;

Ukážka reprezentácie daného článku navrhnutou metódou

Časť vektora	Obsah časti (váhy)
Nadpis	transplantácia_0.5 tvár_0.5
TF slov z nadpisu v tele článku	transplantácia_0.0178571428571429 tvár_0.0714285714285714
Kategória	Sme.sk_0.5 PRESS_FOTO_1.0
Kľúčové slová	klinika_0.0357142857142857 povrch_0.0178571428571429 nos_0.0178571428571429 zub_0.0178571428571429 nerv_0.0178571428571429 svalstvo_0.0178571428571429 pacientka_0.0178571428571429 rozsah_0.0178571428571429
Index čitateľnosti CLI	0.2543

PRÍLOHA D – EXPERIMENT REUTERS

Pre čiastkové overenie metódy hľadania podobných článkov sme sa rozhodli využiť dáta poskytované prostredníctvom API (Application programming interface) na stránke http://prototype.nytimes.com. Stránka umožňuje vytvoriť si vlastný dopyt prostredníctvom grafického rozhrania, ktorý môže byť exportovaný do niekoľkých formátov (JSON, XML a pod.).

Pre získanie dátovej vzorky sme postupne prostredníctvom tohto rozhrania stiahli 100 článkov pre 10 kategórií:

- Breaking City News
- Editors Choice
- Health
- Environment
- Lifestyle
- Science
- Sport
- Tech
- TopNews
- WorldNews

Príklad takto stiahnutého článku vo formáte JSON sa nachádza v prílohe B. Samotná dátová vzorka je k dispozícii na priloženom dátovom médiu v časti Dáta.

Následne sme prostredníctvom nami navrhnutej metódy vypočítali podobnosť medzi týmito článkami a ku každému článku si uchovali 5 najpodobnejších článkov. Ako referenčnú množinu sme použili výsledky metódy, ktorá vypočítala podobné články na základe kľúčových slov priradených jednotlivým článkom. Táto podobnosť bola jednoducho vypočítaná kosínusovou podobnosťou.

Už pri výpočte tejto podobnosti sme si všimli, že kľúčové slová priradené jednotlivým článkom nie sú pravdepodobne dostatočne konkrétne ("Online Report text item", "Canada", "Central Banks", "Private/Government Aid", "Western Europe", a pod.), čo sa prejavilo aj vo výsledkoch experimentu.

Ďalším krokom bolo porovnanie týchto dvoch metód, ktoré prebehlo ako percentuálne vyjadrenie úspešnosti (koľko podobných článkov sa zhoduje v množine 5 vypočítaných článkov pre obe metódy).

Ako sme predpokladali, výsledok experimentu nebol uspokojivý, nakoľko naša metóda našla podobné články voči metóde s porovnaním kľúčových slov len v 32% prípadov. Využitie tejto vzorky sa ukázalo ako nevhodné pre naše potreby, nakoľko sa zameriavame na špecifickejšiu podobnosť, ako len tematickú, pri ktorej by bolo využitie tejto dátovej vzorky vhodné.

