Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5208-5718

Bc. Samuel Molnár

Identifikácia vyhľadávacích epizód založená na aktivite používateľa

Diplomová práca

Vedúci práce: Ing. Tomáš Kramár, PhD.

Máj, 2015

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5208-5718

Bc. Samuel Molnár

Identifikácia vyhľadávacích epizód založená na aktivite používateľa

Diplomová práca

Študijný program: Informačné systémy
Študijný odbor: 9.2.6 Informačné systémy
Miesto vypracovania: Bratislava
Vedúci práce: Ing. Tomáš Kramár, PhD.

Máj, 2015

Anotácia

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Informačné systémy
Autor: Bc. Samuel Molnár
Diplomová práca: Identifikácia vyhľadávacích epizód založená na aktivite používateľa
Vedúci práce: Ing. Tomáš Kramár, PhD.
Máj, 2015

Pre identifikáciu vyhľadávacích epizód existuje viacero prístupov, ktoré sa vo veľkej miere zameriavajú na využitie lexikálnych vlastností dopytov a času medzi zadaním dopytov. V našej práci sme sa zamerali na využitie spoločných výsledkov vyhľadávania medzi dopytmi a aktivity používateľa na daných výsledkoch ako mieru ich vzájomnej podobnosti. Ako aktivitu používateľa sme uvažovali čas, ktorý používateľ strávi prezeraním konkrétneho výsledku. Na základe uvedených vlastností sme navrhli model strojového učenia pozostávajúci z časových atribútov, lexikálnych vlastností a miery spoločných výsledkov spolu s aktivitou používateľa. Náš model sme overeli pomocou dát z vlastného vyhľadávača, ktorý obsahoval dopyty zaradené do vyhľadávacích epizód priamo od používateľov. Overenie preukázalo, že naša metóda je úspešnejšia pri identifikácii vyhľadávacích epizód ako základné metódy uvažujúce len čas. Pomocou dát z nášho vyhľadávača sme realizovali aj porovnanie manuálneho zaradenia dopytov do vyhľadávacích epizód pomocou externých anotátorov a zaradenia priamo od používateľov vyhľadávača. Na základe štatistických testov sme zistili, že manuálne zaradenie dopytov je horšie ako časové metódy a zaradenie priamo od používateľov a kvalita zaradania výrazne závisí od anotátorových doménových znalostí dopytov.

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Information Systems
Author: Bc. Samuel Molnár
Master thesis: Activity-based Search Session Identification
Supervisor: Ing. Tomáš Kramár, PhD.
May, 2015

Several approaches for search session identification were presented recently. Most of the approaches utilize lexical analysis of queries along with time windows to identify similarities between queries. In our work, we focus on utilizing shared results ratio between queries and user's activity on these results as a indicator of their similarity. Based on these features we proposed a machine learning model consisting of time windows, lexical attributes and shared results ratio along with user's activity. We conducted an experiment to evaluate our model's accuracy in identifying queries' search sessions on data from our search engine, which contains search sessions and queries annotated by users themselves. Our evaluation showed that our model is more accurate in identifying sessions than models based on time. As a second part of our evaluation, we conducted a number of statistical tests that show that manually annotated queries by external annotators have worse accuracy in identifying sessions than models based on time and the quality of annotations is highly dependent on annotator's domain knowledge of queries.

POĎAKOVANIE

Chcem sa poďakovať vedúcemu mojej práce, Tomášovi Kramárovi, za vedenie môjho projektu, rady, skúsenosti a všetok vynaložený čas venovaný neustálemu zlepšovaniu môjho projektu. Taktiež sa chcem poďakovať členom skupiny PeWe za hodnotnú spätnú väzbu, ktorá ovplyvnila smerovanie môjho projektu. Ďakujem aj celej svojej rodine a priateľom, ktorí mi vyjadrovali podporu počas celého môjho štúdia.

S radosťou, Samuel Molnár

ČESTNÉ PREHLÁSENIE

Čestne prehlasujem, že na diplomovej práci som pracoval samostatne na základe vlastných teoretických a praktických poznatkov, konzultácii a štúdia odbornej literatúry, ktorej úplný prehľad je uvedený v zozname použitej literatúry.

.....

