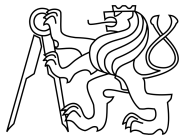
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Aplikace pro doporučování novinových zpráv v reálném čase
Student:	Bc. Christián Golian
Vedoucí:	Ing. Jaroslav Kuchař, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Cílem práce je navrhnout a implementovat doporučovací systém v Javě pro novinové zprávy na zpravodajských serverech. Zaměřte se na dílčí typické aspekty, jako je aktuálnost či „krátká životnost". Součástí práce je také splnění požadavků reálných doporučení (škálovatelnost, časové limity na doporučení, ...).
- Seznamte se s oblastí doporučovacích systémů.
- Proveďte rešerši přístupů a algoritmů pro doporučování novinových zpráv.
- Seznamte se s rozhraním Open Recommendation Platform od Plista (ORP).
- Navrhněte a implementujte aplikaci v Javě pro doporučování zpráv vhodnou pro využití v reálných aplikacích. Musí brát zřetel na škálovatelnost, spolehlivost a časové omezení pro poskytnutí jednotlivých doporučení.
- Proveďte experimenty na dostupné off-line platformě NewsREEL Replay.
- Hotový systém zapojte do on-line platformy NewsREEL Live a vyhodnoťte kvalitu implementovaného řešení.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 3. ledna 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra softwarového inženýrství

Diplomová práca

Aplikace pro doporučování novinových zpráv v reálném čase

Bc. Christián Golian

Vedúci práce: Ing. Jaroslav Kuchař, Ph.D.

5. mája 2017

Poďakovanie

Rád by som poďakoval vedúcemu mojej diplomovej práce Ing. Jaroslavovi Kuchařovi, Ph. D. za poskytnutie serveru, časté konzultácie a cenné rady.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 5. mája 2017 .....

Odkaz na túto prácu

Golian, Christián. *Aplikace pro doporučování novinových zpráv v reálném čase.* Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Táto práca sa zaoberá odporúčaním novinových správ a návrhom a implementáciou odporúčacieho systému. Program, ktorý sa zúčastnil výzvy CLEF NewsREEL využíva na odporúčanie asociačné pravidlá. Teoretická časť práce definuje pojmy súvisiace s odporúčacími systémami a popisuje špecifiká odporúčania novinových správ. Praktická časť sa venuje návrhu a implementácii programu *rule-recommender*. Experimentálna časť práce sa snaží vyvodiť závery z prevedených meraní.

Klíčová slova odporúčacie systémy, CLEF NewsREEL, asociačné pravidlá, odporúčanie novinových správ, Open Recommendation Platform

Abstract

This thesis deals with news recommendation and design and implementation of a recommender system. The resulting program which took part in CLEF NewsREEL challenge uses association rules as a basis for recommendation. The theoretical part of this thesis defines terms connected with recommendation systems and describes the specifics of news recommendation. The practical part deals with design and implementation of recommender system *rule-recommender*. Conclusions based on measurements made are drawn in the experimental part of this thesis.

Keywords recommender systems, CLEF NewsREEL, association rules, news recommendation, Open Recommendation Platform

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

"Many receive advice, only the wise profit from it." – Harper Lee

Motivácia

Odporúčacie systémy sa v posledných rokoch stali neoddeliteľnou súčasťou online sveta.

Tieto systémy sa líšia v položkách, ktoré odporúčajú. Zákaznikom spoločností ako Amazon alebo eBay odporúčajú tovar, ktorý by ich mohol zaujímať. Divákom seriálov a čitateľom kníh odporúčajú služby Netflix[1] a Goodreads[2] ďalšie filmy, seriály a knihy, ktoré by ich mohli zaujať. Odporúčajú dovolenky, poistenia a ďalšie finančné služby, hotely, reštaurácie a dokonca aj partnerov.

Čítanie novinových správ na internete sa takisto stalo neodmysliteľnou súčasťou života veľkej časti ľudí. Dostupných článkov, komentárov a správ je ale príliš veľa. Systémy, ktoré odporúčajú novinové správy, by mali túto situáciu čitateľovi zľahčiť a odporučiť mu veci, ktoré by ho mohli zaujímať.

Odporúčanie novinových správ sa od klasického odporúčania tovaru líši. Systém, ktorý ma takýto obsah odporúčať, nemá k dispozícii niektoré informácie, ktorými odporúčacie systémy v iných oblastiach disponujú. Nemá napríklad k dispozícii profil užívateľa, ako to majú služby, ktoré odporúčajú knihy alebo filmy. Na rozdiel od klasického odporúčania, staré položky a články zaujímajú len malú časť užívateľov. Vďaka týmto a ďalším špecifikám postupy a algoritmy z klasického odporúčania nie sú vždy najúčinnejšie.

Táto práca sa odporúčaním novinových správ zaoberá. Zaoberá sa špecifikami odporúčania, skúma algoritmy a postupy, ktoré už boli vyskúšané. Skúma aj možnosti vyhodnocovania takýchto odporúčaní. V neposlednom rade sa implementáciou jedného prístupu odporúčania zaoberá.

[1] `https://www.netflix.com/`

[2] `https://www.goodreads.com/`

Ciele

Výsledkom tejto diplomovej práce je odporúčací systém novinových článkov *rule-recommender* napísaný v Jave. Mojim prvým cieľom je ukázať ako a v čom sa odporúčanie novinových článkov líši od klasických odporúčacích systémov.

Tento program sa zároveň zúčastnil výzvy CLEF NewsREEL[3]. Mojim hlavným cieľom preto bolo aby mal tento program čo najlepšie výsledky v porovaní s ostatnými. Na meranie výsledkov sa používajú viaceré metriky, medzi inými *click-through rate* a *response rate*. Z tohto dôvodu sú v tejto práci analyzované a na základe týchto metrík porovnané existujúce riešenia.

Okrem metrík existujú aj odlišné spôsoby vyhodnocovania. Ďalším cieľom je preto popísať tieto spôsoby a podrobne popísať metódy vyhodnocovania použité vo výzve CLEF NewsREEL.

Program využíva na získavanie odporúčaní asociačné pravidlá. Tieto pravidlá sú ziskavané z informácii o položkách, ktoré obsahujú jednotlivé správy. Jedným z cieľov je preto konkrétne ukázať ako asociačné pravidlá pri odporúčaní používam a vysvetliť rozhodnutia pri návrhu.

Posledným cieľom je experimentovať s navrhnutým riešením a ukázať ako zmena niektorých parametrov môže zmeniť výsledky algoritmu.

Štruktúra

Túto prácu možno rozdeliť do troch častí - teoretickej, praktickej a experimentálnej.

Teoretická časť sa zaoberá teóriou odporúčania správ a odporúčaním novinových správ. Sú v nej predstavené pojmy a koncepty, ktoré sú využité neskôr.

- V kapitole *Odporúčacie systémy* sú priblížené pojmy ako odporúčacie systémy, užívateľ alebo položka. Okrem nich sú tu uvedené aj ciele týchto systémov a je tu aj priblížených niekoľko známych modelov. Nakoniec sa venuje porovnávaniu a vyhodnocovaniu odporúčacích systémov.

- Kapitola *Odporúčanie novinových správ* je štruktúrou podobná predošlej kapitole. Pojednáva hlavne o výzvach a špecifikách, ktoré odporúčanie novinových správ prináša. Týmto výzvam je venovaná celá sekcia. Je v nej popísaných niekoľko známych algoritmov, ktoré sa na odporúčanie používajú. V závere sa opäť venuje metrikám a možnostiam vyhodnocovania.

- Analýzou existujúcich riešení sa zaoberá kapitola *Analýza existujúcich riešení*. Existujúcami riešeniami sú príspevky do výzvy CLEF NewsREEL. Na základe dostupných metrík sú v nej jednotlivé riešenia porovnané.

[3] `http://www.clef-newsreel.org/`

Praktická časť sa zaoberá architektúrou a implementáciou algoritmu na odporúčanie novinových správ založenom na asociačných pravidlách. Jedná sa o program *rule-recommender* napísaný v Jave.

- V kapitole *Open Recommendation Platform* je popísaná platforma Open Recommendation Platform, ktorá sa v CLEF NewsREEL používa. Je tu popísaná jej architektúra a je tu popísaný aj ORP protokol posielania správ.
- Kapitola *Implementácia* sa zaoberá implementáciou programu *rule-recommender*. V nej je popísaný okrem samotného programu aj základný algoritmus, ktorý ORP používa.

Experimentálna časť tejto práce sa zaoberá meraniami programu a interpretáciou výsledkov meraní.

- Výsledkami online a offline vyhodnocovania sa zaoberá kapitola *Meranie*.

Časť I

Teoretická časť

Kapitola 1

Odporúčacie systémy

Táto kapitola by mala slúžiť ako teoretický úvod do problematiky odporúčacích systémov. Sú v nej definované odporúčacie systémy a pojmy s nimi súvisiace. Takisto sú v nej definované typy týchto systémov a ich použitie. V tejto kapitole sú popísané aj základné algoritmy, ktoré tieto systémy používajú, ich výhody a nevýhody. Ako posledné sú tu spomenuté metriky a spôsoby vyhodnocovania spojené s odporúčacími systémami.

1.1 Defínicia

Odporúčacie systémy poskytujú návrhy vecí, ktoré by mohli byť zaujímavé pre užívateľa[1]. Používajú sa napríklad na získanie odpovedí na otázky[2] ako:

- *Ktorú knihu by som si mal kúpiť?*
- *Ktorú stránku by som mal navštíviť?*
- *Ktorý článok by ma mohol zaujímať?*

Pri týchto systémoch teda existujú tri typy objektov: *veci*, *užívatelia* a *transakcie*.

Veci sú objekty, ktoré su odporúčané. Môžeme ich charakterizovať ich zložitosťou alebo hodnotou.

Hodnota veci je kladná, ak bola vec pre užívateľa nejakým spôsobom užitočná, a záporná ak bola zbytočná.

Čo sa týka zložitosti, je jednoduchšie odporučiť knihu alebo novinový článok ako dovolenku alebo poistenie, ktoré majú jednak vyššiu hodnotu a musia byť viac "šité na mieru".

Tieto veci musia mať nejaké vlastnosti, na základe ktorých ich systém odporúča. Pri filme sú to napríklad herci alebo žáner, pri novinových článkoch ich kategória.

Užívatelia sú osoby, ktoré dostávajú odporúčania. Odporúčacie systémy využívajú pri odporúčaní okrem iného informácie o nich (napríklad ich vek, polohu, príjem, vzdelanie alebo pohlavie).

Na základe týchto informácii môžu byť o užívateľoch vytvárané profily, ktoré sú kľúčové pri odporúčaní založenom na kolaboratívnom filtrovaní.

Odporúčacie systémy takisto môžu využiť zvyklosti užívateľov, napríklad stránky, ktoré navštevujú.

Transakcie zachycujú interakcie medzi vecami a užívateľmi. Táto interakcia môže obsahovať aj kontext a spätnú väzbu užívateľa. Najpopulárnejším príkladom transakcie je hodnotenie veci užívateľom.

1.1.1 Hodnotenia

Odporúčania by sa mali užívateľovi "páčiť" - mali by to byť veci, ktoré ho zaujímajú, ktoré má rád. Je teda potrebná nejaká metodika ako rozdeliť veci na tie, ktoré sú pre užívateľa zaujímavé a tie, ktoré nie sú. Existuje viacero typov hodnotení:

- **Unárne**, kde užívateľ môže vyjadriť, že sa mu vec páči, ale nemá ako vyjadriť, že sa mu nepáči. Príkladom je odporúčanie novinových správ, kde nekliknutie na odporučený odkaz nemusí nutne znamenať, že sa mu obsah nepáči.

- **Binárne**, v ktorom užívateľ má dve možnosti: *páči (sa mi)/nepáči (sa mi)*. Nevýhodou je, že užívateľ nemá možnosť hodnotiť neutrálne a je *nútený* hodnotiť kladne/záporne.

- **Ordinálne**, v ktorom užívateľ vyberá z možností ako napríklad (*Silne nesúhlasím, Nesúhlasím, Neviem, Súhlasim, Silne súhlasím*).[3]

- **Intervalové**, kde užívateľ vyberá z intervalu čísel. Toto používa napríklad Goodreads[4], kde užívateľ hodnotí prečítanú knihu jednou až piatimi hviezdičkami. Podľa Goodreads jedna hviezdička reprezentuje *"nepáčilo sa mi to"*, dve *"bolo to ok"*, tri *"páčilo sa mi to"* , štyri *"veľmi sa mi to páčilo"*, päť *"bolo to úžasné"*. Jedna možnosť teda reprezentuje negatívne hodnotenie, jedna neutrálne a tri pozitívne. Takéto hodnotenie je teda *nevyvážené*.

1.2 Ciele odporúčacích systémov

Odporúčacie systémy dnes používa mnoho spoločností, spomeňme napríklad: Amazon[5], Goodreads alebo Netflix[6].

[4] `https://www.goodreads.com/`
[5] `https://www.amazon.com/`
[6] `https://www.netflix.com/`

Spoločnosti ich používajú na rôzne účely a z rôznych dôvodov. Amazon používa odporúčacie systémy na odporúčanie tovaru, Netflix odporúča filmy, sociálne siete používaju tieto systémy na odporúčanie priateľov. Časté dôvody sú: *zväčšujú spokojnosť zákaznikov, umožňujú im predať špecifickejšie veci, zvyššujú vernosť zákaznikov, umožňujú im lepšie pochopiť požiadavky zákaznikov*[1]. Užívateľom tieto systémy umožňujú: *nájsť dobré veci, vyjadriť sa, pomôcť ostatným, ovplyvniť ostatných.*

Hlavným dôvodom používania zo strany spoločností je samozrejme *zväčšenie tržieb*, ale aby bol dosiahnutý tento cieľ, odporúčacie systémy by sa mali mať tieto vlastnosti:

- **Relevancia** Odporúčací systém by mal odporúčať relevantné veci - teda veci, ktoré sú pre užívateľa zaujímavé.
- **Novosť** Veci odporučené systémom by mali byť nové v tom zmysle, že ich užívateľ ešte nevidel alebo sa s nimi nestretol.
- **Prekvapenie** Táto vlastnosť súvisi s predchádzajucou v tom, že odporučené veci by mali byť pre užívateľa trochu nečakané. V systéme, ktorý odporúča filmy, by bola prekvapivým odporučením pre užívateľa, ktorý obľubuje komédie, dráma.
- **Diverzita** Ak systémy často odporúčajú množinu vecí, ktoré sú si navzájom veľmi podobné, existuje možnosť, že sa užívateľovi nebude páčiť ani jedna z nich. Čím je väčšia diverzita, tým sa pravdepodobnosť toho, že ho nezaujme ani jedna z vecí, znižuje.

1.3 Modely odporúčacích systémov

Odporúčacie systémy teda pracujú z informáciami dvojho druhu: vlastnosti veci a užívateľov a hodnotenia užívateľov. Existuje niekoľko modelov odporúčacích systémov:

- **Kolaboratívne filtrovanie** Modely založené na kolaboratívnom filtrovaní poskytujú odporúčania na základe hodnotení užívateľov. Užívateľ dostáva odporučené veci, ktoré sa páčili užívateľom s podobným "vkusom".

  Ich výhodou je, že môžme doporučiť aj veci, pri ktorých nepoznáme ich vlastnosti. Výhodou je taktiež aj odporúčanie vecí s rôznym obsahom - a tým dostávame prekvapivé odporúčania. Nevýhodou týchto modelov je to, že nemôžu poskytovať odporúčania pre nové veci, ktorým chýbaju hodnotenia ostatných užívateľov.
- **Filtrovanie založené na obsahu** Filtrovanie založené na obsahu odporučí užívateľovi veci, ktoré zdieľajú vlastnosti s vecami, ktoré hodnotil v minulosti kladne.

Jednou výhodou oproti kolaboratívnemu filtrovaniu je skutočnosť, že pri filtrovaním založenom na obsahu nepotrebujeme hodnotenia iných užívateľov a tým pádom získavame istú mieru nezávislosti. Systémy založené na filtrovaní na obsahu takisto nemajú problém odporúčať veci, ktoré nemajú alebo majú málo hodnotení iných užívateľov. Filtrovanie založené na obsahu je okrem toho transparentnejšie, keďže môžme uviesť vlastnosti veci na základe ktorých filtrujeme.

Nevýhodou je, že noví užívatelia a užívatelia s málo hodnoteniami budú dostávať nepresné odporúčania. Odporúčacie systémy ale často nedisponujú údajmi o všetkých vlastnostiach veci, napríklad z dôvodu ochrany osobných údajov. Tieto systémy potom musia pracovať s obmedzeným obsahom. Navyše, pri niektorých veciach je problém obsah popísať. Okrem toho, tieto modely odporúčajú veci s rovnakým alebo podobným obsahom a tieto odporúčania sú tým pádom málo prekvapivé.

- **Filtrovanie založené na znalostiach** Filtrovanie založené na znalostiach sa líši od filtrovania založeného na obsahu v tom, že odporučí veci na základe vlastností, ktoré užívateľ chce[3].

  Systémy založené na filtrovaní založenom na znalostiach majú význam pri odporúčaní vecí, ktoré nie sú kupované/konzumované až tak často alebo nemá zmysel vyberať na základe predošlých hodnotení. Príkladom takýchto vecí sú napríklad dovolenky, autá, nehnuteľnosti alebo finančné služby.

  Pri tomto filtrovaní sa teda nepoužívajú hodnotenia užívateľov priamo na odporúčania. Hľadá sa podobnosť medzi požiadavkami užívateľa a existujúcimi vecami. Toto sa deje za pomoci *báz znalostí*.

- **Hybridné modely** Ako vyplýva z názvu, tieto modely kombinujú predošlé prístupy. Ich cieľom je zvýšiť presnosť odporúčaní[2].

  Môžu fungovať tak, že za behu si budú vyberať jeden algoritmus na základe nejakého skóre alebo môžu výsledky z viacerých predošlých prístupov kombinovať.

1.4 Porovnávanie odporúčacích systémov

1.4.1 Metriky

Aby sme odporúčacie systémy mohli porovnať, musíme ich vedieť vyhodnotiť. Rozmery v ktorých ich vyhodnocujeme, možno rozdeliť do štyroch kategórii podľa aspektov systému, na ktorý sú zamerané[4]:

1.4.1.1 Zamerané na odporúčania

- **Správnosť**, to znamená ako veľmi sa približujú dané odporúčania ku "správnym" odporúčaniam. Správne odporúčania môžu byť množina preddefinovaných odporúčaní alebo odporúčania s nejakými vlastnosťami.

  Pri hodnotení správnosti treba počítať s tým, že veľká väčšina vecí nie je ohodnotená alebo je ohodnotená malým počtom užívateľov a nachádza sa v tzv. *dlhom chvoste*.

  Správnosť sa často počíta buď ako štvorcová ochýlka (RMSE) alebo priemerná absolútna chyba (MAE) medzi predpokladanými a skutočnými hodnoteniami užívateľa.