Vzorové dáta portál REUTERS.COM

Formát JSON


```
{
         "title": "Market Chatter -- Corporate finance press digest",
         "link": "http://www.reuters.com/article/allBreakingNews/idUKBNG21287620091112",
         "guid": "UKBNG21287620091112",
         "category": "allBreakingNews",
         "published": "2009-11-12T04:33:32Z",
         "description": " BANGALORE, Nov 12 (Reuters) - The following corporate\nfinance-
related stories were reported by media on Thursday:",
         "keywords": [
                  "United States of America"
                  ,"United Kingdom"
                  ,"Western Europe"
                  ,"Europe"
                  ,"Mergers and Acquisitions (including Changes of Ownership)"
                  ,"Private equity funds"
                  ,"Banks (industry group)"
                  ,"Financials"
                  ,"Business activities"
                  ,"Contract between 2 or more corporate entities"
                  ,"Fund management and asset allocation"
                  ,"Financial services - diversified"
                  ,"Marine port services"
                  ,"Utilities"
                  ,"Australia"
                  ,"Asia"
                  ,"Company News item"
                  ,"Economic news, EC, business/financial pages"

                      ],
                  "tickers":[
                           "2887.TW"
                           ,"NB.UL"
                           ,"nl;GBP"
                           ,"sg;NYS"
                           ,"tw;2887"
                           ,"us;BCKB"
                           ,"us;NMR"
                           ,"us;NPNY"
                           ,"us;NRSC"
                           ,"us;NYUK"
                           ,"us;RBS"
                           ,"jp;9101"
                           ,"jp;8604"
                           ,"8604.T"
                           ,"au;BBI"
                           ,"BBI.AX"
                           ,"de;NSE"
                           ,"de;NYKF"
                           ,"de;PIG"
                           ,"de;RYSX"
                           ,"gb;NMR"
                           ,"gb;NYQ"
                           ,"gb;RBS"
                           ,"us;RBSP"
                           ,"RBS.L"

                             ],
                  "body": {
                           "type": "html",
                           "lang": "en",
                           "text": "<p> BANGALORE, Nov 12 (Reuters) - The following
corporate\nfinance-related stories were reported by media on Thursday:</p>\n\n<p> * A
Royal Bank of Scotland (RBS.L: <a href=\"/stocks/quote?symbol=RBS.L\">Quote</a>, <a
href=\"/stocks/companyProfile?symbol=RBS.L\">Profile</a>, <a
href=\"/stocks/researchReports?symbol=RBS.L\">Research</a>) consortium will make a\nA$1.5
billion ($1.4 billion) bid for Australian investment firm\nBabcock &amp;amp; Brown
Infrastructure (BBI) (BBI.AX: <a href=\"/stocks/quote?symbol=BBI.AX\">Quote</a>, <a
href=\"/stocks/companyProfile?symbol=BBI.AX\">Profile</a>, <a
href=\"/stocks/researchReports?symbol=BBI.AX\">Research</a>), taking aim at a\nrival
Canadian proposal, the Australian Financial Review said.\n[ID:nSYD473744]</p>\n\n<p> *
Newbridge Capital [NB.UL], investor George Soros, and\nNomura Holdings (8604.T: <a
href=\"/stocks/quote?symbol=8604.T\">Quote</a>, <a
href=\"/stocks/companyProfile?symbol=8604.T\">Profile</a>, <a
```

```
href=\"/stocks/researchReports?symbol=8604.T\">Research</a>) have been approached by
Chinese banks\ninterested in their stakes in Taishin Financial (2887.TW: <a
href=\"/stocks/quote?symbol=2887.TW\">Quote</a>, <a
href=\"/stocks/companyProfile?symbol=2887.TW\">Profile</a>, <a
href=\"/stocks/researchReports?symbol=2887.TW\">Research</a>) in a\nsale that could be
worth T$30 billion ($923 million), the\nCommercial Times reported.
[ID:nTP329108]</p>\n\n<p> * Japanese shipping firm Nippon Yusen (9101.T: <a
href=\"/stocks/quote?symbol=9101.T\">Quote</a>, <a
href=\"/stocks/companyProfile?symbol=9101.T\">Profile</a>, <a
href=\"/stocks/researchReports?symbol=9101.T\">Research</a>) will likely\nraise about 150
billion yen ($1.7 billion) in its first public\nshare offering in 40 years, the Nikkei
business daily reported.\n[ID:nT369500]\n (Compiled by Shivani Singh)\n\n</p>\n\n"
                    },
                    "sources": [
                    ]
          }
```