Bc. Samuel Molnár

Obsah

1 Úvod

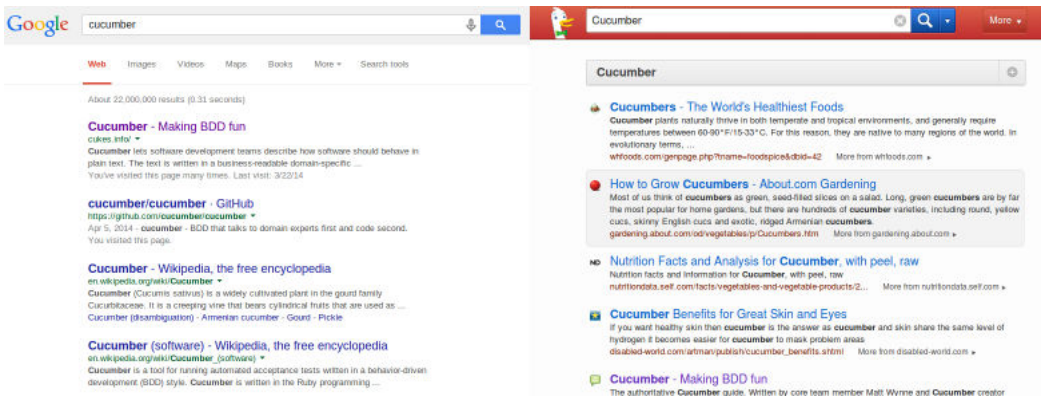
V dnešnej dobe sa web stal jedným zo základných zdrojov informácii. Vďaka rozsiahlemu obsahu dokáže uspokojiť väčšinu našich informačných potrieb. Webový obsah ale každým dňom narastá a z pôvodného problému nedostatku informácii sa stal problém prebytku informácii. Nájsť relevantné informácie v obrovskom množstve obsahu, ktorý web ponúka, sa niekedy zdá nemožné. Riešenie tohto problému ponúkajú internetové prehliadače, ktoré vedia veľmi rýchlo prehľadávať webový obsah pomocou zoznamu kľúčových slov. Keďže vyhľadávače sú momentálne najrýchlejšia forma prehliadania webu, stali jednými z najdôležitejších a súčasne aj najpoužívanejších nástrojov pre získavanie informácii na webe [32, 5, 37].

Vyhľadávač potrebuje dopyt na to, aby vedel, čo chceme nájsť. Dopytom pre tradičný internetový vyhľadávač môže byť slovo alebo zoznam slov, ktoré opisujú to, čo chceme nájsť. Vyhľadávač následne prehľadaním uložených stránok nájde tie, ktoré vyhovujú nášmu dopytu, t.j. obsahujú slová z nášho dopytu, a usporiada ich podľa relevancie. Kvôli veľkému množstvu obsahu, ktorý sa na webe nachádza, vyhľadávače mnohokrát nájdu desiatky tisíc výsledkov, ktoré musia byť rozdelené do viacerých individuálnych stránok, aby ich bolo možné rozumne prehliadať. Hlavným dôvodom tohto problému nie je len veľký informačný priestor na webe, ale fakt, že dopyty používateľov sú v mnohých prípadoch príliš vágne [45] a obsahujú prevažne bežné slová, nachádzajúce sa vo veľkom množstve stránok. K veľkému počtu relevantných výsledkov počas vyhľadávania prispieva aj fakt, že mnohé slová majú podobný význam [50, 34], ktorý vyhľadávač nevie v prípade konkrétneho používateľa určiť, čo sťažuje navigáciu vo výsledkoch vyhľadávania ešte viac.

Vyhľadávače za tento problém nenesú zodpovednosť, pretože nemajú inú možnosť ako len porovnávať text svojich zaindexovaných stránok s dopytom používateľa na základe podobnosti textu. Preto, čím stránka obsahuje viac slov z dopytu, tým sa javí relevantnejšia. Relevanciu výsledkov môže vyhľadávač rozšíriť aj o mieru popularity danej stránky – veľmi dobrým príkladom je funkcia pre výpočet relevancie PageRank [4], ktorú používa aj Google[1] pre usporadúvanie výsledkov vyhľadávania. Relevancia určená vyhľadávačom môže byť čiastočne užitočná, ale keďže je generická pre všetkých používateľov, nemusí splniť svoj účel. Preto sa funkcia PageRank aj v