- **Pokrytie**, u ktorého rozlišujeme pokrytie *užívateľov* a pokrytie *vecí*. To, z akého percentá vecí vie systém poskytnúť odporúčania je pokrytie vecí.

  S pokrytím súvisí aj *cold start* problém. Tento problém nastáva pri užívateľoch alebo veciach, o ktorých ešte systém nemá dostatok informácii. Príkladom teda môže byť práve registrovaný užívateľ, ktorý ešte neohodnotil žiadne filmy alebo nová položka, ktorú nikto nestihol ohodnotiť. Takíto užívatelia a veci potom nie su systémom pokryté.

  Pokrytie vecí môžme merať ako podiel počtu vecí, ktoré môže systém odporučiť a počtu vsetkých vecí. Systém môže takisto poskytovať odporúčania len istej skupine užívateľov, a to by malo odrážať pokrytie užívateľov. Pokrytie užívateľov môžme merať podobne ako pokrytie vecí, a to ako podiel počtu užívateľov ktorým môže systém poskytnúť odporúčanie a počtu vsetkých užívateľov.

- **Diverzita** Táto vlastnosť už bola spomenutá v sekcii 1.2. Diverzita úzko súvisí s pokrytím, pretože vyššia diverzita zaisťuje väčšie pokrytie. Diverzita súvisi ale aj z prekvapením, kde takisto vyššia diverzita vecí zaistí väčšie prekvapenie.

  Pre skupinu odporúčaných vecí by sme diverzitu spočitali ako priemernú podobnosť medzi všetkými pármi vecí. Čím by bola táto hodnota nižšia, tým by bola diverzita vyššia.

- **Dôvera** vyjadruje ako veľmi si je systém istý svojimi odporúčaniami. Na základe dôvery môže hybridný systém vybrať vhodnejší algoritmus. Systém môže dôveru merať napríklad ako pravdepodobnosť toho, že odporúčanie je správne.

1.4.1.2 Zamerané na užívateľa

- **Dôveryhodnosť** vyjadruje mieru ako veľmi užívateľ "verí" odporúčaniam - považuje ich za "dobré" alebo "užitočné"[5]. Jednou z možností

ako získať ich dôveru, je odporúčať im veci, ktoré poznajú[5]. Inou možnosťou je vysvetliť im, ako systém funguje[6].

Dôveryhodnosť môžme merať len na základe spätnej väzby od užívateľov - napríklad ich necháme vyplniť dotazník.

- **Novosť** bola už vysvetlená v sekcii 1.2. Novosť môžme merať počítaním populárných vecí ktoré už boli odporučené. V takomto prístupe vychádzame z toho, že populárne veci užívateľ bude poznať.

- **Prekvapenie** bolo už takisto spomenuté v sekcii 1.2.

  Môžme ju merať je podobne ako u dôveryhodnosti získavať spätnú väzbu od užívateľov. Iný spôsob merania je použiť ako základ jednoduchý systém, o ktorom vieme, že odporúča zrejmé veci. Miera prekvapenia sa potom bude rovnať počtu doporučených vecí, ktoré by neodporučil náš jednoduchý systém.

- **Úžitok** je hodnota ktorú užívateľ odporučením získa. Táto hodnota môže byť preddefinovaná užívateľom alebo systémom ako nejaká úžitková funkcia.

- **Riziko** je spojené s jednotlivými odporúčaniami. Odporúčanie filmov alebo novinových správ je podstatne menej rizikové ako odporučenie refaktorovacieho kroku v IDE, ktorý môže mať väčšie následky[5].

1.4.1.3 Zamerané na systém

- **Robustnosť** je schopnosť tolerovať zlé a nesprávne informácie od užívateľov. Odporúčacie systémy, ktoré používajú e-shopy musia počítať so situáciou ako napríklad autor snažiaci sa zviditeľniť svoju knihu udeľovaním jej pozitívnych hodnotení. Inou sitáciou je udeľovanie negatívnych recenzií konkurencii.

  Riešenim je merať hodnoty pred a po poskytnutí informácie od užívateľa a merať ako sa menia predikcie[7].

- **Miera učenia** je rýchlosť ktorou sa odporúčací systém učí nové informácie a aktualizuje zoznam odporúčaných vecí. Systém s vysokou mierou učenia sa dokáže rýchlo (v krátkom čase) adaptovať na zmenu preferencií užívateľa.

  Mieru učenia môžeme merať ako čas za ktorý sa systém vráti k svojej pôvodnej presnosti po zmene preferencií užívateľa.

- **Škálovateľnosť** je dôležitá vlastnosť väčšiny systémov. Ide o schopnosť systému zvládnuť väčší počet užívateľov alebo väčší počet dát.

  Škálovateľnosť meriame ako počet odporúčaní za sekundu[8].

- **Stabilita** súvisí s dôverou užívateľov v systém. Stabilný systém vracia odporúčania ktoré sú konzistentné a tým pomáha zväčšovať ich dôveru v systém. Užívatelia nedôverujú nestabilnému systému, pretože jeho odporúčania sa často menia.

  Stabilita sa meria podobne ako robustnosť porovnaním v dvoch časových bodoch: pred a po pridaní nových hodnotení.

- **Súkromie** súvisí s faktom, že užívateľ poskytne odporúčaciemu systému dobrovoľne dáta. Tieto dáta by nemali byť poskytnuté tretej strane a osobné preferencie užívateľa by mali ostať skryté.

  Niektoré štúdie ale dospeli k tomu, že narušenie súkromia by sme mali minimalizovať, ale úplne súkromie nie je reálne[9].

  Meranie miery súkromia je náročná úloha, jedným prístupom je merať koľko informácií uniklo k tretej strane. Iným prístupom je diferenciálne súkromie[10], podľa ktorého by výstup výpočtu nemal obsahovať informácie, z ktorých by bolo možné odvodiť prítomnosť alebo neprítomnosť záznamu vo vstupe výpočtu.

1.4.1.4 Zamerané na dodávku

- **Použiteľnosť** systému nastane, ak je systém efektívny, účinný a uspokojivý pre užívateľa. Tieto vlastnosti užívateľ pocíti cez užívateľske rozhranie. Toto rozhranie by sa teda malo držať základnych princípov tvorby rozhrania.

  Použiteľnosť systému môžeme jedine merať pomocou spätnej väzby od užívateľov.

- **Preferencia užívateľa** je vlastnosť, ktorá by mala užívateľovi umožniť systém si do istej miery nakonfigurovať. Mal by teda mať možnosť zvoliť si formu v akej budú výsledky prezentované, filtrovať ich, jednotlivým veciam rôzne váhy alebo dokonca zvoliť si odporúčací algoritmus.

  Metrikou je opäť spätná väzba od užívateľov.

1.4.2 Vyhodnocovanie odporúčacích systémov

Metriky spomenuté v predošlej sekcii 1.4 môžme rozdeliť na kvalitatívne a kvantitatívne. Príkladom kvantitatívnej metriky je napríklad správnosť, pokrytie alebo diverzita. Príkladom kvalitatívnej metriky je dôveryhodnosť. Aby sme mohli zmerať všetky tieto metriky čo najpresnejšie, je nutné použiť viacero rôznych prístupov.

1.4.2.1 Offline vyhodnocovanie

Offline vyhodnocovanie pracuje s historickými dátami - už zozbieranými dátami o užívateľoch, veciach a transkaciách medzi nimi. Predpokadáme teda, že výsledky získané takouto simuláciou sa priblížia ku výsledkom, ktoré dosiahne odporúčací systém po nasadení.

Keďže nepotrebujeme užívateľov, takéto vyhodnocovanie nie je tak nákladné ako online verzia. Takéto vyhodnocovanie takisto setom môžme použiť ako jednotný štandard merania pre iné algoritmy. Príkladom takýchto voľne dostupných datasetov je napríklad MovieLens[7], ktorý obsahuje 20 miliónov rôznych typov hodnotení vyše 27 000 filmov 138 000 užívateľmi. Ešte známejším príkladom je Netflix Prize[8], ktorý je súčasťou súťaže spoločnosti Neflix, ktorá udeľovala cenu za najlepší algoritmus pre odporúčanie filmov založený na kolaboratívnom filtrovaní.

Nevýhodou offline vyhodnocovania je nemožnosť získať informácie o správaní užívateľov a zmene správania užívateľov počas behu systému. Offline vyhodnocovanie sa preto používa skôr na vyradenie nesprávnych algoritmov a na experimenty s hodnotami parametrov algoritmov.

1.4.2.2 Online vyhodnocovanie

Online vyhodnocovanie je použité ak chceme sledovať zmenu v správaní užívateľov. Je realizované presmerovaním zlomku internetového prenosu z portálu na odporúčací systém. Keďže v tomto druhu testovania reálni užívatelia vykonávajú reálne transakcie online, považuje sa za najdôveryhodnejšie a je často používané v reálnom svete[11]

Príkladom online vyhodnocovania je Open Recommendation Platform[9], ktorej sa budem neskôr v tejto práci podrobnejšie venovať.

Existujú rôzne metriky, ktoré môže online vyhodnocovanie merať, často sa používa click-through rate. Online vyhodnocovanie je užitočné pri veľkom počte užívateľov/hodnotení. Pri testovaní nového systému teda z výsledkov nemôžme vyvodzovať všeobecné závery.

1.4.2.3 Spätná väzba od užívateľov

Spätná väzba od užívateľov je jediný spôsob ako získať informácie o metrikách ako použiteľnosť alebo dôveryhodnosť.

Väčšinou sa získava tak, že je vybraná testovacia skupina užívateľov, a každému z nich sa pridelí niekoľko úloh, ktoré má vykonať. Pri tomto teste sa sleduje chovanie užívateľov alebo sa meria sa čas, ktorý potrebujú na jednotlivé úlohy. Po takomto teste môžu byť testovaní užívatelia požiadaní, aby vyplnili dotazník obsahujúci otázky, ktoré nie je možné získať pozorovaním.

[7] `https://grouplens.org/datasets/movielens/`
[8] `http://netflixprize.com`
[9] `https://orp.plista.com/login`

Tento druh testovania je veľmi známy rozšírený aj v iných oblastiach (napríklad testovanie užívateľského rozhrania), preto nebudem bližšie popisovať jeho výhody a nevýhody.

1.5 Zhrnutie

Táto kapitola sa zaoberala oblasťou odporúčacích systémov.

- Boli v nej definované odporúčacie systémy a s nimi súvisiace veci ako užívatelia, položky a transakcie.
- Popísala ich ciele ako prekvapenie, novosť alebo diverzita.
- Vysvetlila základné modely týchto systémov ako kolaboratívne filtrovanie alebo odporúčanie založené na obsahu.
- Boli v nej uvedené metriky, ktorými možno odporúčacie systémy porovnávať
- Boli popísané možnosti vyhodnocovania odporúčacích systémov.

Kapitola 2

Odporúčanie novinových správ

Táto kapitola sa zaoberá odporúčaním novinovým správ a mala byť zrkadliť predošlú kapitolu 1.

Na začiatku sa zaoberá sa výzvami a špecifikami odporúčania takéhoto typu správ ako je rozptyl, dynamika, popularita a kontext. Sú v nej popísané jednotlivé skupiny ľudí, ktoré su zainteresované v odporúčaní novinových správ

Potom sa podobne ako v predošlej kapitole zaoberá algoritmami, ktoré sa na odporúčanie novinových správ používajú a neboli ešte spomenuté v predošlej kapitole. Trochu podrobnejšie sa venuje asociačným pravidlám, keďže práve na nich je postavený môj algoritmus.

Ďalej sú tu zhrnuté online a offline možnosti vyhodnocovania odporúčaní. Súvisiac s tým je popísaná odporúčacia platforma *Open Recommendation Platform*, ktorú som použil v praktickej časti.

2.1 Špecifiká odporúčania novinových správ

Čítanie novinových správ v tlačenej verzii je dnes čoraz zriedkavejšie a užívatelia sa významne presúvajú na internet[12].

Dôvodov je niekoľko. Prvým je fakt, že tlačené média nemôžu ponúknuť také aktuálne správy ako internetové. Ďalším je skutočnosť, že užívateľ má možnosť vybrať si čo chce čítať, a koľko o tom chce čítať. Toto v tlačenej verzii novín jednoducho nie je možné, keďže editori pracujú s obmedzeným miestom. Ďalšou výhodou internetu oproti printovým médiám, je fakt, že na internete ma užívateľ často možnosť čítať správy z viacerých zdrojov bez toho aby za to platil.

Všetky tieto výhody internetových médií majú za následok prebytok informácií. Užívateľ sa často nemôže rozhodnúť čo si vybrať. V tom by mu mali pomôcť odporúčacie systémy. Výstupom odporúčacieho systému je množina článkov, ktoré by mohli užívateľa zaujať.

Na systém, ktorý odporúča novinové správy sú však kladené iné požiadavky ako na klasické odporúčacie systémy.

Systémy, ktoré odporúčajú novinové správy narážajú na štyri výzvy: *rozptyl*, *popularita*, *dynamika* a *kontext*[12].

2.1.1 Rozptyl

Rozptyl (po angl. *sparsity*) je pomer počtu interakcií, ktoré sme zachytili a celkového počtu potenciálnych interakcií.

Pomer interakcií, ktoré sme zachytili vyjadrujeme pomocou funkcie *Int(u, i)*, ktorá nadobúda 1 ak sme pozorovali interakciu medzi užívateľom u a vecou i. Vo zvyšných prípadoch je hodnota tejto funkcie 0.

Počet všetkých potenciálnych interakcií počítame ako súčin počtu všetkých užívateľov a všetkých vecí. Celý vzorec je vidno na 2.1:

$$sparsity = 1 - \frac{\sum_{u \in U} \sum_{i \in I} Int(u, i)}{|U||I|} \tag{2.1}$$

Odporúčacie systémy takmer vždy operujú z malým počtom zachytených interakcií. Väčšina datasetov obsahuje len zlomok potenciálnych interakcií. Pri odporúčaní novinových správ však je tento pomer ešte výraznejší. Data set Netflix zachycuje 1 z 89 potenciálnych interakcií. Datové sety dvoch vybraných novinových portálov zachycujú 1 zo 66623 interakcií[12].

2.1.2 Popularita

Veci, ktoré sú populárne (známe) vzbudzujú dôveru užívateľa v systém. Ako už bolo spomínané v predošlej kapitole 1.4.1.1, relatívne malá časť vecí sa teší veľkej popularite užívateľov a väčšina vecí obdrží len zlomok interakcií. To platí nie len pre knihy, hudbu alebo filmy ale aj pre novinové články.

Príčiny tohoto fenoménu sú psychologické - veľa užívateľov chce vidieť blockbuster, prečítať si bestseller alebo si vypočuť hit.

2.1.3 Dynamika

Každý systém musí počítať s tým, že sa budú doňho pridávať nové veci. Je potrebné si uvedomiť, že novinové články, ktorých len jednotlivé portály vyprodukujú stovky tisíc za rok, sa produkujú s oveľa vyššou frekvenciou ako napríklad knihy, filmy alebo pesničky.

Ďalším faktom je, že s vecami ako napríklad filmy interagujú užívatelia po dlhšiu dobu. Pre datový set *MovieLens* je medián rozdielu medzi prvou a poslednou interakciou 2254 dní. U novinových článkov viac ako polovica interakcií prebehne do 24 hodín po ich zverejnení. Pre populárne novinové články je to dokonca ešte väčší pomer[12].

Poslednou vecou, ktorú treba brať na zreteľ je skutočnosť, že u ostatných vecí ich užívatelia často spotrebujú niekoľko krát. Veľa ľudí si pozrie viac krát svoj obľúbený film alebo prečíta svoju obľúbenú knihu. V prípade piesní je

to taktiež veľmi časté. U novinových článkov to nemožno povedať. Podstatné teda je, že u novinových článkov ich *relevancia* klesá časom.

2.1.4 Kontext

Spotreba novinových správ záleži na viacerých faktoroch, medzi inými čas, deň v týždni, poloha, zariadenie alebo nálada. Príkladom jedného možného kontextu by boli užívatelia, ktorí čítajú novinové správy ráno na svojich tabletoch.

Väčšina interakcií na stolných počítačoch prebehne počas pracovnej doby. Interakcie na mobiloch a tabletoch zas prebehnú počas večerov a víkendov. Nočná doba je z hľadiska interakcií zanedbateľná.

Tieto informácie nám umožňujú napríklad počítať s väčšou záťažou na mobilné zariadenia cez víkend. Na základe toho môžme vybrať vhodný odporúčací algoritmus.

2.2 Zúčastnené strany

V odporúčaní novinových správ je ďalej zainteresovaných niekoľko strán. Sú tu tak ako u ostatných typov systémov *užívatelia*, ktorí očakávajú relevantné články, ktoré ich budú zaujímať.

Ďalej sú tu *poskytovatelia obsahu*, ktorí vyberajú články, ktoré sa majú zobrazovať. Ich záujmom je aby algoritmus reprezentoval všetky relevantné články a užívateľ mal možnosť sa o nich dozvedieť.

Ďalšou skupinou sú *ľudia, ktorí platia za reklamy* a chcú aby si návštevníci ich reklamu všimli, prípadne kúpili ich produkt. Väčšinou sa platí za klik a najrelevantnejšiou metrikou pre nich je click-through rate.

Nakoniec sú tu *vydavatelia*, ktorí chcú aby užívatelia strávili čo najviac času na ich stránkach, keďže sa tým zvyšuje šancu, že si všimnú reklamu. Tým sa zvyšujú aj ich príjmy.

2.3 Algoritmy odporúčania novinových správ

Na odporúčanie sa všeobecne môžu používať alebo používajú algoritmy spomenuté v sekcii 1.3 ako kolaboratívne filtrovanie alebo odporúčanie založené na obsahu. V tejto sekcii by som ale chcel spomenúť algoritmy špecifické pre tento typ odporúčania.

2.3.1 Najpopulárnejšie správy

Tento algoritmus odporúča správy na základe ich popularity.

Algoritmus berie do úvahy obmedzený životný cyklus správ spomenutý v sekcii 1.2, a preto vracia najpopulárnejšie správy z preddefinovaného časového

okna. Pred tým ako vráti jednotlivé správy, skontroluje či ich už daný užívateľ nečítal.

Najpopulárnejšie správy je ešte možné rôznymi spôsobmi filtrovať[13]. Napríklad môžme vrátiť správy z rovnakej kategórie, akú ma článok, na ktorom sa užívateľ práve nachádza, môžme vrátiť správy, ktoré boli čítané v rovnakom dni v týždni ako aktuálny článok alebo môžme vrátiť správy čítané užívateľmi z rovnakou polohou akú má cieľový užívateľ.

2.3.2 Najnovšie správy

Tento jednoduchý algoritmus správy zoradí ich chronologicky podľa doby ich vytvorenia. Najnovšie články budú odporúčané ako prvé.

Jeho výhodou je, že môže odporučiť užívateľovi správy s informáciami, ktoré ešte nepoznal. Slabinou je, že neberie ohľad na preferencie užívateľa a kontext.