PRÍLOHA E – TECHNICKÁ DOKUMENTÁCIA

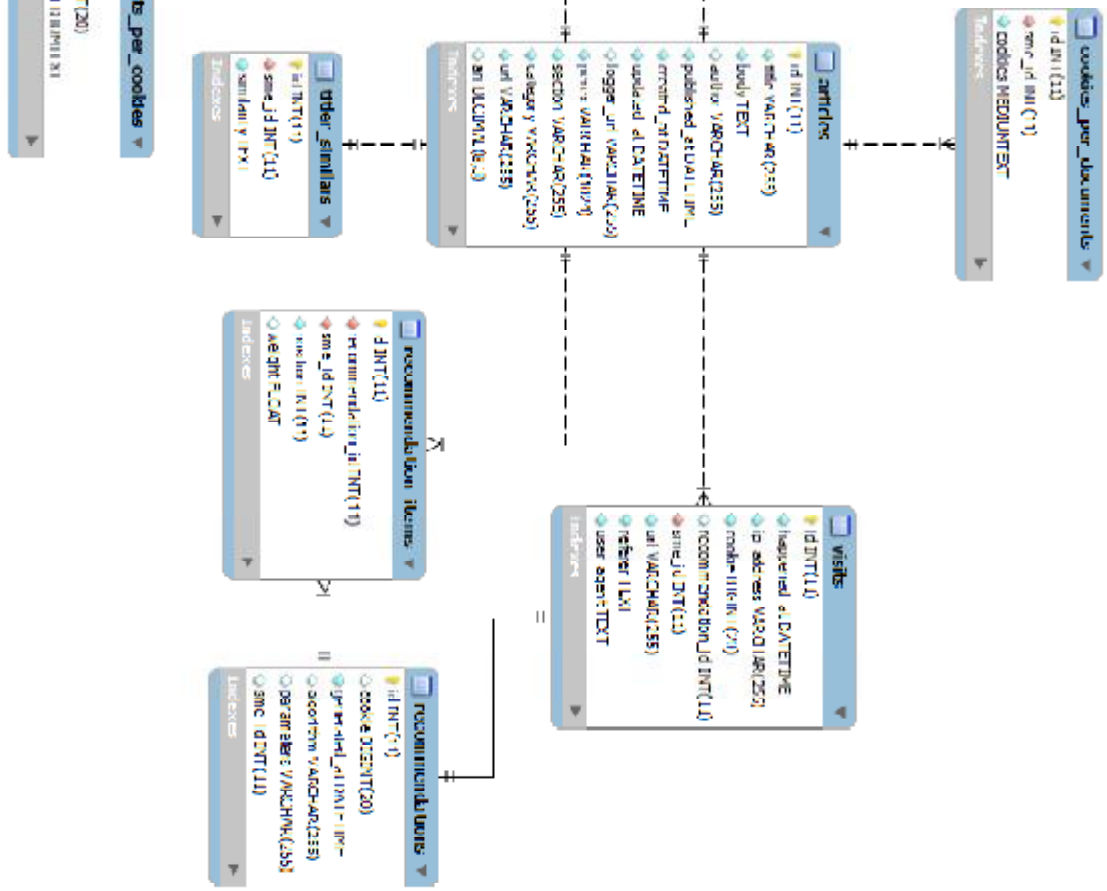
Ukážka zdrojového kódu metódy odporúčania článkov

```
require 'algorithms/title_recommender/Preprocessing.rb'
require 'algorithms/title_recommender/Vectors.rb'
require 'algorithms/title_recommender/Similarity.rb'

class Title_Recommender

  def recommend(visit)
    documents = GetRecommendation(visit.cookie)
    recommendation = Recommendation.new
    documents.each do |id|
      recommendation.items.build(:sme_id => id)
    end
    return recommendation
  end

  def GetSimilarArticle(id)#vrati hash podobnych clankov s hodnotou
                           podobnosti
    res=Hash.new()
    item=Titler_similar.find(:first, :conditions => ["sme_id = ?", id])
    if(item==nil)
      return res
    else
      res=decode(item.similarity)
      return res
    end
  end

  def GetRecommendation(cookie,num=10)#odporucanie pre cookie vrati NUM
                                      pocet odporucanych clankov

    visited=Array.new()
    visitedRecom=Array.new()
    recommend=Array.new()
    similar=0
    data=Visit.find(:all,
                    :conditions => ["cookie = ?", cookie],
                    :select => 'DISTINCT sme_id, recommendation_id',
                    :order => "happened_at DESC", :limit => 50)
    data.each do |row|
      count+=1
    if(row.recommendation_id.to_s!="0")#precital som to
      visitedRecom.push(row.sme_id)
      else
        visited.push(row.sme_id)
      end
    end

    #ak neprecital ziadne tak koncime s random , ak nie tak vypocitam
     pomer kolko akych odporucit
    if(count==0)
      recommend=get_random(num,cookie)
       return recommend
      else
        similar=(num*(1-(visited.size.to_f/count.to_f))).round
```

```
      end

    #k precitanym a odporucenym najdem podobne, ktore este neboli
     precitane a ked ich je malo tak doplnim o random
    visitedRecom.shuffle.each do |item|
      if(recommend.size==similar)
        break
      end
      list=GetSimilarArticle(item).keys
      list.each do |similarA|
        if(visitedRecom.include?(similarA)==false &&
           visited.include?(similarA)==false &&
           recommend.include?(similarA)==false)
          recommend.push(similarA)
          break
        end
      end
    end

    if(recommend.size<similar)
      get_random(similar-recommend.size,cookie).each do |random|