[1]Google: https://google.com

prípade Google stala tiež len jedným z mnohých atribútov pre usporiadanie výsledkov. Rozdiel v relevancii výsledkov si môžeme všimnúť na príklade anglického slova *cucumber* – používateľ môže mať na mysli uhorku ako zeleninu, ale druhého používateľa môžu zaujímať výsledky relevantné pre testovací nástroj Cucumber[2]. Ako je možné vidieť pri porovnaní výsledkov personalizovaného a nepersonalizovaného vyhľadávača na obrázku 1.1, personalizovaný vyhľadávač sa prispôsobil záujmom používateľa – testovanie softvéru. Okrem viacznačnosti slov do relevantnosti výsledkov zasahuje aj samotná osobnosť používateľa. Každý človek má svoj vlastný vzťah ku niektorým významom slov – pre každého človeka na základe jeho zážitkov alebo záľub vyjadruje konkrétne slovo iné pocity. Preto dva rovnaké dopyty môžu mať diametrálne odlišný význam pre dvoch rôznych ľudí.



Obr. 1.1: Porovnanie personalizovaného vyhľadávača – Google a nepersonalizovaného – DuckDuckGo pre dopyt cucumber (slov. uhorka)

Keďže bežné vyhľadávače tieto skryté významy nevedia zachytiť, používatelia sa prispôsobili a svoje ciele sa snažia opísať pomocou viacerých kľúčových slov. V tomto prípade ale nastáva ďalší problém – používatelia niekedy nevedia presne vyjadriť svoje ciele pomocou kľúčových slov, pretože ich cieľ je buď príliš abstraktný alebo nie sú experti v danej doméne [46] a chcú sa navigovať v novom obsahu. Preto používatelia svoj dopyt často menia, rozširujú ho o ďalšie slová a skúšajú, kedy sa vyhľadávač *trafí* do relevantných výsledkov. Používatelia tým pádom v každom danom kroku viac a viac konkretizujú svoj cieľ, namiesto toho, aby túto konkretizáciu identifikoval sám vyhľadávač na základe znalostí o používateľovi.

Nedávny výskum zaznamenal, že mnohí používatelia začali častejšie tvoriť dopyty súvisiace s ich každodennými činnosťami [33], ako napríklad zistenie času premietania filmu v kine alebo nájdenie ingrediencií na varenie. V oboch prípadoch ciele používateľov väčšinou prekračujú kontext jedného dopytu a používatelia tak tvoria viacero úzko-súvisiacich dopytov. Napríklad, ak nás zaujme konkrétny film, vyhľadáme si bližšie informácie o ňom a následne hľadáme najbližšie kiná, pričom

[2]Cucumber: http://cukes.info/

naším cieľom je nájsť najbližšie kino, kde premietajú daný film. Nepersonalizovaný vyhľadávač tento súvis neidentifikuje, pretože dopyt s názvom filmu a najbližšími kinami nemá z lexikálneho hľadiska veľa spoločného. Vyhľadávač túto situáciu nedokáže zachytiť bez toho, aby sme mu explicitne vyjadrili súvislosť medzi dopytom o najbližších kinách a daným filmom. Ako by ale túto súvislosť mohol vyhľadávač zistiť implicitne, bez našej pomoci?

Používateľ svojou aktivitou počas vyhľadávania vyjadruje, ktoré výsledky sú preňho relevantné a ktoré nie. V prípade všeobecného dopytu ponúkne používateľovi vyhľadávač výsledky z viacerých oblastí, a keď si používateľ vyberie konkrétny výsledok, vyjadrí svoje preferencie k určitej oblasti a implicitne tak rozšíri význam svojho dopytu. Ak vyhľadávač túto aktivitu zachytí, vie na základe obsahu výsledku zistiť oblasť záujmu používateľa a využiť túto informáciu na personalizáciu usporiadania výsledkov alebo v našom prípade omnoho dôležitejšiu úlohu – nájdenie súvisiacich dopytov.