2.3.3 Náhodné správy

Toto je ďalší jednoduchý algoritmus, ktorý vráti náhodne vybrané správy. Pri jeho použití hrozí riziko, že odporučí veci, ktoré nebudú užívateľa vôbec zaujímať.[10] Na druhej strane, môže odporúčať veci, ktoré nie sú nové ani populárne a napriek tomu budú *prekvapivo* zaujímavé pre užívateľa.

2.3.4 Odporúčanie založené na pravidlách

2.3.4.1 Definícia

Jednou z možností, ako odporúčať novinové správy, je odporúčať na základe asociačných pravidiel. Asociačné pravidlá sú jedna z techník dolovania dát a nachádza uplatnenie mimo iné aj pri odporúčaní novinových správ. **Asociačné pravidlo** je definované v [14] na príklade supermarketu takto:

> *Nech $I = \{I_1, I_2, \ldots, I_m\}$ je množina binárnych atribútov nazývaných položky. Nech $D = \{t_1, t_2, \ldots, t_m\}$ je množina transakcií nazývaná databáza. Každá transakcia v D má jedinečný identifikátor a obsahuje podmnožinu položiek z I.*
>
> *Pravidlo je definované ako implikácia v tvare $X \Longrightarrow Y$, kde X, Y sú podmnožinami I, $X, Y \subseteq I$ a majú prázdny prienik $X \cap Y$.*
>
> *X sa nazýva aj antecedent alebo ľavá strana pravidla, a Y sa nazýva konsekvent alebo pravá strana pravidla.*

[10] `https://orp.plista.com/login`

ID	položky
1	chlieb, jablko, syr
2	jablko, mlieko
3	chlieb, jablko, syr, mlieko
4	šunka, syr
5	jablko, banán

Tabuľka 2.1: Príklad množiny transakcií

2.3.4.2 Metriky

S asociačnými pravidlami súvisia aj metriky ako podpora, dôvera alebo lift, ktoré nám umožňujú pravidlá porovnávať a filtrovať ich.

Podpora množiny položiek X (supp(X)) je definovaná ako podiel počtu transakcií v databázi, ktoré obsahujú X a všetkých transakcií.

Dôvera pravidla $X \Longrightarrow Y$ 2.2 je definovaná ako podiel podpory zjednotenia X a Y $X \cup Y$ a podpory X.

$$conf(X \Longrightarrow Y) = supp(X \cup Y)/supp(X) \tag{2.2}$$

Dôveru je možné interpretovať aj ako pravdepodobnosť nájdenia pravej strany pravidla v transakciách ak tieto transakcie obsahujú ľavú stranu. Pri vyberaní pravidiel ich filtrujeme s hraničnými hodnotami minimálnej dôvery a minimálnej podpory. Ak ich je stále priveľa môžme ich zoradiť na základe ďalších metrík.

Lift pravidla $X \Longrightarrow Y$ 2.3 je definovaný ako podiel podpory zjednotenia X a Y $X \cup Y$ a súčinu podpory X a podpory Y.

$$lift(X \Longrightarrow Y) = supp(X \cup Y)/supp(X)supp(Y) \tag{2.3}$$

Ako už bolo spomenuté asociačné pravidlá a metriky súvisiace s nimi sa často vysvetľujú na príklade supermarketu[14][15]. Príkladom databázy alebo množiny transakcií by mohla byť tabuľka 2.1, ktorá obsahuje záznamy nákupov v supermarkete. Túto databázu tvorí päť transakcií a šesť položiek: chlieb, jablko, syr, mlieko, šunka, banán.

Príkladom pravidla by mohlo byť $\{chlieb, jablko\} \Longrightarrow \{syr\}$, inak povedané: ak si niekto kúpi chlieb a jablko tak si kúpi aj syr. Ľavú stranu tohto pravidla tvorí $\{chlieb, jablko\}$ a pravú stranu tvorí $\{syr\}$.

Antecedent pravidla $\{chlieb, jablko\}$ sa objavuje v dvoch transakciách z piatich, má teda podporu $2/5 = 0.4$, to je 40%. Podpora $\{chlieb, jablko, syr\}$ je takisto 0.4, a tým pádom dôvera je $0.4/0.4 = 1$. Lift tohto pravidla by bol $0.4/0.4 \times 0.6$, čo je po zaokrúhlení hore približne 1.7.

2.3.4.3 Algoritmus

V prípade novinových správ by teda položkami pravidiel mohli byť na ľavej strane informácie o interakcii, ktorá práve prebieha/prebehla, a pravú stranu pravidla by mohol tvoriť identifikátor článku.

Informácie o interakcii, ktoré by pre nás mohli byť zaujímavé z hľadiska odporúčania, by mohli byť informácie o článku ako jeho kategória, jeho pozícia na stránke alebo kľúčové slová, ale aj informácie o užívateľovi - jeho identifikátor, jeho poloha alebo zariadenie na ktorom článok číta.

Jednotlivé prichádzajúce interakcie by sme si teda nejakým spôsobom ukladali. Z týchto interakcií by sme si priebežne vytvárali pravidlá použitím nejakého algoritmu na dolovanie - napríklad apriori. V tomto bode je nutné si uvedomiť, že všetky hodnoty nemusia byť vždy k dispozícii a to bude mať samozrejme dopad na vytvorené pravidlá. Okrem toho, pri ťažení dát by mali byť hodnoty numerické a diskrétne. Táto skutočnosť by mohla spôsobiť problémy v prípade, že informácie o interkacii by neboli reprezentované číselnými identifikátormi. Vyťažené pravidlá by sme si v pamäti zoradili napríklad podľa dôvery a podpory.

Pri obdržaní žiadosti o odporučenie, by sme prešli všetky existujúce pravidlá v pamäti a pre každé pravidlo porovnávali údaje zo žiadosti s ľavou stranou pravidla. Ak by sa ľavá strana zhodovala, je možné že užívateľa by položka v pravej strane mohla zaujať a odporučili by sme ju.

Bol to práve podobný algoritmus založený na asociačných pravidlách, ktorý som sa rozhodol použiť v súťaži CLEF NewsREEL. To aké položky som si vybral na pravé a ľavé strany pravidiel, aké hodnoty podpory a dôvery som použil pri dolovaní dát je podrobnejšie popísané v časti II.

2.4 Vyhodnocovanie odporúčania novinových správ

Systémy odporúčajúce novinové správy môžme takisto vyhodnocovať online a offline. Súťaž CLEF NewsREEL[11][16], ktorej som sa zúčastnil poskytovala obidve možnosti vyhodnocovania.

2.4.1 Online

2.4.1.1 Open Recommendation Platform

Spoločnosť plista[12] prevádzkuje službu, ktorá tisícom webstránok zabezpečuje odporúčanie obsahu a reklám[12]. Keďže webové portály musia svoje odporúčania poskytnúť okamžite, je potrebné návštevy a žiadosti o odporúčania spracovať v reálnom čase.

[11] http://www.clef-newsreel.org/tasks/

[12] https://www.plista.com

Open Recommendation Platform bola spoločnosťou plista vytvorená v roku 2010 a umožňuje svojim používateľom interagovať s reálnymi užívateľmi v reálnom čase[17]. Týmto sa jedná o jeden z prvých, ak nie o úplne prvý systém, ktorý niečo takéto umožňuje[18].

Obsah, s ktorými pracujú používatelia ORP pochádza z niekoľkých rôznych portálov, ktoré sa líšia obsahom. Patria sem portály zaoberajúce sa spravodajstvom, športom, podnikaním alebo novinkami zo sveta IT[18]. Tento obsah poskytujú vydavatelia malých, stredných a veľkých novinových portálov.

Interakcia medzi ORP a používateľom prebieha v dvoch fázach. V prvej z nich si návštevník novinového portálu načíta novinovú stránku a požiada o odporúčania. V druhej server účastníka obdrží žiadosť a vráti zoznam odporúčaných položiek.

Komunikácia medzi účastnikmi a ORP prebieha pomocou HTTP API posielaním správ v JSON formáte[19]. Tieto správy sú buď žiadosti o odporúčania, informácie o tom, že užívateľ klikol na nejaký článok, informácie o tom, že užívateľ si prečítal nejaký článok alebo chybové správy. Správy obsahujúce informácie o článku, ktorý užívateľ prečítal obsahujú jeho identifikátor, kľúčove slová, kategóriu článku ale aj typ zariadenia na ktorom bol prečítaný, informácie o polohe užívateľa alebo vek užívateľa. Podrobnejšie sa tomu venuje sekcia 4.2.4.

Algoritmy jednotlivých účastnikov vyberá náhodne ORP princípom *multi-armed bandit*. Týmto umožní účastnikom zastaviť tok dát, vymeniť alebo opraviť algoritmus a znovu ho spustiť. Ak by napriek tomu algoritmy účastnikov zlyhaly, obsahuje ORP aj záložný algoritmus, ktorý v takejto situácii odporúčania poskytne.

Keďže sa jedná o reálny svet, prichádzajú z ORP aj určité obmedzenia. Prvým je maximálny čas na poskytnutie odporúčania, ktorý je 100 milisekúnd. S tým súvisí metrika *miera odozvy* 2.4, ktorá dáva do pomeru počet správ na ktoré stihol systém odpovedať a počet všetkých správ.

$$\text{response rate} = \frac{\text{počet odpovedí}}{\text{počet poslaných správ}} \tag{2.4}$$

Ak je tento interval prekročený, plista nedokáže odporučenie zahrnúť do zobrazovanej stránky. Druhým obmedzením je, že systém by mal vraciať odporúčania na obsah od toho istého vydávateľa. Tretím obmedzením je, že systém by nemal vraciať medzi odporúčaniami články, ktoré sú na tzv. *blackliste*. ORP zbiera hlavne tri údaje. Najpodstatnejším je metrika *click-through rate* 2.5, čo je pomer odporučených článkov, na ktoré užívateľ klikol ku všetkým žiadostiam o odporúčanie.

$$\text{ctr} = \frac{\text{počet kliknutých odporúčaní}}{\text{počet odporučených článkov}} \tag{2.5}$$

Okrem toho zbiera aj počet návštev.

2.4.2 Offline

Jednou z možností ako vyhodnocovať offline je zbierať údaje o užívateľoch a ich interakciách. Takto by mohol byť zhotovený záznam z nejakého časového úseku (napríklad týždeň alebo mesiac). S pomocou tohto záznamu by sme potom mohli nasimulovať reálnu prevádzku.

Takúto simuláciu umožňuje napríklad framework *Idomaar*[20], ktorý to robí s pomocou data setu obsahujúceho interakcie za jeden mesiac. Tieto interakcie sú potom chronologicky prehrané a odporúčania poskytnuté systémom sú porovnané s reálnymi odporúčaniami na ktoré užívatelia klikli. Takto sa počíta v podstate *offline click-through rate*.

Architektúre Idomaaru sa takisto venujem podrobnejšie v druhej II časti tejto práce.

Pri offline vyhodnocovaní treba spomenúť ich nevýhodu súvisiacu s konceptom *zaznamenanej histórie*[18]. Ide o to, že môžu existovať také odporúčania, ktoré mohli byť pre užívateľa dobré alebo zaujímavé, ale chýba o tom záznam, keďže mu neboli prezentované. V offline vyhodnocovaní ale budú takéto odporúčania vyhodnotené ako nesprávne.

2.5 Zhrnutie

Táto kapitola sa venovala problematike odporúčania novinových správ.

- Zaoberala sa špecifikami odporúčania novinových správ ako je rozptyl, popularita, dynamika a kontext.
- Popísala algoritmy špecifické pre odporúčanie novinových správ ako je odporúčanie najnovších správ, najpopulárnejších správ alebo náhodných správ.
- Zaoberala sa aj asociačnými pravidlami, keďže na nich je postavené moje riešenie.
- Nakoniec tu boli popísané možnosti online a offline vyhodnocovania a použité metriky.

Kapitola 3

Analýza existujúcich riešení

Táto kapitola sa venuje súťaži CLEF NewsREEL[13], ktorej sa program *rule-recommender* zúčastnil. Jej hlavným cieľom je ale analyzovať niektoré algoritmy, ktoré v nej boli prihlásené a popísať postupy, ktoré v nej boli vyskúšané.

Jednotlivé prístupy sa od seba významne líšia. Niektoré sú postavené na vlastnostiach položky, iné sú postavené na použití technológie, ktorá urýchľuje spracovanie. Všetky majú ako spoločnú metriku click-through rate a u niektorých je uvedená aj miera odozvy.

3.1 CLEF NewsREEL

CLEF NewsREEL je skratka pre *Conference and Labs of the Evaluation Forum - News Recommendation Evaluation Lab*. Úlohou je poskytovať odporúčania novinových článkov pre užívateľov navštevujúcich novinové portály. V poradí ôsma konferencia, na ktorej budú prezentované výsledky vyhodnocovania sa bude konať 11. až 14. septembra 2017.

NewsREEL sa skladá z dvoch častí **NewsREEL Live** a **NewsREEL Replay**. Tieto časti umožňujú online a offline vyhodnocovanie v tom zmysle ako to bolo uvedené v predošlej kapitole 2.4. Technické aspekty obidvoch úloh sú popísané v praktickej časti tejto práce.

3.1.1 NewsREEL Live

Z hľadiska softvérového inžinierstva a obmedzení z reálneho sveta je ale zaujímavejšie online vyhodnocovanie NewsREEL Live.

Architektúra, na ktorej je založené online vyhodnocovanie, je podrobne popísaná v kapitole Open Recommendation Platform 4. Celý proces online testovania prihláseného algoritmu sa delí na tri časti

[13] `http://www.clef-newsreel.org/`

- *Testovacie obdobie 1*, ktoré sa uskutočnilo v roku 2017 medzi 13. a 19. marcom.
- *Testovacie obdobie 2*, ktoré sa uskutočnilo 27. marca až 3. apríla 2017.
- *Vyhodnocovacie obdobie*, ktoré sa konalo 24. apríla až 7. mája 2017.

Úlohou testovacích období je otestovať algoritmy online pred ostrým vyhodnocovaním. NewsREEL navyše z týchto dvoch období poskytuje užitočné metriky, ktoré môžu vývojárom pomôcť. Výsledkami testovacích období a ich interpretáciou sa zaoberám v poslednej časti tejto práce.

3.2 Existujúce riešenia

3.2.1 Odporúčanie založené na použití Apache Spark

3.2.1.1 Popis

Apache Spark[14] je open-source framework umožňujúci spracovanie veľkého množstva dát. Z tohto dôvodu bol použitý v [21] na implementáciu algoritmu odporúčajúceho najpopulárnejšie správy.

Algoritmus odporúča najpopulárnejšie správy na základe domén alebo kategórií a s použitím *MapReduce* mechanizmu, ktorý Apache Spark poskytuje. Prichádzajúce interakcie, ktoré prichádzajú na server sa delegujú na ďalšiu komponentu. Tá ich ukladí do kruhového bufferu pevnej veľkosti. Následne sa púšťa MapReduce algoritmus, ktorý spočíta a aktualizuje počet obrdržaných klikov pre každý článok. Na základe takto spočítaných štatistík sa počítajú odporúčania. Online testovaniu boli vystavené tri verzie tohto prístupu, ktoré sa líšia veľkosťou použitej dátovej štruktury.

3.2.1.2 Vyhodnotenie

Prvou relevantnou metrikou algoritmu je click-through rate. CTR algoritmu pracujúcom na dátovej štruktúre o veľkosti 300 položiek bolo 2.93 percenta. Algoritmus berúci do úvahy kategórie pracujúci maximálne so 100 položkami pre jednu kategóriu dosiahol CTR 2.84 precenta. V porovnaní s tým uvádzajú autori CTR baseline algoritmu, ktoré je 0.59 percenta[21].

Miera odozvy (response rate) sa v prípade NewsREEL meria ako percento odpovedí, ktoré neprekročia limit 100ms. Tento prístup dosiahol po 30 dňoch testovania minimálnu response rate 99.43 percent a priemernú 99.76 percent. Jedná sa teda o prístup s vysokou dostupnosťou.

[14] `https://spark.apache.org/`

3.2.2 Odporúčanie založené na použití Apache Flink

3.2.2.1 Popis

Podobným frameworkom umožňujúcim spracovanie dát v reálnom čase ako Apache Storm, je **Apache Flink**[15]. Tento prístup bol použitý v [22]. Aj tento prístup vychádza z odporúčania najpopulárnejších správ.

Systém sa skladá zo štyroch komponent:

- **HTTP koncový bod** prijímajúci správy. Na základe toho, o aký typ správy sa jedná, sa rozhoduje ďalej. Ak sa jedná o klik, správa sa posunie Apache Flink. Ak sa jedná o žiadosť o odporúčanie, tá sa posunie komponente, ktorá má odporúčanie na starosť. Táto komponenta má navyše na starosť konverziu odpovede do správneho formátu.
- **Apache Flink komponenta**, ktorá na základe údajov o klikoch počíta štatistiky. Pracuje iba s aktuálnymi položkami. Robí to za pomoci metódy časového okna, kde všetky staré položky sa neberú do úvahy.
- **Modely** vytvorené zo štatistík, ktoré sú uložené v databáze. Ukladajú sa portály, kategórie položiek, položky a počty ich zhliadnutí. Ako databáza sa používa relačná MySQL[16].
- **Komponenta**, ktorá počíta odporúčanie pre príchadzajúce žiadosti. Odporúčania sa počítajú na základe komunikácie s databázou, kde sa hľadajú najpopulárnejšie články v danej kategórii alebo od daného vydávateľa.

3.2.2.2 Vyhodnotenie

Takýto odporúčací systém sa zúčastnil offline vyhodonocovania. V ňom dosiahol offline click-through rate pohybujúci sa medzi 1 až 1.6 percentami[22]. Hodnota offline CTR sa menila vzhľadom ku veľkosti časového okna, ktoré bolo použité.

Autori neuvádzajú aká veľká bola miera odozvy. Určitým indikátorom by mohla byť skutočnosť, že systém zvládal až tisíc žiadostí súčasne, čo je omnoho viac ako nastáva pri online vyhodnocovaní[22].

3.2.3 Odporúčanie na základe obrázkov

3.2.3.1 Popis

Tento prístup[23] odporúča články na základe toho, či obsahujú alebo neobsahujú zaujímavé obrázky. Autori výskumom zistili, že obrázok je zaujimavý, ak obsahuje v strede jednu osobu alebo jeden objekt[23].

[15] `https://flink.apache.org/`
[16] `https://www.mysql.com/`

Ak systém obdrží notifikáciu o novej správe, pomocou atribútu `url_img` sa získa z odkazu príslušný obrázok. Ten sa potom na základe algoritmu Viola-Jones označí ako zaujímavý alebo nezaujímavý a pridá do zoznamu možných odporúčaní. Ak dorazí žiadosť o odporúčanie a neexistuje dostatok odporúčaní založených na obrázkoch, doplnia sa tieto odporúčania v jednej verzii najpopulárnejšími, v druhej náhodnými článkami.