        recommend.push(random)
      end
    end

   #koniec odporucenych

    #k neodporucenym najdem podobne, ktore neboli precitane a ked ich
     je malo tak doplnim random
    visited.shuffle.each do |item|
     if(recommend.size==num)
        break
     end
     list=GetSimilarArticle(item).keys
      list.each do |similarA|
        if(visitedRecom.include?(similarA)==false &&
           visited.include?(similarA)==false &&
           recommend.include?(similarA)==false)
         recommend.push(similarA)
         break
       end
     end
    end


    if(recommend.size<num)
     get_random(num-recommend.size,cookie).each do |random|
        recommend.push(random)
      end
   end
   #koniec neodporucenych
   return recommend
  end

  private
  def get_random(num,cookie) #vráti NUM posledne citanych clankov okrem
                              clankov ktore citala COOKIE
    temp=Array.new()
    data=Visit.find(:all ,
                    :conditions =>["cookie != ?", cookie],
                    :select =>'DISTINCT sme_id',
```

```
                    :limit => num,
                    :order =>'happened_at desc')
    data.each do |item|
      temp.push(item.sme_id)
    end
    return temp
  end

  def encode(data)
    result=""
    data.each_pair {|key,value| result+=key+";"+value.to_s+"*"}
    return result
  end

  def decode(data)#prevedie zapis z db do spracovatelnej formy -
                  vektory
    pairs=data.split(/\*/)
    result=Hash.new()
    pairs.each do |row|
      temp=row.split(';')
      result[temp.at(0).to_s]=temp.at(1).to_f
    end
    return result
  end
end
```

Dátový model



PRÍLOHA F – OBSAH PRILOŽENÉHO MÉDIA

/Clanky	Článok ECWEB, IIT.SRC 2010, Návrh článku do časopisu
/Data	Vzorky dát použité pri riešení (SME, REUTERS, kľúčové slová, stop slová, lematizátor)
/Extrakcia dat	Metódy pre extrakciu dát (XML –SME, JSON-REUTERS, TXT Anotovaná vzorka), Nástroj použitý na anotáciu vlastnej dátovej vzorky
/Klucove slova	Metóda pre identifikáciu podstatných mien – kľúčových slov (slovnik.juls.savba.sk)
/Metoda odporucania	Metóda pre odporúčanie v rámci portálu SME.SK
/Podobnost	Metóda pre tvorbu reprezentatívnych vektorov a výpočet podobnosti
/Praca	Text diplomovej práce v elektronickej podobe
/Predspracovanie	Samostatná metóda pre predspracovanie textu
/RDoc	Technická dokumentácia