Súvisiace dopyty predstavujú ucelený zoznam dopytov, ktoré zdieľajú podobný alebo rovnaký cieľ. Týmto cieľom môže byť kúpa auta v autobazáre alebo nájdenie najbližšieho kina, ktoré premieta náš obľúbený film. Náročnosť identifikácie týchto dopytov je sťažená tým, že následnosť dopytov môže byť prerušená, pretože jednotlivé ciele sa prelínajú a ľudia v rýchlosti za sebou vyhľadávajú rôzne informácie. Úlohou vyhľadávača je identifikovať tieto prelínania a priradiť dopyty ku správnym cieľom. Následne dopyty, ktoré zdieľajú rovnaký cieľ, zaradím spoločne do vyhľadávacej epizódy.

Automatická identifikácia vyhľadávacích epizód je výskumný problém, ktorým sa zaoberajú viaceré práce [36, 21, 33, 28, 17, 16, 15].Väčšina prístupov sa zameriava na využitie strojového učenia [36, 21, 33, 28, 17], pričom uvažujú zoznam atribútov (angl. *features*) dopytu a kontextu používateľa pre určenie podobnosti dopytov. Jedným z najjednoduchších atribútov je čas. Na základe definovaného časového intervalu môžeme priradiť k dopytom, ktoré sa nachádzajú blízko vedľa seba, spoločný cieľ. Ak niekto vyhľadáva informácie o filme a následne hľadá najbližšie kiná v rámci poslednej pol hodiny, môžeme predpokladať, že sa snaží nájsť najbližšie kino, v ktorom hrajú daný film. Viaceré výskumy potvrdili účinnosť časového intervalu na spájanie dopytov [21, 50], ale taktiež upozornili, že časový interval ako samotný atribút nie je veľmi presný pri určovaní súvislosti dopytov, pretože lepšie výsledky dosahuje práve v spojení s inými atribútmi ako napríklad lexikálnymi vlastnosťami dopytu. Mnohé podobnosti medzi dopytmi sa dajú identifikovať čisto len porovnaním lexikálnych odlišností medzi aktuálnym a predchádzajúcimi dopytmi, pretože používatelia mnohokrát len rozširujú alebo upravujú ich predchádzajúci dopyt. Avšak, ako zakročiť v prípade, ak dopyty sú lexikálne odlišné, ale sémanticky podobné? Príkladom môže byť dopyt *Lacný Peugeot* a *Autobazáry Bratislava*, v prípade ktorých si používateľ

pravdepodobne chce kúpiť lacné nové auto v autobazári v Bratislave. Časové intervaly spolu s lexikálnou analýzou v tomto prípade zlyhávajú, a preto je potrebné zamerať sa na využitie ďalších atribútov, ktoré prezrádzajú sémantickú význam dopytu.

V tejto práci sa zaoberáme využitím výsledkov vyhľadávania a aktivity používateľa ako atribútov pre podobnosť dopytov. Podobné dopyty reprezentujú určitý obraz o informačnej potrebe používateľa, pričom aktivita výrazne zasahuje do pochopenia tohto obrazu. Spoločne tak podobné dopyty a aktivita reprezentujú určitú epizódu vyhľadávania. Naša práca sa zakladá na predpoklade, že podobné dopyty väčšinou zdieľajú výsledky vyhľadávania a na základe miery prieniku týchto výsledkov vieme určiť ich vzájomnú podobnosť. Pomocou implicitnej spätnej väzby od používateľa vieme určiť, ktoré výsledky sú naozaj relevantné pre dopyt daného používateľa a na základe jeho aktivity tieto výsledky preferovať v prípade podobných dopytov. Cieľom nášho prístupu je prispôsobiť identifikáciu podobných dopytov správaniu používateľa a objaviť tak skryté súvislosti medzi dopytmi, ktoré sú relevantné práve preňho. V kapitole 2 sa venujeme analýze existujúcich prístupov ku identifikácii vyhľadávacích epizód. V kapitole 3 predstavujeme zhodnotenie existujúcich prístupov strojového učenia k identifikácii vyhľadávacích epizód. Kapitola 4 obsahuje návrh našej metódy využívajúcej aktivitu používateľa pre identifikáciu vyhľadávacích epizód, kapitola 5 reprezentuje realizáciu našej metódy, kapitola 6 opisuje overenie našej metódy, kapitola 7 obsahuje našu štúdiu a štatistické vyhodnotenie anotovania vyhľadávacích epizód pomocou externých anotátorov a kapitola 8 zhodnocuje našu prácu a predostiera plán na ďalší výskum.