3.2.3.2 Vyhodnotenie

Výsledky tohto algoritmu sú z časového obdobia 24 dní. V porovnaní s baseline algoritmom bol o 28 percent horší. Autori sa ale domnievajú, že takýto výsledok je hlavne spôsobený technickými problémami. Púšťanie algoritmu na spracovanie obrázkov bolo výpočetne náročné vzhľadom na príchod správ. Toto zrejme aj zapríčinilo nižšiu mieru odozvy.

3.2.4 Odporúčanie založené na popularite položky

3.2.4.1 Popis

Internetový portál **reddit**[17] umožňuje užívateľom po tom ako sa registrujú hodnotiť položky pozitívne alebo negatívne. Na základe týchto hodnotení a aktuálnosti položky sa zostavuje poradie podľa ktorého sú položky na hlavnej stránke zobrazené.

Z tejto myšlienky vychádzali autori prístupu[24]. Popularitu položky počítajú ako počet návštev položky, ktorá nie je nasledovaná kliknutím na inú položku. Užívateľ teda musí ostať čítať článok po dobu aspoň 30 sekúnd, aby sa považoval za populárny. Do úvahy sa berie podobne ako pri reddite čas, a aby bol článok považovaný za hodný odporúčania, musí ho navštíviť určitý počet užívateľov.

3.2.4.2 Vyhodnotenie

V offline testovaní mal tento algoritmus lepší click-through rate ako baseline algoritmus. Takisto zvládol aj väčší počet maximálnych simultánnych požiadavok a zvládol lepšie prácu s pamäťou[24].

3.2.5 Odporúčanie založené na kolaboratívnom filtrovaní

3.2.5.1 Popis

Spoločnosť **Amazon**[18] odporúča položky na základe toho, aké druhy položiek majú užívatelia vo svojom nákupnom košíku. Jedná sa o kolaboratívne filtrovanie založené na položkách.

[17] `https://www.reddit.com/`
[18] `https://www.amazon.com/`

Prístup[24] to preniesol do domény článkov. Ak viacero užívateľov prečítalo dva články, tieto články majú niečo spoločné. Ak teda dorazí požiadavok na odporúčanie, užívateľovi sa odporučí článok podobný jednému z tých, ktoré už v minulosti čítal. Takýto prístup sa autori rozhodli zvoliť napriek tomu, že sledovať články, ktoré užívateľ navštívil je možné len po veľmi krátku dobu.

3.2.5.2 Vyhodnotenie

Tento prístup dosiahol horší click-through rate ako baseline algoritmus. Jeho maximálny počet simultánnych žiadostí bol takisto nižší ako u baseline algoritmu.

3.2.6 Odporúčanie využívajúce množinu algoritmov

3.2.6.1 Popis

Možným riešením ako využiť odlišné výhody jednotlivých algoritmov je navrhnúť systém, ktorý bude na odporúčanie využívať viacero algoritmov.

Podobný systém navrhli autori v [25]. Každému vydávateľovi v ňom prislúcha skupina systémov. Hlavný agent, ktorý prijíma žiadosti o odporúčanie vyberá z tejto skupiny najlepší algoritmus ma základe jeho doterajších výsledkov a ďalších kritérií. Týmito kritériami sú hodina, deň v týždni, počet položiek, ktoré majú byť odporúčané a frekvencia s akou užívateľ navštevuje portál.

3.2.6.2 Vyhodnotenie

Online testovania sa zúčastnili tri modifikácie tohto algoritmu, líšiace sa v kritériach na základe ktorých bola správa delegovaná. Prvá modifikácia brala do úvahy deň v týždni a počet položiek, ktoré majú byť odporučené. Druhá modifikácia vybrala algoritmus na základe toho, ako si viedol za posledných 25 správ. Posledná modifikácia delegovala žiadosti na základe popularity článkov. Popularita sa počítala za poslednú hodinu.

Všetky modifikácie sa ukázali ako úspešné. Ich click-through rate sa pohyboval počas celého vyhodnocovacieho obdobia medzi 1.18 až 1.44 percenta. Priermerný click-through rate prvej bol 1.33, druhej 1.32 a tretej 1.26 percenta. Takýto výsledok prekonal baseline a bol prekonaný len jedným algoritmom.

3.2.7 Odporúčanie na základe pravidiel

3.2.7.1 Popis

Odporúčanie na základe pravidiel je hlavnou témou tejto diplomovej práce. Jedná sa o prístup, ktorý už bol odskúšaný z dobrými výsledkami[26].

Tento algoritmus používal kontextuálne informácie o položke a užívateľovi získané z interakcie ako ľavú stranu pravidla. Pravú stranu pravidla tvoril identifikátor položky. Na získanie pravidiel sa použil algoritmus apriori. Ak sa ľavá strana nejakého pravidla zhodovala z informáciami obsiahnutými v žiadosti o odporúčania, identifikátor z pravej strany bol zaradený medzi odporuччené položky.

V prípade, že neexistovalo vhodné pravidlo boli použité dve ďalšie algoritmy. Prvý vracal najčítanejšie správy v rámci jedného dňa a druhý najaktuálnejšie správy z rovnakej domény.

3.2.7.2 Vyhodnotenie

Tento prístup sa osvedčil a v roku 2014 sa umiestnil na treťom mieste v celkovom počte úspešných odporúčaní. Dosiahnutý click-through rate bol 1.28 percenta a click-through rate odporúčania len na základe asociačných pravidiel bol 1.5 percenta.

3.3 Zhrnutie

V tejto kapitole bola priblížená súťaž CLEF NewsREEL a boli v nej aj analyzované algoritmy a prístupy, ktoré sa tejto súťaže zúčastnili.

Jednotlivé prístupy a algoritmy sa od seba výrazne líšia. Niektoré z nich sú postavené na konkrétnej technológii, ktorá má spracovanie správ urýchliť. Iné vychádzali z klasických odporúčacích algoritmov. Ďalšie vychádzali z poznatkov o správaní užívateľa.

Celkovo sa jednalo o tieto algoritmy:

- Odporúčanie založené na použití Apache Spark[21]
- Odporúčanie založené na použití Apache Flink[22]
- Odporúčanie na základe obrázkov[23]
- Odporúčanie založené na popularite položky[24]
- Odporúčanie založené na kolaboratívnom filtrovaní[24]
- Odporúčanie využívajúce množinu algoritmov[25]
- Odporúčanie na základe pravidiel[26]

Ich výsledky su zhrnuté v tabuľke 3.1 a tabuľke 3.2.

Algoritmus	**CTR**
Apache Spark	Medzi 2.84 a 2.93 percentami (online)
Apache Flink	Medzi 1 a 1.6 percentami (offline)
Obrázky	Nižší ako u baseline algoritmu (online)
Popularita	Vyšší ako u baseline algoritmu (online)
Kolaboratívne filtrovanie	Nižší ako u baseline algoritmu (offline)
Množina algoritmov	Medzi 1.26 a 1.33 percentami (online)
Pravidlá	1.28 percenta (online)

Tabuľka 3.1: Click-through rate jednotlivých algoritmov

Algoritmus	**Response rate**
Apache Spark	99.76 percent
Apache Flink	Nie je uvedená
Obrázky	Nižšia ako u baseline algoritmu
Popularita	Vyššia ako u baseline algoritmu
Kolaboratívne filtrovanie	Nižšia ako u baseline algoritmu
Množina algoritmov	Nie je uvedená
Pravidlá	Nie je uvedená

Tabuľka 3.2: Miera odozvy jednotlivých algoritmov

Časť II

Praktická časť

Kapitola 4

Open Recommendation Platform

Táto kapitola sa venuje metódam offline a online vyhodnocovania použitým vo výzve CLEF NewsREEL.

Začína popisom frameworku Idomaar, ktorý je použitý na offline vyhodnocovanie. Následne je tu popísaná Open Recommendation Platform (ORP), ktorá bola v súťaži použitá na online vyhodnocovanie. Je tu popísaná architektúra ORP a sú tu popísané správy, ktoré posiela spolu s ich formátom.

4.1 Idomaar

Na offline vyhodnocovanie sa používa open-source vyhodnocovací framework Idomaar[19], spomenutý v sekcii 2.4.1.1.

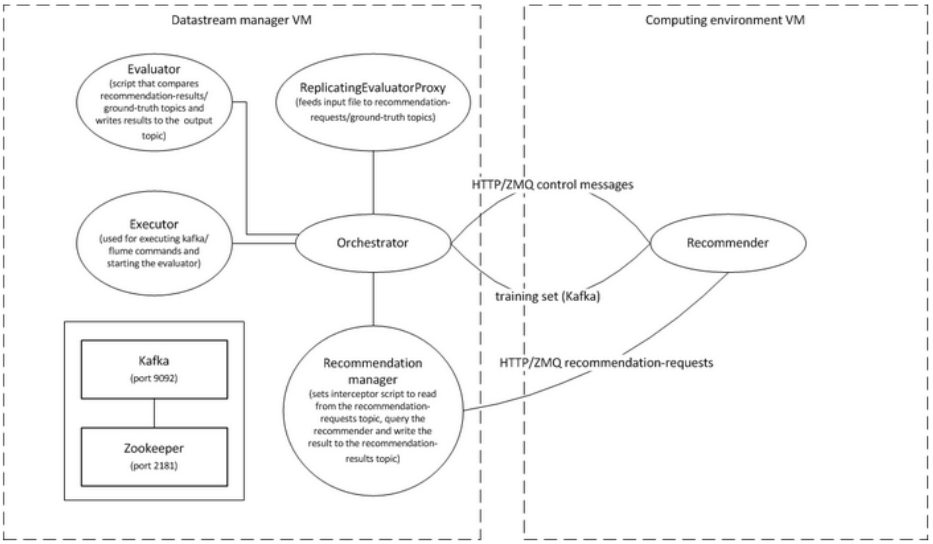
Tento framework sa skladá z niekoľkých častí. Tvorí ho výpočetné prostredie *computing environment*, obsahujúce algoritmus, ktorý chceme otestovať. Ďalej ho tvorí ho *vyhodnocovacia logika*, ktorá algoritmy vyhodnocuje a počíta ich metriky. Tvorí ho aj *orchestrator*, ktorý má na starosť súhru jednotlivých častí. Nakoniec ho tvoria samotné *dáta*, ktoré posielame computing environmentu.

4.1.1 Výpočetná časť

Ako už bolo uvedené, výpočetné prostredie by malo obsluhovať jednotlivé žiadosti. Na tomto prostredí by mal byť nasadený odporúčací algoritmus.

Výpočetné prostredie komunikuje z orchestrátorom pomocou dvoch kanálov. Prvý kanál sa používa na to, aby sa ku prostrediu dostali dáta. Tento kanál je jednosmerný a je realizovaný ako *Apache Kafka*[20] konektor.

[19] `https://github.com/crowdrec/idomaar`
[20] `https://kafka.apache.org/`



Obr. 4.1: Architektúra Idomaaru

Druhý kanál je obojsmerný a používa sa na výmenu riadiacich správ. Pomocou nich posiela orchestrátor žiadosť o odporúčanie a výpočetné prostredie odporúčané položky. Toto je realizované pomocou protokolu HTTP alebo *ZeroMQ*[21].

4.1.2 Vyhodnocovacia časť

Vyhodnocovacia časť má dve úlohy. Prvou je rozdeliť dáta na časti na základe vyhodnocovacej stratégie. Druhou je počítať metriky ako RMSE alebo čas odozvy.

Jednotlivé komponenty a ich interakcia na obrázku 4.1.2 [22].

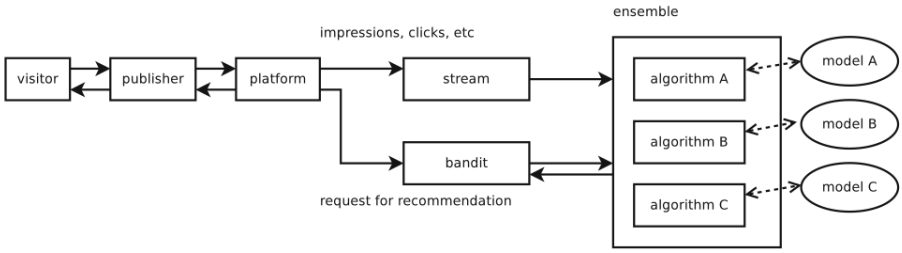
4.2 Open Recommendation Platform

Ako už bolo spomenuté v sekcii 2.4.1.1, na online testovanie môjho algoritmu som použil Open Recommendation Platform. Odporúčací algoritmus, ktorý chce z ORP komunikovať, by mal byť postavený na Software Development Kit (SDK), ktorý plista poskytuje.

Toto SDK tvorí kostru, obsahujúcu controller triedu umožňujúcu spracovanie prichádzajúcich a odchádzajúcich správ. SDK je zatiaľ dostupné v jazykoch PHP, Java a Node.js. V mojej aplikácii som použil SDK pre Javu.

[21] `http://zeromq.org/`

[22] Zdroj: `https://github.com/crowdrec/idomaar/wiki/Idomaar-architecture`



Obr. 4.2: Architektúra ORP. Zdroj[12]

4.2.1 Architektúra

Ak algoritmus teda implementuje rozhranie poskytnuté plistou, môže za pomoci POST príkazov požiadať ORP o spustenie alebo zastavenie premávky. Toto dáva možnosť opraviť si chybu v algoritme.

Celú ORP architektúru možno vidieť na obrázku 4.2. Návštevník spustí svojou návštevou článku udalosť, ktorú obdrží vydavateľ. Ten túto správu poskytne platforme. Ak sa jedná o obyčajnú návštevu článku alebo o kliknutie na odporúčaný článok, túto správu obdržia všetky aktuálne prihlásené algoritmy. Ak sa jedná o žiadosť o odporúčanie ORP vyberie metódou *multi-armed bandit* algoritmus, ktorý prevedie výpočet a poskytne odpoveď.

Odporučené položky alebo články sú reprezentované ich číselnými identifikátormi a sú uložené v jednorozmernom poli. Poskytnuté odporúčanie alebo odporúčania dostane vydavateľ pomocou statického javascriptu, ktorý ich získa z ORP. Toto odporúčanie sa následne zobrazí vo widgete s popiskom "You might also be interested in..."[12].

4.2.2 ORP protokol

ORP pracuje na princípe programovania založeného na udalostiach, ktoré určujú chod programu.

Od ORP môžme obdržať správy patriace do štyroch skupín udalostí: *recommendation_request*, *item_update*, *event_notification* a *error_notification*. Komunikácia prebieha pomocou protokolu HTTP a typy udalostí sú zakódované ako parameter POST správy. Frekvencia príchodu týchto udalostí sa pohybuje v tisícoch za sekundu. Limit na odpoveď je 100 ms, a je kľúčové ho dodržať. Samotné udalosti sú uložené vo formáte JSON.

4.2.3 Typy udalostí

`Recommendation_request` je žiadosť o odporúčanie. Po obdržaní tohto typu správy by mal algoritmus previesť výpočet, získať odporúčania a vrátiť ich.

```
{
  "item1": "cheese",
  "item2": "apple",
  "item3": "milk"
}
```

Obr. 4.3: Príklad JSON objektu

`Item_update` je typ správy, ktorí upozorňuje na to, že buď pribudla nová položka alebo existujúca položka bola zmenená.

`Event_notification` upozorňuje na rôzne udalosti, ktoré nastali potom ako a užívateľovi zobrazil článok. Delí sa ďalej na `impression`, `impression_empty` a `click`. Impression reprezentuje situáciu keď užívateľ navštívil článok a zobrazili sa mu odporúčania. Ak navštívil článok a neobdržal žiadne odporúčania dostaneme správu impression_empty. Click reprezentuje kliknutie užívateľa na odporúčanie.

Identifikátory správ, ktoré dostal užívateľ odporučené a identifikátor položky na ktorú klikol, sú uložené v jednotlivých JSON objektoch.

`Error_notification` je typ správy oznamujúci chybu, ktorá mohla nastať pri spracovaní.

4.2.3.1 JSON

JSON je spôsob zápisu dát nezávislý na počítačovej platforme a nezávislý na jazyku. Jedná sa o najčastejšie používaný formát pre asynchrónnu komunikáciu medzi prehliadačom a serverom.

Dáta sú prenášané pomocou dvojíc kľúč-hodnota. Jeho základné dátove typy sú `Number`, `String`, `Boolean`, `Array`, `Object` a `null`.

Pre úplnosť ešte treba uviesť, že JSON povoľuje biele znaky a nepodporuje syntax umožňujúcu komentáre. Jednoduchý JSON objekt môžme vidieť na obrázku 4.3.

V programovacom jazyku Java uľahčujú prácu z JSON formátom triedy `javax.json.JSONObject` a `javax.json.JSONArray`, ktoré som aj použil.

4.2.4 ORP datové typy

Dáta sú reprezentované ako štruktúry nazývané vektory[27]. Tieto vektory sú identifikované numerickými identifikátormi a asociované z prvkami rozličných datových typov. Vektory sa ďalej delia na vstupné, ktoré obsahujú informácie o správach a kontext správ, a výstupné, ktoré obsahujú informácie o výsledkoch výpočtu, medzi inými aj identifikátory odporúčaných položiek.

Existujú tri typy vektorov: `simple`, `list` a `cluster`. Prvý uvedený obsahuje jednu hodnotu (väčšinou numerickú), druhý obsahuje zoznam hodnôt a

```
{
  "simple": {
    "4": 415,
    "5": 1364,
  },
  "lists": {
    "11": [1478, 1479, 1480],
  },
  "clusters": {
    "1": {
      "147": 11,
      "148": 12,
      "1": 19,
    }
  }
}
```

Obr. 4.4: Príklad ORP správy

tretí obsahuje mapu hodnôt. V samotnej správe sú jednotlivé hodnoty zoskupené podľa vektorov.

Príklad takejto skrátenej správy je možné vidieť na obrázku 4.4. V tomto prípade tvoria 4, 5, 11 a 1 vstupné vektory a podľa ORP dokumentácie[23] zodpovedajú internetovému prehliadaču, poskytovateľovi internetového pripojenia, zozname kategórii a pohlaviu.

ORP obsahuje až 59 takýchto vstupných vektorov obsahujúcich informácie od odhadovaného veku a odhadovaného príjmu až po po pozíciu článku v rámci stránky. Treba ale aj dodať, že plista u väčšiny týchto vektorov nezverejňuje čo daný vektor reprezentuje, a tak je ťažko zistiť aký je rozdiel medzi `CATEGORY_SEM` a `CATEGORY`, alebo `GEO_TYPE` a `GEO_USER`.

Najdôležitejšími vektormi sú `PUBLISHER`, `ITEM_SOURCE`, `KEYWORD` a `USER_COOKIE`. Vektor `PUBLISHER` reprezentuje číselným identifikátorom vydavateľa, zvykne sa používať aj výraz *domainId*. Všetky algoritmy odoberajúce správy z ORP, by mal vraciať odporúčania na položky z rovnakej domény. `ITEM_SOURCE` je vektor, ktorý jednoznačne identifikuje položku. `KEYWORD` je cluster, ktorý obsahuje kľúčové slová súvisiace s článkom. Tento cluster obsahuje kľúčove slová ako kľúče clusteru, a počet ich výskytov ako hodnotu spárovanú s príslušným kľúčom. Posledný dôležitý vektor je `USER_COOKIE`, ktorý reprezentuje užívateľa celým číslom. Neznámy užívateľ je reprezentovaný 0.

[23] `https://orp.plista.com/documentation/download`

4.3 Zhrnutie

Táto kapitola sa venovala metódam offline a online vyhodnocovania použitým vo výzve CLEF NewsREEL.

- Zaoberala sa frameworkom Idomaar umožňujúcim offline vyhodnocovanie.
- Predstavila platformou Open Recommendation Platform umožňujúcou online vyhodnocovanie
- Popísala architektúru ORP.
- Zaoberala sa ORP protokolom, typmi správ a dátovymi typmi použitými v ORP.

Kapitola 5

Implementácia

Táto kapitola sa venuje Java programu *rule-recommender*, ktorý bol mojím príspevkom do súťaže CLEF NewsREEL. Nachádza sa tu aj popis základného *baseline* algoritmu, ktorý používa ORP.

Následne sa venuje popisu samotného programu a stručne aj použitým technológiám. Popisuje zmeny oproti základnej verzii, ale aj realizáciu odporúčania na základe asociačných pravidiel.

5.1 Použité technológie

Príspevky do výzvy CLEF NewsREEL by mali implementovať rozhranie poskytnuté organizátormi. Toto rozhranie (ORP SDK) je zatiaľ dostupné v troch jazykoch: PHP, Java a Node.js. Na základe analýzy existujúcich riešení som sa rozhodol v mojom riešení využiť odporúčanie založené na asociačných pravidlách. Z toho vyplynulo, že v zvolenom jazyku by malo byť možné buď pravidlá priamo ťažiť alebo by mal tento jazyk byť schopný komunikovať s prostredím, ktoré tieto pravidlá získa. Z týchto dôvodov som sa rozhodol pre programovací jazyk Java.

Existuje mnoho prostredí, ktoré umožňujú asociačné pravidlá ťažiť. Jedinou požiadavkou bolo aby bola možná integrácia tohto prostredia a programu v Jave. Zvolil som programovací jazyk R[24] a konkrétne prostredie RServe[25]. Prostredie RServe poskytuje jednoduché rozhranie umožňujúce vykonávanie príkazov v R z Javy.

[24] `https://cran.r-project.org/`
[25] `https://rforge.net/Rserve/`

5.2 Baseline

5.2.1 Úvod

V SDK pre Open Recommendation Platform, ktorú by mal každý účastník implementovať sa nachádza jednoduchá implementácia odporúčacieho algoritmu. Tento algoritmus by mal slúžiť ako *baseline*, mal by umožniť porovnať si oproti nemu vlastnú implementáciu a vlastná implementácia by mala mať lepšie výsledky.

V prípade môjho algoritmu navyše platí, že ak sa nenájde žiadne pasujúce pravidlo, algoritmus "spadne" na baseline a vráti odporúčanie podľa toho. Okrem toho, moja implementácia používa triedy používané baseline algoritmom a celková architektúra programu je podobná baseline algoritmu. Z týchto dôvodov považujem za užitočné popísať ho tu.

5.2.2 Model

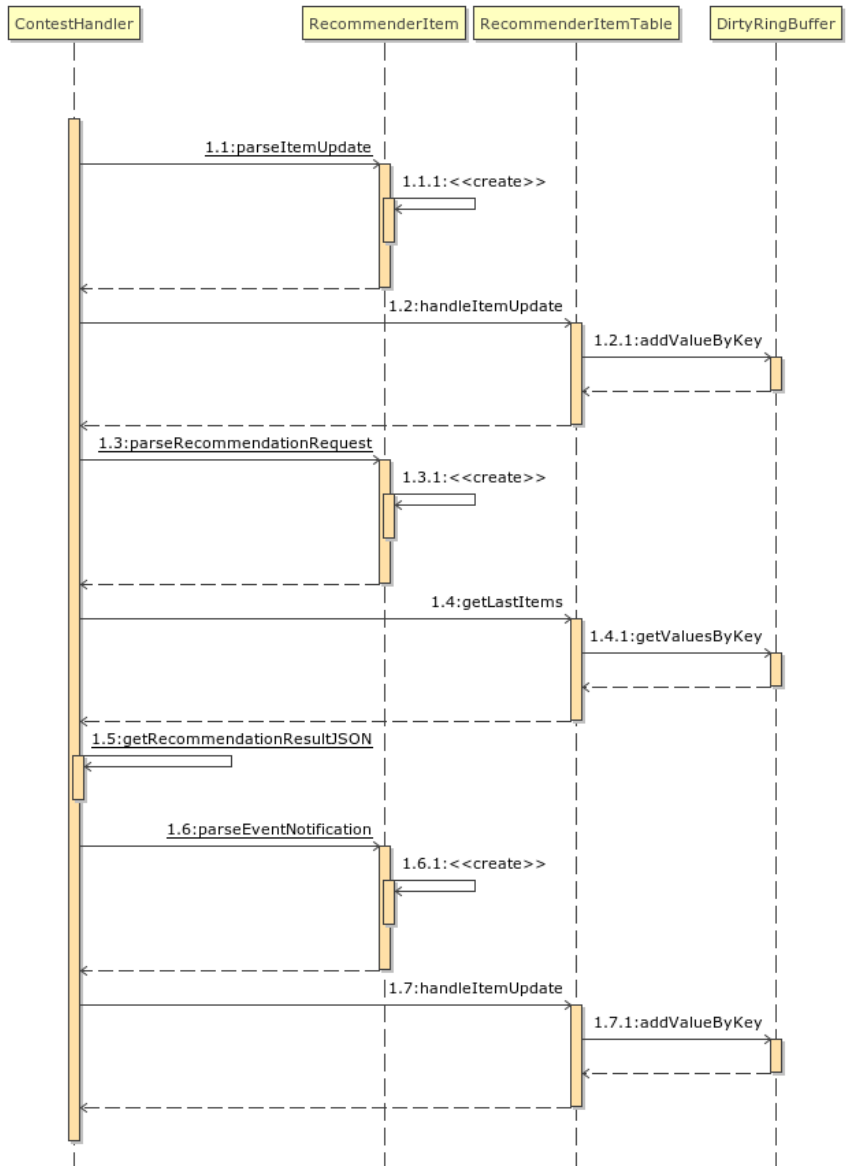
5.2.2.1 Položka

Položka je reprezentovaná triedou `RecommenderItem`. Jednotlivé atribúty veci, ktoré získame rozparsovaním JSON správy (to sa deje v metódach RecommenderItem *parseEventNotification*, *parseRecommendationRequest* a *parseItemUpdate*) sú uložené ako hodnoty v objekte triedy `java.util.HashMap` s názvom `valuesById`.

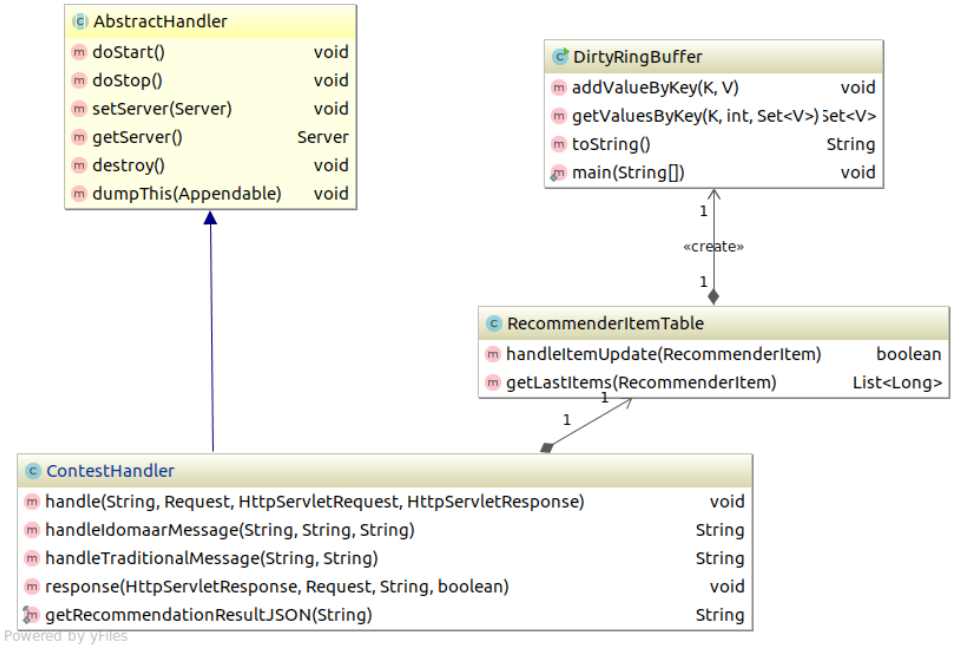
Každá položka, ktorá vznikne parsovaním JSON správy, je následne uložené do objektu `RecommenderItemTable`. Tento objekt zapuzdruje tabuľku `DirtyRingBuffer`. DirtyRingBuffer tvoria páry kľúč-hodnota, kde kľúčom je doména položky (vydavateľ) a hodnotou zoznam položiek z tej domény. Ak dĺžka tohto zoznamu prekročí istú hranicu (to vstupuje do konštruktoru ako argument), najstaršie položky sú vyhodené. Vzťahy medzi triedami zobrazuje diagram 5.2.

5.2.3 Proces spracovania správy

Proces spracovania správy je vidieť na diagrame 5.1. Trieda `ContestHandler` spracováva správy. Na základe toho o aký typ správy sa jedná zavolá buď metódu parseItemUpdate, parseEventNotification alebo parseRecommendationRequest. Tieto metódy vytvoria objekt triedy RecommenderItem. Ak sa jedná o pridanie novej položky alebo úpravu existujúcej položky, upraví sa DirtyRingBuffer. Ak sa jedná o žiadosť o odporúčanie volaním *getValuesByKey* sa získaju identifikátory položiek z DirtyRingBuffer.



Obr. 5.1: Spracovanie správy - sekvenčný diagram.



Obr. 5.2: ORP SDK - diagram tried.

5.3 Rule recommender

Ako bolo spomenuté v predošlej sekcii, pri implementácii finálnej verzii algoritmu som vychádal z tried a datových štruktúr použitých v baseline verzii.

Prvou zmenou by sa dalo nazvať použitie nástroja `Apache Maven`[26], ktorý umožňuje spravovať a automatizovať zostavenia. Tento nástroj jednak umožnil jednoduchšiu prácu s novými závislosťami a umožnil aj pohodlnejšie vytvárať JAR súbory, ktoré boli nasadzované na server a na ktoré smerovali žiadosti z ORP.

5.3.1 Model

5.3.1.1 Položka

Doplnením oproti baseline verzii je trieda `RecommenderItemBuilder`, ktorá má na starosť vytvorenie položky. Súvisí s pridávaním veľkého počtu nových atribútov ku položke ako počasie, vek, pohlavie, príjem alebo poloha, ktorých hodnota môže ale nemusí byť null. Riešením je implementácia návrhoveho vzoru *Builder*.

[26] `https://maven.apache.org/`

```
"itemId","category","gender","income","age","geo_user","keyword"
196213792;81565;0;0;1;2878841;254879229
254879229;0,0;1;2878841;222716;254297974
254297974;NA;0;NA;2878841;38581;255886863
```

Obr. 5.3: Ukážka CSV súboru s klikmi

```
"","rules","support","confidence","lift"
"1","{} => {itemId=257046085}",0.36,0.36,1
"20","{category=2875514} => {itemId=254370758}",0.06,0.33,5.55
"53","{geo_user=18850} => {itemId=257046085}",0.06,0.5,1.38
```

Obr. 5.4: Ukážka CSV súboru s pravidlami

Samotné parsovanie JSON správ som presunul do osobitnej triedy `Parser`. Tým sa stala RecommenderItem priehľadnejšia. Zrozumiteľnejším sa stal aj výstup z chýb, ktoré nastali pri parsovaní.

5.3.1.2 Asociačné pravidlo

Asociačné pravidlo je reprezentované triedou `Rule`. Pravidlá z jednej domény sú zoskupené v objekte triedy `RuleCollection`. RuleCollection má aj metódy umožňujúce pravidlá zoradiť, odstrániť duplicititné pravidlá a nakoniec aj pokúsiť sa nájsť ku danej položke pravidlo so zhodujúcou sa ľavou stranou.

Položky a asociačné pravidlá sa ukladajú dvojako. Ukladajú sa do kolekcií v Jave a ukladajú sa do súborov. Ukladanie do kolekcií je výhodné kvôli pohodlnejšiemu porovnávaniu s ľavými stranami pravidiel.

Zmeny vo vzťahoch medzi triedami sú zobrazené na diagrame 5.8.

5.3.1.3 CSV súbory

Ukladanie do súborov umožňuje jednoduchšiu prácu s nimi v prostredí Rserve. Tieto súbory sa potom nachádzajú v zložkách `clicks` a `rules`, ktoré si aplikácia vytvorí. Konverziu položky na riadok v CSV súbore má na starosti trieda `CsvConverter`. Neznáme hodnoty sú reprezentované hodnotou NA. Ukážka takéhoto CSV súboru sa nachádza na obrázku 5.3 a obrázku 5.4.

5.3.2 Synchronizácia

Ako už bolo spomenuté, na rozdiel od baseline verzie sa používa viacero dátových štruktúr na čítanie a zápis jednotlivých klikov a pravidiel. V situácii ak dve alebo viacero vlákien sa pokúsi pristúpiť ku kolekcii jednej domény alebo k jednému súboru, nastávajú synchronizačné problémy. V prípade Javy to

je väčšinou výnimka `ConcurrentModificationException`. Jedným z riešení tohto problému je použiť na to špeciálne dátové štruktúry.

5.3.2.1 Tabuľky

Ako objekt mapujúci číselný identifikátor domény na kliky prichádzajúce z tejto domény som použil triedu `java.util.concurrent.ConcurrentHashMap`. Táto trieda implementuje rozhranie známej `java.util.Map`, takže práca s ňou je veľmi jednoduchá.

5.3.2.2 Zoznamy

Vhodnou náhradou klasického `java.util.ArrayList` sa ukázala byť `ArrayBlockingQueue` z balíčku `java.util.concurrent`. Nejedná o klasický zoznam, ale o dátovú štruktúru FIFO, takže na pridávanie a odoberanie sa používajú operácie ako *add()* a *poll()*. Jej kapacita je pevná a nastavuje sa v konštruktore. Táto hodnota bola takisto získavaná ako parameter zo súboru.

5.3.2.3 Súbory

Ďalšou problémovou situáciou by mohla byť úprava CSV súborov viacerými vláknami. Tu bolo riešením použitie `synchronized` blokov, kde samotný súbor tvoril zámok.

5.3.2.4 R

Posledným problémom bola komunikácia so serverom Rserve, ktorý je podrobnejšie popísaný v ďalšej sekcii, a samotné ťaženie pravidiel. Tento proces má preto na starosť osobitné vlákno.

5.3.3 RServe

Server **Rserve**[27] som sa rozhodol použiť preto, lebo je to jedna z mála možností ako v Jave využívať prednosti jazyka R. Rserve podporuje okrem Javy aj C/C++ a PHP. Podporuje vzdialené pripojenie, autentikáciu a prenos súborov.

Rserve sa používa veľmi jednoducho. Zrejme najdôležitejšia trieda je `RConnection`. Tú inicializujeme zavolaním bezparametrického konštruktoru. RConnection obsahuje metódy ako `voidEval` a `eval`, ktorých argumentami sú kusy kódu v R, ktoré sa majú vykonať a ktoré nám umožňujú previesť kód v R a prípadne získať jeho výsledok. Ukážku použitia Rserve môžme vidieť na obrázku 5.5.

[27] `https://rforge.net/Rserve/`

```
rConnection.voidEval("clicks <- read.csv(\"clicks1234.csv\")")
```

Obr. 5.5: Ukážka volania Rserve

Pri ladení môže pomôcť fakt, že nesprávny alebo problémový príkaz hodí výnimku triedy `RserveException` alebo `REXPMismatchException`. RConnection ešte obsahuje metódu `getLastError`, ktorá vráti poslednú chybu, ktorá nastala. Táto chyba ale nie je až taka veľavravná ako samotný chybový výpis v jazyku R.

5.3.4 Spracovanie správy

Spracovanie správy prebieha podobne ako v baseline verzii, trieda `RController` sa stará o komunikáciu s Rserve, zapisovanie nových položiek, aktualizovanie starých položiek, zapisovanie nových pravidiel a aktualizáciu pravidiel. Tieto skutočnosti sú vyznačené na sekvenčnom diagrame 5.10 a diagrame 5.9.

5.3.5 Pravidlá

5.3.5.1 arules

Z jednotlivých "klikov" ktoré sa nachádzajú v príslušnom CSV súbore, sa vytvárajú asociačné pravidlá. Toto umožňuje v jazyku R knižnica **arules**[28].

Ťaženie pravidiel má na starosť osobitné vlákno, objekt triedy `RuleMiningThread`. Proces aktualizovania pravidiel sa spúšťa v určitom časovom intervale.

5.3.5.2 Proces

Po uplynutí určitého časového intervalu zavolá `RuleMiningThread` pre každú doménu metódu `updateRules`. V tejto metóde sa udeje väčšna práce súvisiaca so získavaním asociačných pravidiel.

V prvom rade sa kliky pre danú doménu zapíšu do CSV súboru. Z toho súboru sa pomocou Rserve sa načítajú ako objekt `data.frame`. Tento objekt poskytuje možnosť vytvoriť z klikov transakcie. Dataframe zapuzdruje binárnu maticu incidencie. Riadky tejto matice reprezentujú jednotlivé položky a stĺpce hodnoty atribútov týchto položiek.

Aby sme mohli vytvoriť z objektu dataframe transakcie, každý stĺpec musí obsahovať buď logické hodnoty alebo hodnoty z obmedzeného intervalu, v R nazývané *faktory*. Keďže v prípade článku väčšinu informácii tvoria číselné identifikátory reprezentujúce napríklad kategóriu alebo kľúčové slovo, faktorizácia bola vždy nutná pred vytváraním transakcií.

[28] `https://cran.r-project.org/web/packages/arules/index.html`

```
clicks <- read.csv("clicks1234.csv")
clicks$a <- factor(clicks$a)
clicks$b <- factor(clicks$b)
trans <- as(clicks, "transactions")
rules <- apriori(trans,
  parameter= list(confidence= 0.05,
                  support= 0.01,
                  minlen= 1,
                  maxlen= 6))
```

Obr. 5.6: Získavanie asociačných pravidiel v R

Transakcie sa vytvoria po načítaní klikov do objektu dataframe a jeho faktorizácii. Z týchto transakcií sú následne vytvorené asociačné pravidlá.