2 Vyhľadávacie epizódy

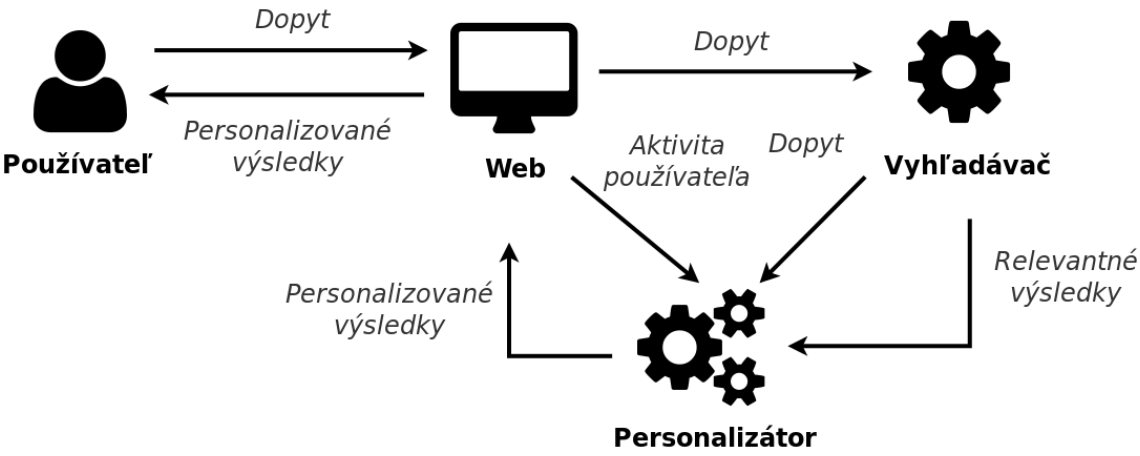
Efektivita získania informácii na webe v prípade bežných vyhľadávačov závisí od toho, ako dokážeme vyjadriť našu informačnú potrebu pomocou kľúčových slov. Jeden z faktorov, ktoré môžeme v prípade vyhľadávania ovplyvniť my, je náš dopyt. Pokiaľ chceme, aby vyhľadávač našiel čo najrelevantnejšie výsledky, musíme špecifikovať náš dopyt čo najpresnejšie, pretože bežný nepersonalizovaný vyhľadávač pre rovnaký dopyt rôznych používateľov vždy vráti rovnaké výsledky bez toho, aby bral do úvahy rôzne záujmy ľudí alebo rôzne kontexty pre získanie informácií.

Pokiaľ by vyhľadávač vedel o tom, ktoré dopyty spolu súvisia, aké záujmy a znalosti máme a aké webové stránky nás zaujímajú, mohol by na základe nášho dopytu a našich záujmov nájsť podobné dopyty a identifikovať náš cieľ. Potom za predpokladu, že dvaja používatelia majú odlišné záujmy, vyhľadávač by každému na základe znalosti ich cieľu a obsahu súvisiacich dopytov poskytol rôzne výsledky, prispôsobené presne pre nich [34]. Ukážka 1 prezentuje reálny dopyt, ktorý má výrazne odlišný význam pre dvoch rôznych používateľov.

Ukážka 1. *Dvaja rôzni používatelia zadajú dopyt* Java. *Význam tohto dopytu pre prvého používateľa môže byť programovací jazyk Java, pretože je daný používateľ programátor. Druhý používateľ môže byť cestovateľ, ktorý sa chce dozvedieť viac o ostrove* Java*, na ktorý sa chystá vycestovať. V oboch prípadoch by bežný vyhľadávač vrátil rovnaký zoznam výsledkov [42], ktorý by pozostával nielen z výsledkov o programovacom jazyku, ale aj ostrove a dokonca aj druhu kávy.*

Výrazne odlišný význam rovnakého dopytu pre dvoch rôznych ľudí nám napovedá, že vyhľadávacia epizóda predstavuje nielen zoznam dopytov, ale najdôležitejším kľúčom k jej významu je cieľ používateľa. Úlohou vyhľadávača je preto využitie dostupných znalostí o používateľovi a ostatných faktorov, ktoré ovplyvňujú vyhľadávanie, na nájdenie tohto významu. Prispôsobenie sa všetkým uvedeným faktorom ale prekračuje rámec bežných vyhľadávačov sústreďujúcich sa na dopyt ako zdroj informácii a je doménou tzv. personalizovaných vyhľadávačov.

Personalizovaný vyhľadávač odbúrava potrebu presnej špecifikácie nášho dopytu, pretože znalosti o našich záľubách a preferenciách čerpá z našej aktivity počas vyhľadávania a je tak schopný identifikovať náš cieľ bez našej explicitnej pomoci [34]. Na



Obr. 2.1: Architektúra personalizovaného vyhľadávača využívajúceho aktivitu používateľa pre určenie modelu používateľ

to, aby sme mohli identifikovať cieľ a personalizovať tak vyhľadávanie, potrebujeme určitý model, ktorým reprezentujeme uvedené záujmy používateľa [42]. Tento model môže pozostávať z viacerých atribútov, ktoré určujú [23]

- názory a ciele,
- znalosti – vzdelanie a zručnosti,
- históriu vyhľadávania – zadané dopyty, navštívené výsledky vyhľadávania a celkové správanie používateľa počas vyhľadávania.

Atribúty ako záľuby, vzdelanie a zručnosti neidentifikuje vyhľadávač priamo, nereprezentuje model používateľa ako presný profil jeho znalostí a záľub, nepomenúva jeho ciele priamo ako *kúpa auta*, ale reprezentuje ich práve pomocou aktivity používateľa. Dopyt sám o sebe ponúka len slabú informáciu o cieli používateľa, avšak ak je spojený s výsledkami vyhľadávania, ktoré používateľa zaujali, je možné rozšíriť informačnú hodnotu dopytu o atribúty získané z výsledkov vyhľadávania [42] a predstaviť význam daného dopytu pre používateľa. Informácie o kontexte môže vyhľadávač získať na základe aktivity používateľa počas vyhľadávania, pretože ako je možné vidieť na obrázku 2.1, používateľ neustále interaguje z výsledkami a ponúka implicitnú spätnú väzbu nielen kliknutím na výsledok, ale aj jeho vynechaním. Keď používateľ pre dopyt *Java* klikne na výsledok o programovacom jazyku, vyhľadávač zistí, že cieľom vyhľadávania pre programátora určite nie je ostrov.

2.1 Cieľ vyhľadávania ako zdroj personalizácie

Identifikácia cieľu dopytu pre personalizáciu vyhľadávania je náročná úloha, ktorú dokáže čiastočne zvládnuť vyhľadávač tak, že vie, ktoré dopyty zdieľajú približne rov-

naký cieľ na základe znalosti kontextu používateľa a jeho aktuálneho dopytu. Tým pádom dokáže rozdeliť dopyty s rovnakým alebo podobný cieľom do spoločných skupín – táto technika sa nazýva segmentácia dopytov do vyhľadávacích epizód (angl. *search session segmentation*) [21]. Definície vyhľadávacej epizódy sa líšia v závislosti od autorov a uváženia jednotlivých atribútov pre identifikáciu epizód. Viacerí autori [21, 40, 49, 44] pokladajú vyhľadávaciu epizódu za podmnožinu cieľu vyhľadávania, keďže cieľ vyhľadávania môže byť dlhodobejší a rozdelený do viacerých dní, pričom vyhľadávacia epizóda v ich prípade predstavuje časové okno, počas ktorého používateľ vyhľadával a môže tým pádom obsahovať dopyty s rôznymi cieľmi. Preto podľa Jones a ďalších [21], cieľ vyhľadávania môže zasahovať do viacerých vyhľadávacích epizód a predstavuje skupinu dopytov, ktoré zdieľajú spoločný cieľ.