Na získanie asociačných pravidiel je použitý algoritmus `apriori`, ktorého parametrami sú:

- *transakcie*, reprezentujúce dáta, ktoré chceme ťažiť
- minimálna hodnota *podpory*
- minimálna hodnota *dôvery*
- *minimálna dĺžka* vytvorených asociačných pravidiel
- *maximálna dĺžka* vytvorených asociačných pravidiel

Hodnota a dôvera a ich význam boli definované v sekcii 2.3.4. Hodnoty, ktoré boli nakoniec zvolené, boli predmetom offline experimentovania a experimentov počas testovacieho obdobia CLEF NewsREEL. Získanie pravidiel je teda výsledok sekvencie príkazov v R, veľmi podobnej ako tej na obrázku 5.6.

Pre účely odporúčania sú relevantné pravidlá, ktorých pravú stranu tvorí identifikátor položky. Z tohto dôvodu sú pravidlá, ktoré sú výstupom algoritmu apriori filtrované.

Po prefiltrovaní prebehne ešte pruning pravidiel, popísaný v ďalšej časti. Z výsledných pravidiel je vytvorený objekt dataframe a sú zapísané do CSV súboru pre príslušnú doménu. Tento proces je znázornený na diagrame 5.11.

5.3.5.3 Pruning

Medzi pravidlami získanými algoritmom apriori sa môžu nachádzať aj také, ktoré neobsahujú žiadne nové informácie. Táto informácia je totiž zahrnutá v ostatných pravidlách. Proces odstránenia takýchto zbytočných pravidiel, ktorých metriky sú horšie sa nazýva *pruning*.

```
prunedFrame <- rCBA::pruning(clicks, rulesFrame, method="m2cba")
```

Obr. 5.7: Pruning za pomoci knižnice rCBA

Okrem toho, že medzi pravidlami sa nachádzali aj zbytočné, na hornú hranicu počtu pravidiel získaných algoritmom apriori nebolo kladené žiadne obmedzenie.

Z tohto dôvodu som sa rozhodol pre zmenšenie počtu pravidiel použiť knižnicu jazyka R `rCBA`[28][29]. Pruning pravidiel prebehne takisto pomocou Rserve, ukážka použitia je na obrázku 5.7.

5.3.5.4 Parametry

Aby bolo možné výsledky programu meniť za behu, je nutné za behu meniť parametry ako minimálna dôvera pravidiel, minimálna podpora pravidiel ale aj interval počtu vecí, po ktorom sa pravidlá prepočítajú.

V programe sú tieto a ďalšie parametry spolu s ich počiatočnými hodnotami uložené v triede `RulesConstants`. Na načítanie hodnôt z externého súboru sa používa `org.apache.commons.configuration.PropertiesConfiguration`, ktorá dovoľuje nastaviť stratégiu znovunačítania súboru. Menenie hodnôt za behu umožňuje práve `FileChangedReloadingStrategy` z rovnakého balíčku.

Parametry a ich hodnoty by sa mali nachádzať v zložke s programom, konkrétne v `conf/params.properties`.

5.3.5.5 Logovanie

Logovanie sa stalo veľmi dôležitou časťou implementácie, keďže umožnilo sledovať správanie programu a odhaliť chyby v programe. Na logovanie je použitý framework **log4j** a celkovo sú použité štyri rôzne loggery.

- *clientLogger*, ktorý zachytáva spracovanie správ na najvyššej úrovni - to znamená podarilo/nepodarilo sa. Ďalej zachytáva aký počet položiek bol odporučený baseline algoritmom a aký pomocou pravidiel.
- *itemLogger* zachytáva informácie a chyby, ktoré nastali pri vytváraní objektov RecommenderItem reprezentujúcich položky.
- *rControllerLogger* informuje o komunikácii s prostredím Rserve. Okrem toho informuje aj o počte klikov pre jednotlivé domény a o tom, že boli premazané.

[29] `https://github.com/jaroslav-kuchar/rCBA`

- *ruleLogger* podáva správy o tom kedy prebehlo pre danú doménu ťaženie pravidiel. Na nižších úrovniach je možné aj zistiť, na ktoré pravidlo pasovala položka.

Potrebných informácii je ale priveľa a z toho dôvodu bola v konfiguračnom súbore nastavená maximálna veľkosť jedného log súboru 100 MB a archivuje sa nanajvýš jeden súbor. Logy aplikácie sa nachádzajú v zložke `conf`.

5.3.6 Testovanie

Testovanie programu *rule-recommender* prebiehalo v troch fázach.

- Prvou fázou boli unit testy napísané v Jave s použitím testovacieho frameworku `jUnit`[30]. V nich sa testovali veci ako vytváranie položiek zo správ (trieda `RecommenderItemTest`), konverzia položiek do formátu CSV (trieda `CsvConverterTest`) alebo porovnávanie položiek s pravidlami (trieda `RuleCollectionTest`). Všetky unit testy sú súčasťou programu.

- Druhou fázou bolo testovanie programu na dátach poskytnutých organizátormi CLEF NewsREEL. To umožnilo merať výsledky programu (jednalo sa o metriku offline click-through rate). Na základe týchto meraní boli zvolené parametry použité verziu programu, ktorá bežala v testovacom období. Výsledky týchto experimentov sú podrobnejšie popísané v kapitole 6.

- Treťou úrovňou testovania bola účasť programu v druhom testovacom období. V tomto období boli na program presmerované kliky z reálnych serverov. To umožnilo pozorovať chovanie programu v reálnej prevádzke. Na základe týchto pozorovaní boli opäť zmenené niektoré parametry.

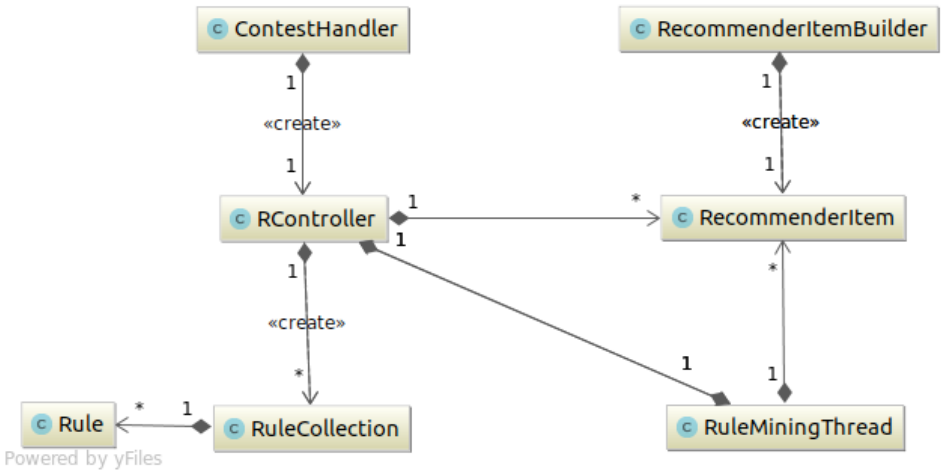
Program, ktorý sa zúčastnil vyhodnocovacieho obdobia výzvy, prešiel postupne všetkými týmito fázami. Vzťahy medzi jednotlivými komponentami sú zachytené na diagrame nasadenia 5.12.

5.4 Zhrnutie

Táto kapitola sa venovala môjmu príspevku do výzvy CLEF NewsREEL, programu *rule-recommender*.

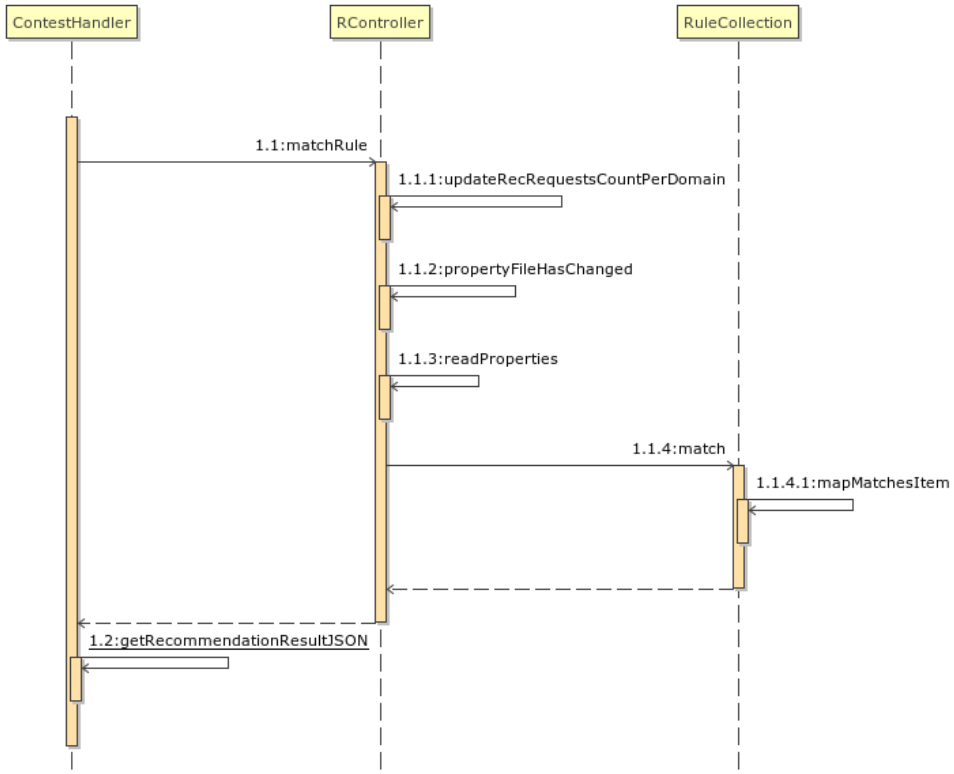
- Zaoberala sa implementáciou baseline algoritmu
- Popísala zmeny, ktoré nastali oproti pôvodnému programu
- Popísala algoritmus, ktorý je použitý na odporúčanie

[30] `http://junit.org/junit4/`

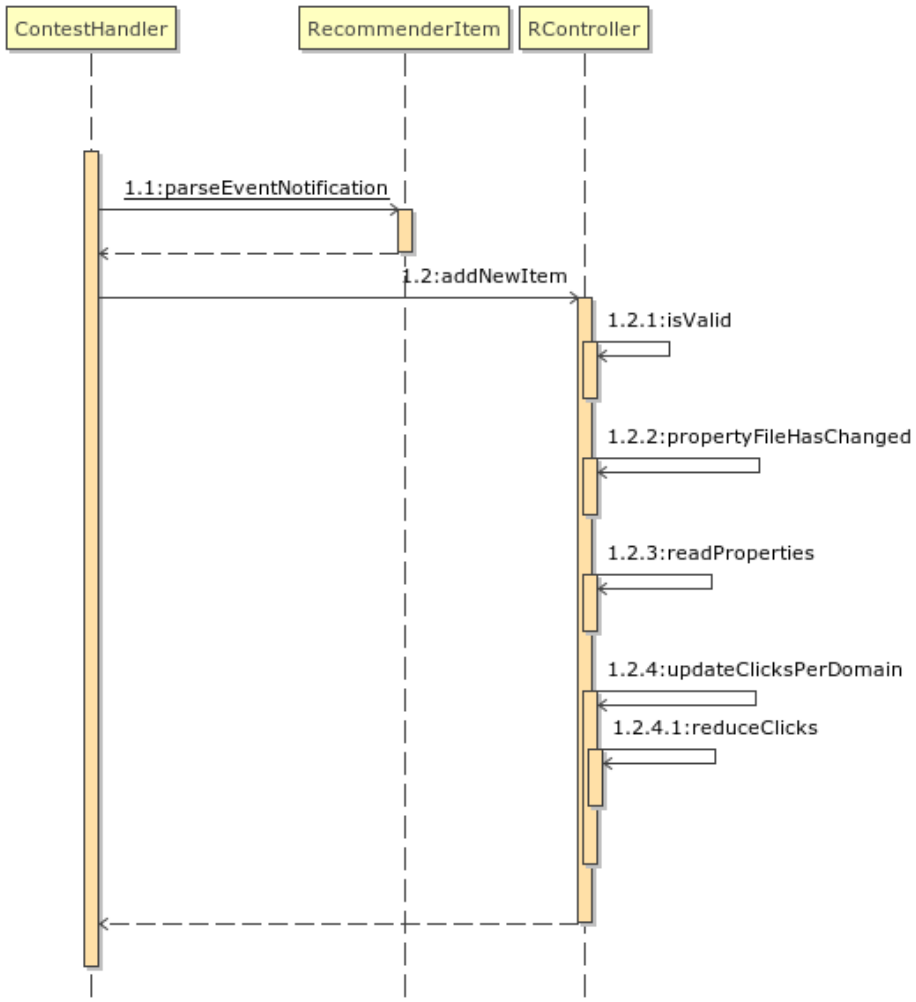


Obr. 5.8: Program rule-recommender - triedny diagram.

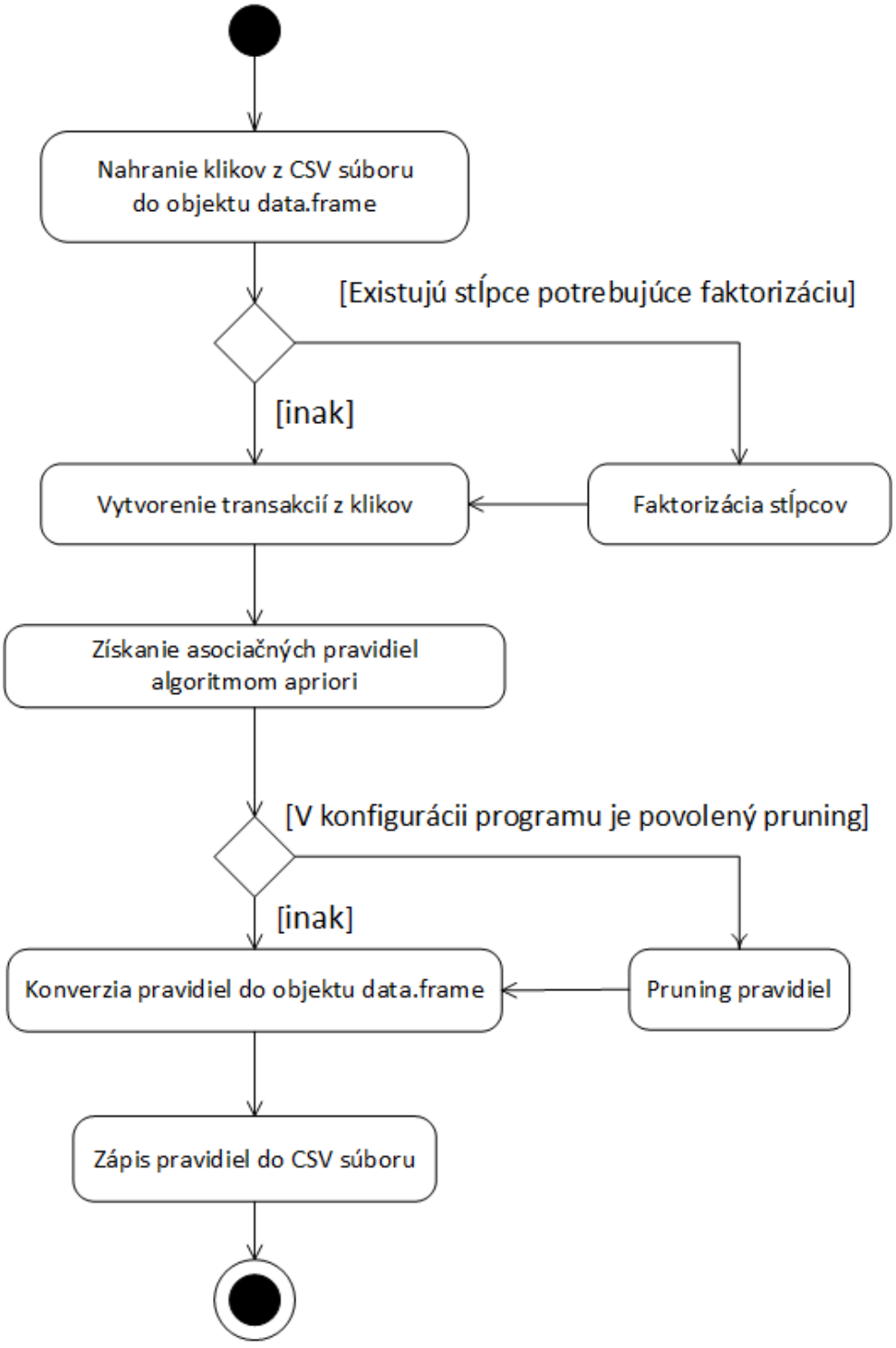
- Venovala sa serveru `Rserve` a knižniciam `arules` a `rCBA`
- Popísala proces testovania programu



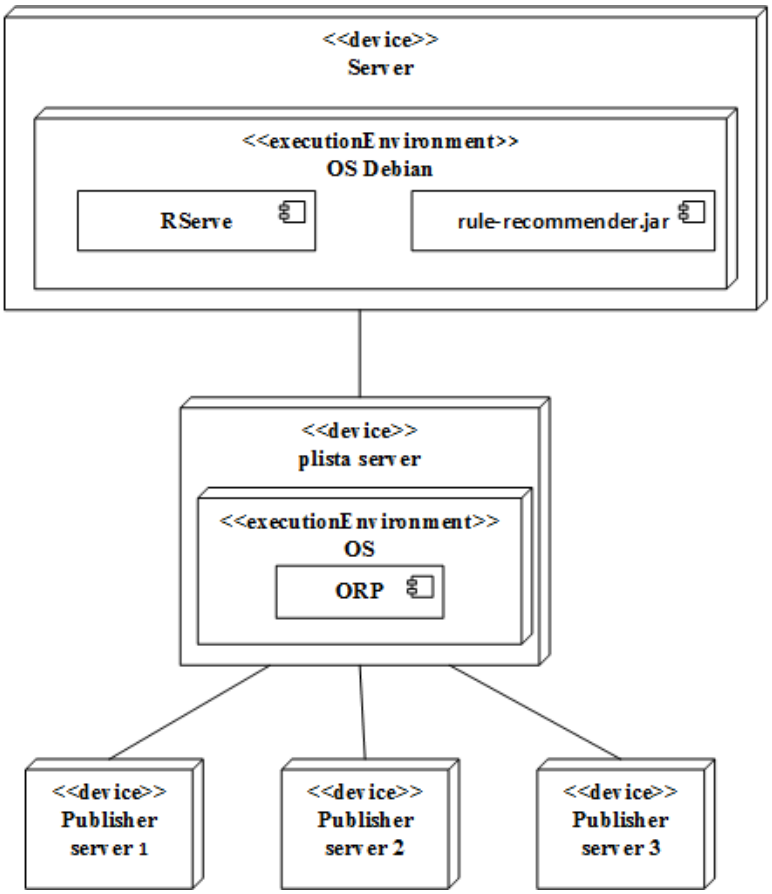
Obr. 5.9: Žiadosť o odporúčanie - sekvenčný diagram.