Naším cieľom je zamerať sa na identifikáciu epizód ako cieľov vyhľadávania. Každá vyhľadávacia epizóda reprezentuje cieľ používateľa pri vyhľadávaní – tento cieľ môže byť kúpa auta alebo nájdenie najbližšieho kina, ktoré premieta náš obľúbený film. Preto sme definovali vyhľadávaciu epizódu podľa [21] nasledovne:

Definícia 1. *Vyhľadávacia epizóda (angl. search session) predstavuje všetku aktivitu používateľa s rovnakým alebo podobným cieľom v určitom časovom intervale.*

Na základe definície 1 uvažujeme vyhľadávaciu epizódu ako uzavreté časové okno, počas ktorého používateľ tvorí dopyty so spoločným cieľom. Časový interval, ktorý oddeľuje epizódy navzájom môže byť niekoľko minút alebo aj niekoľko dní. Väčšina existujúcich prístupov [21, 50, 32] overila, že najlepším ohraničením je časový interval medzi 5 minútami až 1 hodinou. Lucchese a ďalší [32] identifikovali, že nezáleží na dĺžke intervalu v rámci 1 hodiny, pretože signifikancia každého intervalu je relatívne rovnaká. Keďže sa berie do úvahy maximálne len jedna hodina ako uzatvárajúci interval, vyhľadávacia epizóda potom zachytí buď len krátkodobý záujem používateľa alebo časť z dlhodobého cieľu vyhľadávania.

Pokiaľ chceme uvažovať ciele, ktoré prekračujú dĺžku hodiny a sú rozdelené do viacerých dní, ako napríklad kúpa auta, musíme zaviesť spôsob ako združovať podobné vyhľadávacie epizódy do skupín podľa podobnosti. Preto môžeme definovať cieľ vyhľadávania ako združovateľ podobných epizód pre reprezentáciu dlhodobejších cieľov vyhľadávania.

Definícia 2. *Cieľ pri vyhľadávaní (angl. search goal) je atomická informačná potreba reprezentovaná jednou alebo viacerými súvisiacimi vyhľadávacími epizódami [21].*

2.2 Identifikácia cieľu vyhľadávania

Automatická identifikácia cieľu vyhľadávania závisí od viacerých atribútov súvisiacich s kontextom používateľa a aj samotného dopytu. Záznamy o aktivite používateľa

počas vyhľadávania – dopyty, navštívené výsledky, čas strávený vyhľadávaním – nám dokážu ponúknuť komplexný obraz o tom ako používatelia vyhľadávajú na webe a pomôžu nám odhadnúť, aký je reálny motív za ich dopytmi. Na základe toho predchádzajúce práce [45, 32] rozdelili metódy pre identifikáciu cieľa počas vyhľadávania nasledovne:

Založené na čase

Čas bol prevzatý ako atribút kontextu dopytu hlavne z dôvodu jeho jednoduchosti. Radlinski a ďalší [40] ukázali, že používatelia mnohokrát tvoria dopyty s podobnou informačnou potrebou v krátkom časovom intervale a tieto dopyty identifikovali ako reformulácie. Reformulované dopyty nazvali ako sledy dopytov (angl. *query chains*), pričom ich reprezentovali ako záznamy o vyhľadávaní v rámci 30-minútových časových okien. Čas však nemusí byť najlepším oddeľovačom podobnosti dopytov, pretože ako ukázali aj predchádzajúce prístupy [21, 50, 32], ciele používateľov sa mnohokrát prelínajú a počas určitého času tvoria dopyty, ktoré sa vzťahujú na rôzne ciele.

Založené na obsahu

Niektoré práce [33, 36, 21] využili lexikálne vlastnosti dopytov pre vzájomné porovnanie. Vďaka komplexným algoritmom pre porovnávanie reťazcov je lexikálna analýza pomerne úspešná pri identifikácii podobných dopytov, pričom najužitočnejšia je v prípade reformulácie dopytov. Lexikálna analýza ale zlyháva v prípade tzv. problému významového nesúladu (angl. *vocabulary mismatch problem*), kedy majú dva rovnaké slová rôzny význam [32]. Riešením je rozšírenie samotného dopytu o ďalšie slová, ktoré upresňujú význam pre konkrétneho používateľa. Aj napriek rozšíreniu významu dopytu, stále pretrvávajúcim problémov v prípade lexikálnej analýzy sú krátke dopyty. V prípade vyhľadávačov je štatisticky známe, že dopyty neobsahujú viac ako tri slová [45].