Obr. 5.10: Pridanie novej položky - sekvenčný diagram.



Obr. 5.11: Proces získavania pravidiel



Obr. 5.12: Diagram nasadenia.

Časť III

Experimentálna časť

Kapitola 6

Merania

Pri príprave môjho programu na vyhodnocovaciu časť výzvy CLEF NewsREEL som experimentoval z rôznymi hodnotami. Jednalo sa o experimenty s parametrami programu ako napríklad maximálny počet klikov alebo interval aktualizovania pravidiel, ale aj parametry algoritmu ťažiaceho pravidlá ako minimálna podpora, minimálna dôvera, minimálna a maximálna dĺžka vyťažených pravidiel.

Výsledky a domnienky, ktoré vyplynuli z týchto pravidiel som sa rozhodol otestovať na väčšom datasete. Táto kapitola sa zaoberá popisom týchto experimentov a popisom týchto výsledkov.

6.1 Plista - malý dataset

6.1.1 Úvod

Na offline vyhodnocovanie som použil prvých 100 000 riadkov datasetu poskytnutého spoločnosťou plista. Takýto upravený dataset sa skladá zo šiestich domén a ich zloženie je nasledovné:

- `Doména 418` (celkovo 5223 správ)

  event_notification: 71 správ

  recommendation_request: 5151 správ

  item_update: 1 správa

- `Doména 13554` (celkovo 6400 správ)

  event_notification: 0 správ

  recommendation_request: 6400 správ

  item_update: 0 správ

- `Doména 694` (celkovo 1808 správ)

event_notification: 6 správ

recommendation_request: 1802 správ

item_update: 0 správ

- `Doména 3336` (celkovo 1 správa)

  event_notification: 0 správ

  recommendation_request: 1 správa

  item_update: 0 správ

- `Doména 1677` (celkovo 10041 správ)

  event_notification 321 správ

  recommendation_request 9660 správ

  item_update 60 správ

- `Doména 35774` (celkovo 76257 správ)

  event_notification 1061 správ

  recommendation_request 75458 správ

  item_update 8 správ

Môžme teda vidieť, že nie všetky domény obsahujú všetky typy správ. Zo šiestich domén sú to len tri. Okrem toho, dve domény obsahujú len žiadosti o odporúčanie bez klikov, z ktorých by bolo možné nejaké odporúčanie vytvoriť. Zaujímavé je, že žiadosti o odporúčanie výrazne prevyšujú správy o kliknutí.

Časová značka prvej správy v tomto datasete je `2016-01-21 16:15:18`. Posledná správa má časovú značku `2016-01-31 23:34:59`. Jedná sa teda o správy z desiatich dní. Správy ale nie sú rozdelené rovnomerne medzi jednotlivé dni. 99.96 percent pochádza z posledného dňa. Okrem toho, väčšina správ z posledného dňa pochádza z večerných hodín (22. a 23.). V podstate sa teda jedná o záznam správ z jedného večera jedného dňa.

Parametry, s ktorých hodnotami som sa rozhodol experimentovať sú: *frekvencia aktualizovania pravidiel*, *dôvera*, *podpora*, *maximálna dĺžka pravidla*, *maximálny počet klikov*, *pruning* a *atribúty*.

Ako metriku som zvolil offline click-through rate. Hodnoty offline CTR pri použití základného baseline algoritmu pre jednotlivé domény sú uvedené v tabuľke 6.1.

Merania prebehli na laptope s procesorom Intel Core^{TM} i3 3217U, pamäťou 8GB a operačnom systéme Ubuntu 16.04. Jedno meranie trvalo medzi dvomi a štyrmi minútami.

Počiatočné hodnoty parametrov pre odporúčanie založené na asociačných pravidlách sú uvedené v tabuľke 6.2 a ich výsledky v tabuľke 6.3.

Doména	**CTR**
418	0.07%
694	0%
1677	0.4%
3336	0%
13554	0%
35774	1.3%
celkovo	1.04%

Tabuľka 6.1: CTR (malý dataset) - baseline algoritmus.

Parameter	**Hodnota**
Maximálny počet položiek	30 000
Maximálna dĺžka pravidla	6
Dôvera	2%
Podpora	0.5%
Frekvencia aktualizovania pravidiel	15 sekúnd
Pruning	povolený

Tabuľka 6.2: Počiatočné hodnoty parametrov

Doména	**CTR**
418	0.14%
694	0%
1677	0.5%
3336	0%
13554	0%
35774	1.62%
celkovo	1.29%

Tabuľka 6.3: CTR (malý dataset) - odporúčanie na základe pravidiel s počiatočnými hodnotami

Doména	5s	7s	30s	60s
418	0.12%	0.12%	0.12%	0.07%
694	0%	0%	0%	0%
1677	0.39%	0.46%	0.41%	0.43%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
35774	1.67%	1.65%	1.58%	1.44%
celkovo	1.31%	1.32%	1.26%	1.15%

Tabuľka 6.4: CTR (malý dataset) - rôzne hodnoty intervalu

6.1.2 Frekvencia

Prvým parametrom, ktorý som skúšal meniť bola frekvencia aktualizovania pravidiel. Vlákno, ktoré má na starosť získavanie nových asociačných pravidiel pre každú doménu sa po vykonaní tejto činnosti uspí na určitý časový úsek. Je to práve tento časový úsek, ktorého hodnoty som menil. Výsledky CTR pre štyri rôzne hodnoty sú v tabuľke 6.4.

Ako môžme vidieť, celkové CTR po dosiahnutí určitej hodnoty začne klesať. Tento jav má jednoduché vysvetlenie. Pre príliš dlhý časový úsek už posielanie správ z datasetu dobehne, ale nestihnú sa vytvoriť pravidlá z pribudnutých klikov. Pre príliš kratký časový úsek sa zas pravidlá aktualizujú zbytočne často.

6.1.3 Dôvera a podpora

Nasledujúcimi parametrami, ktoré som skúšal meniť boli hodnoty vstupujúce do algoritmu apriori. Jednalo sa o minimálne hodnoty dôvery a podpory definované v sekcii 2.3.4. Hodnota minimálnej dôvery 0.5 percenta znamená, že sa jednalo o počet záznamov medzi 2 (minimálny počet klikov) a 150 (maximálny počet klikov). Pri hodnote minimálnej dôvery 1 percenta sa jednalo o minimálne 5 záznamov a pri 2 percentách o 10 záznamov.

Výsledky (dôvera označená ako c, podpora ako s) môžme vidieť v tabuľke 6.5.

Tieto výsledky nie je možné interpretovať tak jednoducho ako v predošlom prípade. Najlepšie výsledky dosahuje program pre päťpercentnú dôveru a dvojpercentnú podporu- Platí ale, že čím je nižšia minimálna možná hodnota podpory, tým viac transakcií ju spĺňa. Získame teda väčší počet pravidiel. Na druhej strane by sme mali snažiť o čo najvyššiu možnú mieru dôvery. Od vyššej hodnoty dôvery očakávame, že odfiltruje časť pravidiel, ktorá by buď nebola použitá alebo by odporučila nesprávne položky.

Doména	**c:1%, s:0.5%**	**c:1%, s:1%**	**c:2%, s:0.5%**	**c:5%, s:2%**
418	0.14%	0.16%	0.13%	0.18%
694	0%	0%	0%	0%
1677	0.45%	0.43%	0.45%	0.46%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.62%	1.59%	1.60%	1.67%
celkovo	1.29%	1.27%	1.28%	1.34%

Tabuľka 6.5: CTR (malý dataset) - rôzne hodnoty dôvery a podpory

Doména	**2**	**8**	**10**	**12**
418	0.16%	0.15%	0.13%	0.13%
694	0%	0%	0%	0%
1677	0.42%	0.43%	0.42%	0.43%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.66%	1.58%	1.60%	1.62%
celkovo	1.32%	1.26%	1.30%	1.29%

Tabuľka 6.6: CTR (malý dataset) - rôzne hodnoty maximálnej možnej dĺžky pravidla

6.1.4 Maximálna dĺžka pravidla

Maximálna dĺžka pravidla je parameter, ktorý takisto vstupuje do apriori algoritmu, ktorý asociačné pravidlá získava. CTR pre rôzne hodnoty môžme vidieť v tabuľke 6.6.

CTR sa zvyšuje s väčšou maximálnou dĺžkou pravidla, ale len do určitého bodu - tým sa zdá byť číslo 6. To by znamenalo, že väčšina pasujúcich pravidiel, ktoré boli použité na odporučenie sú dĺžky 1 alebo 2. To potvrdzuje aj pohľad do logov programu. Zväčšenie maximálnej dĺžky pridá zopár chýbajúcich pravidiel väčšej dĺžky (3 a viac), ktoré budú použité.

6.1.5 Počet klikov

Parametrom počet klikov je myslený maximálny možný počet klikov, ktorý sa používa na výpočet asociačných pravidiel. Po prekročení tohto počtu sa kliky premažú a zachová sa len časť z nich (tento pomer je možné nastaviť ako v parametroch programu). Výsledky sa nachádzajú v tabuľke 6.7.

Výsledky možno interpretovať tak, že premazávanie klikov pri prekročení ich maximálnej hranice nespôsobuje problémy so spracovaním a aplikácia stihne odpovedať. Problematickou je skutočnosť, že so zvyšujúcim sa počtom

Doména	5000	10 000	50 000	100000
418	0.14%	0.17%	0.12%	0.14%
694	0%	0%	0%	0%
1677	0.41%	0.42%	0.43%	0.40%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.65%	1.61%	1.65%	1.65%
celkovo	1.31%	1.28%	1.31%	1.31%

Tabuľka 6.7: CTR (malý dataset) - rôzne hodnoty maximálneho počtu klikov

Doména	CTR
418	0.18%
694	0%
1677	0.53%
3336	0%
13554	0%
37554	1.75%
celkovo	1.4%

Tabuľka 6.8: CTR (malý dataset) - zákazaný pruning

klikov sa zvyšuje aj počet pravidiel. To zaberie viac času pri ich samotnom získavaní, ale hlavne pri ich následnom filtrovaní.

6.1.6 Pruning

Filtrovanie získaných pravidiel pruningom bolo spomenuté v kapitole 5. CTR bez použitia pruningu je vidieť v tabuľke 6.8.

Zakázanie pruningu, zdá sa, prináša lepšie výsledky. Treba ale vziať do úvahy fakt, že počet pravidiel stúpne rádovo. Napríklad pre doménu 37554 sa so stovky pravidiel stanú desaťtisíce. Okrem toho, objavili sa situácie, v ktorých program nestihol odpovedať v časovom limite. Toto je ale dôležitý faktor pri online vyhodnocovaní.

6.1.7 Atribúty

Počet atribútov položky a to, ktoré atribúty budú vybraté, sa stali tiež predmetom experimentovania. Výsledky sú v tabuľke 6.9.

Pre počet atribútov tri boli zvolené identifikátor položky, identifikátor kategórie článku a identifikátor kľúčového slova s najčastejším výskytom obsiahnutého v článku. Rozdiel medzi stĺpcami *6a* a *6b* je v použitých atribútoch. Okrem identifikátorov položky, kategórie a kľúčového slova to boli v prvom

Doména	**3**	**6a**	**6b**	**9**
418	0.12%	0.20%	0.15%	0.16%
694	0%	0%	0%	0%
1677	0.43%	0.40%	0.42%	0.41%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.62%	1.58%	1.56%	1.55%
celkovo	1.29%	1.26%	1.24%	1.25%

Tabuľka 6.9: CTR (malý dataset) - rôzne hodnoty počtu atribútov

prípade poloha užívateľa, prehliadač a pozícia článku na stránke, a v druhom prípade typ zariadenia, poskytovateľ internetového pripojenia a počasie.

Z výsledkov je zaujímavé zlepšenie CTR pre doménu 418 pre šesť atribútov. Pravdepodobne je to spôsobené pravidlami obsahujúcimi novo pridané atribúty. Obecne ale výsledky tohto merania nemožno ďalej použiť. Jediným povinným atribútom článku je jeho doména. Pri pozorovaní reálnych prichádzajúcich správ u niektorých správ chýba aj ich identifikátor. Atribúty, ktoré majú teda pri offline vyhodnocovaní dobré výsledky sa pri prichádzajúcich správach teda vôbec nemusia objaviť.

6.2 Plista - veľký dataset

6.2.1 Úvod

Tvrdenia, ktoré vyplynuli z meraní v predošlej sekcii, som sa rozhodol overiť na väčšom datasete. Väčší dataset sa skladá z prvých 500 000 riadkov plista datasetu poskytnutého CLEF NewsREEL. Tento dataset sa skladá zo šiestich domén a ich zloženie je nasledovné:

- `Doména 418` (celkovo 20118 správ)

  event_notification: 334 správ

  recommendation_request: 19780 správ

  item_update: 4 správy

- `Doména 13554` (celkovo 26928 správ)

  event_notification: 0 správ

  recommendation_request: 26928 správ

  item_update: 0 správ

- `Doména 694` (celkovo 8191 správ)

  event_notification: 41 správ

Doména	**CTR**
418	0.10%
694	0%
1677	0.51%
3336	0%
13554	0%
37554	1.12%
celkovo	0.99%

Tabuľka 6.10: CTR (veľký dataset) - baseline algoritmus.

recommendation_request: 8150 správ

item_update: 0 správ

- `Doména 3336` (celkovo 3 správy)

  event_notification: 0 správ

  recommendation_request: 3 správa

  item_update: 0 správ

- `Doména 1677` (celkovo 45745 správ)

  event_notification 1398 správ

  recommendation_request 44107 správ

  item_update 60 správ

- `Doména 35774` (celkovo 76257 správ)

  event_notification 1061 správ

  recommendation_request 75458 správ

  item_update 240 správ

Prvý malý dataset je podmnožinou veľkého. Vo veľkom datasete sa ale opakujú zvláštnosti malého. Tri domény obsahujú všetky typy správ a počet žiadostí o odporúčanie výrazne prevyšuje správy o klikoch a aktualizovaní položiek.

Časová značka prvej správy v tomto datasete je `2016-01-21 16:15:18`. Posledná správa má časovú značku `2016-02-01 05:24:24`. V tomto prípade sa jedná o správy z dvanástich dní. Správy opäť nie sú rozdelené rovnomerne. 29.96 percent pochádza z predposledného dňa a až 64.1 percenta pochádza z posledného dňa. Správy z posledného dňa pochádzajú z ranných hodín. V podstate sa teda jedná o správy z večera jedného dňa a skorých ranných hodín ďalšieho dňa.

Doména	**CTR**
418	0.14%
694	0%
1677	0.49%
3336	0%
13554	0%
37554	1.39%
celkovo	1.16%

Tabuľka 6.11: CTR (veľký dataset) - odporúčanie na základe pravidiel s počiatočnými hodnotami

Doména	**30s**	**60s**	**120s**	**180s**
418	0.14%	0.15%	0.08%	0.14%
694	0%	0%	0%	0%
1677	0.53%	0.51%	0.54%	0.52%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.37%	1.34%	1.32%	1.27%
celkovo	1.14%	1.12%	1.11%	1.07%

Tabuľka 6.12: CTR (veľký dataset) - rôzne hodnoty intervalu

Merania boli vykonané na rovnakom laptope v rovnakých podmienkach ako v predošlom prípade. Výsledky baseline algoritmu sú v tabuľke 6.10 a výsledky programu v tabuľke 6.11.

6.2.2 Frekvencia

Výsledky sa nachádzajú v tabuľke 6.12. V tomto prípade sa zdá, že sa potvrdzuje domnienka z menšieho datasetu. Celkové CTR so zväčšujúcim intervalom začne klesať, pretože pre príliš dlhý časový úsek už posielanie správ z datasetu dobehne a nestihnú sa vytvoriť pravidlá z pribudnutých klikov.

6.2.3 Dôvera a podpora

Výsledky (dôvera označená ako c, podpora ako s) môžme vidieť v tabuľke 6.13. Najlepšie výsledky program dosahuje pre nižšie hodnoty podpory - konkrétne 0.5 percenta. Takáto hodnota minimálnej podpory sa pohybuje medzi 2 a 150 záznamami. V menšom datasete ale program dosahoval najlepšie výsledky pre päťpercentnú dôveru a dvojpercentnú podporu.

Doména	c:1%, s:0.5%	c:1%, s:1%	c:2%, s:0.5%	c:5%, s:2%
418	0.19%	0.21%	0.17%	0.19%
694	0%	0%	0%	0%
1677	0.49%	0.48%	0.49%	0.56%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.41%	1.35%	1.39%	1.34%
celkovo	1.17%	1.11%	1.16%	1.12%

Tabuľka 6.13: CTR (veľký dataset) - rôzne hodnoty dôvery a podpory

Doména	2	8	10	12
418	0.17%	0.16%	0.16%	0.16%
694	0%	0%	0%	0%
1677	0.49%	0.49%	0.49%	0.49%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.36%	1.37%	1.37%	1.37%
celkovo	1.13%	1.14%	1.14%	1.14%

Tabuľka 6.14: CTR (veľký dataset) - rôzne hodnoty maximálnej možnej dĺžky pravidla

6.2.4 Maximálna dĺžka pravidla

Výsledky sa nachádzajú v tabuľke 6.14. V tomto prípade to vyzerá tak, že sa opäť potvrdili výsledky z menšieho datasetu. CTR sa zvyšuje s väčšou maximálnou dĺžkou pravidla, ale len do určitého bodu - tým sa zdá byť číslo 6. Pravidlami dosahujúce najlepšie výsledky sú zrejme krátke pravidlá buď s prázdnou ľavou stranou alebo ľavou stranou obsahujúcou jeden alebo dva atribúty položky.

6.2.5 Počet klikov

Výsledky sa nachádzajú v tabuľke 6.15. Vyšší maximálny počet klikov nespôsobuje problémy so spracovaním. Program dokáže spracovať aj väčšie množstvo klikov a vytvoriť z nich pravidlá. Na druhej strane, väčší počet klikov neprináša vyšší CTR.

6.2.6 Pruning

CTR bez použitia pruningu je vidieť v tabuľke 6.16. Zakázanie pruningu síce prinieslo lepší CTR, ale za cenu zvýšena počtu chybných odpovedí. Vyhodnocovanie bez pruningu navyše trvalo podstatne dlhšie ako s ním.

Doména	**5000**	**10 000**	**50 000**	**100000**
418	0.18%	0.18%	0.17%	0.16%
694	0%	0%	0%	0%
1677	0.46%	0.48%	0.47%	0.51%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.38%	1.36%	1.40%	1.41%
celkovo	1.15%	1.14%	1.18%	1.16%

Tabuľka 6.15: CTR (veľký dataset) - rôzne hodnoty maximálneho počtu klikov

Doména	**CTR**
418	0.28%
694	0%
1677	0.9%
3336	0%
13554	0%
37554	1.75%
celkovo	1.5%

Tabuľka 6.16: CTR (veľký dataset) - zákazaný pruning

Doména	**3**	**6a**	**6b**	**9**
418	0.19%	0.20%	0.20%	0.21%
694	0%	0%	0%	0%
1677	0.51%	0.49%	0.49%	0.47%
3336	0%	0%	0%	0%
13554	0%	0%	0%	0%
37554	1.14%	1.13%	1.14%	1.13%
celkovo	1.18%	1.16%	1.13%	1.14%

Tabuľka 6.17: CTR (veľký dataset) - rôzne hodnoty počtu atribútov

6.2.7 Atribúty

Výsledky sa nachádzajú v tabuľke 6.17. Výsledky pre doménu 418, ktoré boli na malých dátach lepšie pre šesť atribútov tu sú približne rovnaké ako u iných počtov. Z dôvodov už spomenutých pri meraní na malých dátach som sa preto rozhodol vo vyhodnocovacom období použiť čo najviac atribútov.

Tím	CTR	počet klikov	počet odporúčaní
WIRG	0.55%	122	21849
BL2beat	0.48%	89	18505
A	0.37%	116	31032
B	0.35%	87	24591
C	0.33%	56	16715
D	0.30%	106	34224
E	0.30%	52	17029
F	0.27%	18	6647
G	0.25%	56	22007
H	0.23%	56	24305
I	0%	0	0
J	0%	0	89
K	0%	0	0

Tabuľka 6.18: Druhé testovacie obdobie - CTR

6.3 Online

V kapitole 3 bolo uvedené, že výzva CLEF NewsREEL je rozdelená do troch fáz: dve testovacie fázy a vyhodnocovacej fázy.

Program *rule-recommender* sa zúčastnil druhej testovacej fázy. V dobe písania tejto práce ešte neboli k dispozícii ucelené výsledky od organizátorov, a tak nasledujúce údaje pochádzajú z priebežného leaderboadu poskytnutého organizátormi.

Druhá testovacia fáza trvala od 27. marca 2017 do 3. apríla 2017. Celého testovacieho obdobia sa zúčastnilo dvanásť algoritmov. Počas týchto ôsmich dní boli nasadené rôzne verzie môjho programu. To bolo väčšinou z dôvodu odstránenia chýb, ktoré sa prejavili až pri ostrej prevádzke.

CTR sa pohybovala medzi 0.5 až 0.59 percenta. Výsledky z 10. apríla sa nachádzajú v tabuľke 6.18. *BL2beat* je označenie baseline algoritmu. Mená ostatných tímov sú anonymizované.

V tomto období sa program (tím *WIRG*) držal väčšinou na prvom mieste. Výnimkou boli situácie, keď nejaký algoritmus obdržal malý počet žiadosti o odporúčanie, dosiahol relatívne vysoký počet klikov a prestal poskytovať odporúčania.

Čo sa týka miery odozvy, bohužiaľ nemám k dispozícii presné údaje. Indikátorom by mohol byť počet správ od ORP serveru s kódom 408 reprezentujúcim timeout. Tých sa objavilo po dobu testovania málo, väčšinou pri prerušovaní a obnovovaní premávky. Myslím teda, že mierou odozvy je program podobný baseline algoritmu.

6.4 Zhrnutie

Predmetom tejto kapitoly boli experimenty s programom *rule-recommender*. Poznatky z offline vyhodnocovania:

- Na malom a veľkom plista datasete mal program lepší click-through rate ako baseline algoritmus. Response rate nemá pri offline vyhodnocovaní zmysel merať.
- Ukázalo sa, že frekvencia s akou sa prepočítavajú pravidlá z klikov ovplyvňuje click-through rate.
- Takisto sa ukázalo, že program dosahoval najlepšie výsledky pre nižšie hodnoty podpory a dôvery.
- Click-through rate sa do určitého bodu zvyšuje s maximálnou dĺžkou pravidla.

Poznatky z druhého testovacieho obdobia:

- Počas druhého testovacieho obdobia program dosiahol opäť lepší click-through rate ako baseline a väčšinou sa držal na prvom mieste.
- Presné výsledky neboli v dobe písania práce k dispozícii. Odhadom mal program približne rovnaký response rate ako baseline algoritmus.

Poznatky z vyhodnocovacieho obdobia:

- Program mal v prých dňoch mierne horšie výsledky ako baseline a celkovo sa držal približne v strede tabuľky (siedme miesto zo sedemnástich účastníkov)
- Presná hodnota respone rate v tejto chvíli nie je k dispozícii, čo sa celkového počtu poskytnutých odporúčaní týka program bol druhý.
- Tieto údaje je ale nutné brať s rezervou, keďže leaderboard obsahujúci údaje o click-through rate nebol vždy k dispozícii. Okrem toho, vyhodnocovacie obdobie sa v dobe písanie tejto práce neskončilo. Hodnoty teda pochádzajú z prvých piatich dní tohto obdobia

V sekcii 1.4 bolo definovaných niekoľko rôznych aspektov podľa ktorých možno odporúčací systém vyhodnocovať. Tieto aspekty možno v podstate brať ako nefunkčné požiadavky systému.

- **Správnosť**, sa v prípade novinových článkov meria pomocou click-through rate. V tomto bode dosahoval program v offline vyhodnocovaní a v druhom testovacom období lepšie výsledky ako baseline algoritmus.

- **Pokrytie** vecí nie je úplné. Položky, ktoré môžu byť odporučené sa vyberajú z klikov, ktoré sú uchované. Maximálny počet týchto klikov je určený parametrom programu. Uchovávať údaje o väčšom počte klikov nie je podľa môjho názoru možné kvôli pamäti. S touto skutočnosťou súvisí aj nižšia **diverzita**.

- **Dôvera** systému v poskytnuté odporúčania je postavená na asociačných pravidlách. Tie sú vytvorené z určitými hodnotami dôvery a podpory. Zjednodušene povedané, musí nastať určitý počet interakcií aby boli o informácie o nich použité na odporúčanie. Vďaka je tomu je zaručená aj **stabilita** systému.

- Pri kliknutí na novinový článok užívateľ nemusí obetovať ani veľké množstvo času ani peniaze. Myslím, že z tohto dôvodu merať **dôveryhodnosť** nemá význam.

- Keďže položky sa odporúčajú z pravidiel vytvorených z určitého počtu posledných klikov, **novosť** považujem za zaručenú.

- **Prekvapenie** môžu poskytnúť pravidlá, ktorých ľavú stranu tvoria veci ako počasie v lokalite užívateľa alebo jeho príjem.

- **Riziko** spôsobené zlým odporúčaním je podľa môjho názoru nízke. Užívateľ môže prísť nanajvýš o pár minút času.

- **Robustnosť**. Odporúčané položky pochádzajú od toho istého vydavateľa ako článok na ktorom sa užívateľ práve nachádza. Nemôže teda nastať situácia, že užívateľ by sa preklikol ku konkurencii.

- **Miera učenia**. Pravidlá, na základe sa odporúčania poskytujú sú aktualizované každú polhodinu z najnovších klikov. To je spôsob akým sa systém adaptuje na zmenu preferencií užívateľov.

- Počas testovacieho obdobia program zvládol reálnu prevádzku s väčším počtom užívateľov. **Škálovateľnosť** systému by ďalej mohla byť realizovaná rozdelením na niekoľko častí pre každú doménu.

- Údaje o interakciách sú reprezentované číselnými identifikátormi. Navyše užívatelia si pri čítaní novinových správ väčšinou nevytvárajú profil a toho pomáha pri ochrane ich **súkromie**.

- O **použiteľnosti** systému sa nedá hovoriť v celku - stránky jednotlivých novinových portálov sa líšia. Predpokladám, že odporúčané položky sa zobrazia nejakým spôsobom pri pôvodnom článku. Na jednotlivých novinových portáloch je aj spôsob akým berú do úvahy **preferencie užívateľov**.

Záver

Cieľom tejto práce bolo navrhnúť a implementovať odporúčací systém v Jave pre novinové správy na spravodajských serveroch. Výsledkom je program *rule-recommender*, ktorý je založený na asociačných pravidlách.

Odporúčanie novinových správ má oproti klasickému odporúčaniu kníh alebo filmov niekoľko špecifík. Z tohto dôvodu som sa najprv snažil odporúčacie systémy a pojmy s nimi súvisiace definovať. Následne som sa venoval odporúčaniu novinových správ a požiadavkam, ktoré sú na takéto systémy kladené. Venoval som sa aj asociačným pravidlám a metrikám, ktoré sa na porovnávanie asociačných pravidiel používajú.

Existuje mnoho odlišných prístupov na odporúčanie novinových správ. Túto skutočnosť som sa snažil demonštrovať analýzou existujúcich riešení, ktoré zahŕňajú klasické algoritmy ako je kolaboratívne filtrovanie, cez prístupy postavené na ľudskom správaní, až po použitie technológií, ktoré by mali urýchliť spracovanie správ.

Program *rule-recommender* sa zúčastnil výzvy CLEF NewsREEL, ktorá si pri odporúčaní novinových správ dáva za cieľ premostiť medzeru medzi akademickou pôdou a komerčnou sférou tým, že odporúčacie systémy porovnáva na základe výsledkov, ktoré preukážu v ostrej prevádzke. Z tohto dôvodu na program boli kladené aj požiadavky z reálneho sveta, ako napríklad vysoká dostupnosť a spracovanie väčšieho množstva požiadavkov. Takýmito požiadavkami som sa musel pri implementácii zaoberať.

Za prínos mojej práce považujem jej experimentálnu časť, v ktorej experimentujem s rôznymi parametrami algoritmu použitého na vytváranie pravidiel, ale aj programu samotného. Pre tieto merania bolo použité offline vyhodnocovanie NewsREEL Replay, keďže ucelené výsledky z online vyhodonocovania neboli v dobe písania práce k dispozícii.

Väčšina systémov s najlepšími výsledkami v predošlých ročníkoch výzvy bola založená na odporúčaní najčítanejších alebo najnovších správ. Vzhľadom k tomu ukazuje odporúčanie na základe novinových správ sľubný potenciál.

Literatúra

[1] Ricci, F.; Rokach, L.; Shapira, B.; aj. (editoři): *Recommender Systems Handbook*. Springer, 2011, ISBN 978-0-387-85819-7.

[2] Robillard, M. P.; Maalej, W.; Walker, R. J.; aj.: *Recommendation Systems in Software Engineering*. Springer Publishing Company, Incorporated, 2014, ISBN 3642451349, 9783642451348.

[3] Aggarwal, C. C.: *Recommender Systems: The Textbook*. Springer Publishing Company, Incorporated, první vydání, 2016, ISBN 3319296574, 9783319296579.

[4] Avazpour, I.; Pitakrat, T.; Grunske, L.; aj.: Dimensions and Metrics for Evaluating Recommendation Systems. In *Recommendation Systems in Software Engineering*, editace M. P. Robillard; W. Maalej; R. J. Walker; T. Zimmermann, Springer, 2014, ISBN 978-3-642-45134-8, s. 245–273.

[5] Simon, F.; Steinbruckner, F.; Lewerentz, C.: Metrics based refactoring. In *Software Maintenance and Reengineering, 2001. Fifth European Conference on*, IEEE, 2001, ISBN 0-7695-1028-0, s. 30–38.

[6] Tintarev, N.; Masthoff, J.: A Survey of Explanations in Recommender Systems. In *ICDE'07: Workshop on Recommender Systems and Intelligent User Interfaces*, ICDEW '07, IEEE Computer Society, 2007, ISBN 978-1-4244-0831-3, s. 801–810.

[7] O'Mahony, M.; Hurley, N.; Kushmerick, N.; aj.: Collaborative recommendation: A robustness analysis. *ACM Trans. Inter. Tech.*, ročník 4, č. 4, Listopad 2004: s. 344–377, ISSN 1533-5399.

[8] Das, A. S.; Datar, M.; Garg, A.; aj.: Google news personalization: scalable online collaborative filtering. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-654-7, s. 271–280.

[9] Frankowski, D.; Cosley, D.; Sen, S.; aj.: You are what you say: privacy risks of public mentions. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA: ACM, 2006, s. 565–572.

[10] Dwork, C.: Differential privacy: A survey of results. In *In Theory and Applications of Models of Computation*, Springer, 2008, s. 1–19.

[11] Kohavi, R.; Longbotham, R.; Sommerfield, D.; aj.: Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, ročník 18, č. 1, 2009: s. 140–181, ISSN 1573-756X.

[12] Kille, B.; Lommatzsch, A.; Brodt, T.: *News Recommendation in Real-Time*. Cham: Springer International Publishing, 2015, ISBN 978-3-319-14178-7, s. 149–180.

[13] Doychev, D.; Rafter, R.; Lawlor, A.; aj.: *News Recommenders: Real-Time, Real-Life Experiences*. Cham: Springer International Publishing, 2015, ISBN 978-3-319-20267-9, s. 337–342.

[14] Hahsler, M.; Gruen, B.; Hornik, K.: arules – A Computational Environment for Mining Association Rules and Frequent Item Sets. *Journal of Statistical Software*, ročník 14, č. 15, October 2005: s. 1–25, ISSN 1548-7660.

[15] Agrawal, R.; Imieliński, T.; Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, New York, NY, USA: ACM, 1993, ISBN 0-89791-592-5, s. 207–216.

[16] Hopfgartner, F.; Brodt, T.; Seiler, J.; aj.: Benchmarking News Recommendations: The CLEF NewsREEL Use Case. *SIGIR Forum*, ročník 49, č. 2, 2015: s. 129–136.

[17] Hopfgartner, F.; Kille, B.; Lommatzsch, A.; aj.: Benchmarking News Recommendations in a Living Lab. In *CLEF'14: Proceedings of the 5th International Conference of the CLEF Initiative*, LNCS, Springer Verlag, 09 2014, s. 250–267.

[18] Said, A.; Lin, J.; Bellogín, A.; aj.: A Month in the Life of a Production News Recommender System. In *Proceedings of the 2013 Workshop on Living Labs for Information Retrieval Evaluation*, LivingLab '13, New York, NY, USA: ACM, 2013, ISBN 978-1-4503-2420-5, s. 7–10.

[19] Brodt, T.; Hopfgartner, F.: Shedding Light on a Living Lab: The CLEF NEWSREEL Open Recommendation Platform. In *IIiX'14: Proceedings of the Information Interaction in Context Conference*, ACM, 08 2014, s. 223–226.

[20] Scriminaci, M.; Lommatzsch, A.; Kille, B.; aj.: Idomaar: A Framework for Multi-dimensional Benchmarking of Recommender Algorithms. In *Proceedings of the Poster Track of the 10th ACM Conference on Recommender Systems (RecSys 2016), Boston, USA, September 17, 2016.*, 2016. Dostupné z: `http://ceur-ws.org/Vol-1688/paper-14.pdf`

[21] Domann, J.; Meiners, J.; Helmers, L.; aj.: Real-time News Recommendations using Apache Spark. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016.*, 2016, s. 628–641. Dostupné z: `http://ceur-ws.org/Vol-1609/16090628.pdf`

[22] Ciobanu, A.; Lommatzsch, A.: Development of a News Recommender System based on Apache Flink. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016.*, 2016, s. 606–617. Dostupné z: `http://ceur-ws.org/Vol-1609/16090606.pdf`

[23] Corsini, F.; Larson, M.: CLEF NewsREEL 2016: Image based Recommendation. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016.*, 2016, s. 618–827. Dostupné z: `http://ceur-ws.org/Vol-1609/16090618.pdf`

[24] Lommatzsch, A.; Werner, S.: Optimizing and Evaluating Stream-based News Recommendation Algorithms. In *Proceedings of the Sixth International Conference of the CLEF Association, CLEF'15*, LNCS, vol. 9283, Heidelberg, Germany: Springer, 2015, s. 376–388.

[25] Lommatzsch, A.: Real-Time News Recommendation Using Context-Aware Ensembles. In *Advances in Information Retrieval*, *Lecture Notes in Computer Science*, ročník 8416, Springer International Publishing, 2014, s. 51–62.

[26] Kliegr, T.; Kuchar, J.: Benchmark of Rule-Based Classifiers in the News Recommendation Task. In *Proceedings of the Sixth International Conference of the CLEF Association, CLEF'15*, 2015, s. 130–141.

[27] Kille, B.; Hopfgartner, F.; Brodt, T.; aj.: The plista Dataset. In *NRS'13: Proceedings of the International Workshop and Challenge on News Recommender Systems*, ICPS, ACM, 10 2013, s. 14–22.

[28] Vojir, S.; Zeman, V.; Kuchar, J.; aj.: EasyMiner/R Preview: Towards a Web Interface for Association Rule Learning and Classification in R. In *Proceedings of the RuleML 2015 Challenge, the Special Track on Rule-based Recommender Systems for the Web of Data, the Special Industry*

Track and the RuleML 2015 Doctoral Consortium hosted by the 9th International Web Rule Symposium (RuleML 2015), Berlin, Germany, August 2-5, 2015., 2015.

Dodatok **A**

Plista - veľký dataset 2

Tvrdenia o dopade veľkosti intervalu aktualizácie pravidiel a dopade maximálnej dĺžky pravidla na click-through rate som sa rozhodol overiť ešte na jednom data sete. Tento data set tvorí posledných 500 000 riadkov plista data setu. Výsledky sa nachádzajú v tabuľke A.1 a tabuľke A.2.

Doména	**60s**	**120s**	**180s**
418	0.2%	0.18%	0.16%
694	0%	0%	0%
1677	0.90%	0.91%	0.97%
3336	0%	0%	0%
13554	0%	0%	0%
37554	1.27%	1.25%	1.14%
celkovo	1.0%	1.1%	0.94%

Tabuľka A.1: CTR (veľký dataset 2) - rôzne hodnoty počtu atribútov

Doména	**2**	**10**	**12**
418	0.2%	0.26%	0.19%
694	0%	0%	0%
1677	0.9%	0.92%	0.93%
3336	0%	0%	0%
13554	0%	0%	0%
37554	1.27%	1.25%	1.29%
celkovo	1.00%	1.01%	1.03%

Tabuľka A.2: CTR (veľký dataset 2) - rôzne hodnoty maximálnej dĺžky pravidla

Dodatok **B**

Zoznam použitých skratiek

CSV Comma-separated values

CLEF Conference and Labs of the Evaluation Forum

FIFO First In First Out

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

MAE Mean Absolute Error

NewsREEL News Recommendation Evaluation Lab

ORP Open Recommendation Platform

RMSE Root Mean Square Error

Dodatok C

Inštalačná príručka

Spustenie programu *rule-recommender* vyžaduje:

- `Java 1.7`
- `R`
- knižnicu `arules`
- knižnicu `rCBA`
- prostredie `RServe`

Program je možné spustiť skriptom `run-rule-recommender.sh`, ktorý sa spoločne s jar súborom nachádza v zložke `exe`.

Dodatok D

Obsah priloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `exe` ..... spustiteľná verzia programu
- `src`
  - `impl` ..... zdrojové kódy implementácie
  - `thesis` ..... zdrojová forma práce vo formáte LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce vo formáte PDF