Založené na aktivite používateľa

Pomocou výsledkov vyhľadávania môžeme reprezentovať aktivitu používateľa počas vyhľadávania. Pokiaľ používateľ klikne na výsledok vyhľadávania, je to pre nás informácia o tom, že tento výsledok je relevantný pre jeho dopyt. Okrem binárnej informácie o zaujímavosti výsledku vieme určiť, ako dlho daný používateľ strávi prezeraním danej stránky a zaradiť stránku medzi kategórie na základe analýzy jej obsahu. Viacerí autori identifikovali [8, 3, 27, 15, 40], že implicitná spätná väzba od používateľa pri výsledkoch vyhľadávania výrazne zlepší určenie cieľa a významu dopytu pre konkrétneho používateľa, pretože webová stránka poskytuje omnoho viac informácii ako dopyt skladajúci sa len s niekoľkých slov.

Keď si používateľ prezerá výsledky vyhľadávania, vyberá si výsledky, ktorého ho zaujímajú a sú z jeho pohľadu relevantné pre dosiahnutie jeho cieľa. Svojou aktivitou odovzdáva vyhľadávaču implicitnú spätnú väzbu o jeho preferenciách v prípade daných výsledkov vyhľadávania. Následne vyhľadávač môže pre daný dopyt zistiť z obsahu stránok, aké ďalšie kľúčové slová sú relevantné pre aktuálnych dopyt používateľa [42] a rozšíriť tak automaticky dopyt bez potreby explicitného vyjadrenia daných slov používateľom. Toto je jedna z možností vyhľadávača ako personalizovať vyhľadávanie a implicitne určiť cieľ používateľa.

Výsledky vyhľadávania okrem obsahu ponúkajú aj ďalšie atribúty – samotné kliknutie na výsledok. Ak vezmeme do úvahy výsledky dopytov idúce za sebou a niektoré výsledky pre jednotlivé dopyty sú zdielané, môžeme čiastočne predpokladať, že dva dopyty majú podobný cieľ. Pokiaľ ale používateľ klikol na niektorý z týchto výsledkov, môžeme predpokladať, že daný výsledok obsahuje relevantné informácie pre aktuálny dopyt používateľa, čiže z daného výsledku vieme čiastočne identifikovať význam dopytu pre používateľa. Kliknutie ale nemusí byť úplne najlepším indikátorom relevancie výsledku pre dopyt používateľa, pretože častokrát sa používatelia len navigujú vo výsledkoch a snažia sa nájsť práve ten, ktorý je naozaj relevantný. Na niektorých výsledkoch tak strávia len pár sekúnd a opäť sa vrátia späť na vyhľadávanie. Preto, presne ako časové okná, kliknutie na výsledok vie výrazne zvýšiť zlepšiť identifikáciu vyhľadávacích epizód v spojení s iným atribútmi vyhľadávania, medzi ktoré môžeme zaradiť

- čas strávený prezeraním výsledku,
- zoznam kľúčových slov výsledku,
- pozíciu výsledku v zozname a
- počet výskytov daného výsledku.

Uvažujúc pozitívnu spätnú väzbu formou kliknutia, používateľ svojou aktivitou ešte implicitne udáva, ktoré výsledky preňho nie sú zaujímavé. Pokiaľ používateľ pri prehliadaní výsledkov klikne na piaty výsledok z desiatich, môžeme predpokladať, že prvé štyri výsledky nie sú pre neho relevantné, aj napriek ich vysokej relevancii určenej vyhľadávačom. Preto môžeme znevýhodniť ich relevanciu pre daného používateľa a ich vysoká globálna relevancia nebude zasahovať do konštrukcie záujmov a preferencii používateľa.

2.2.1 Podobnosť dopytov

Na podobnosť dopytov vplýva viacero atribútov, ktoré sa viažu nielen na kontext daného dopytu, ale aj na samotnú vyhľadávaciu epizódu. Zaraďujeme medzi ne

- časový odstup medzi dvoma dopytmi,
- lexikálna podobnosť s predchádzajúcimi dopytmi a
- počet spoločných slov s prechádzajúcimi dopytmi.

Časový odstup dopytov

Časový interval medzi dopytmi dokáže identifikovať zmeny v krátkodobých cieľoch používateľov. Preto, ak rozdiel medzi dopytmi používateľa je viac ako 1 hodina, predpokladáme, že dané dopyty spolu nesúvisia v prípade krátkodobých cieľov [50]. Na základe nášho predpokladu, že cieľ vyhľadávania môže byť rozdelený do viacerých vyhľadávacích epizód, tak časové atribúty dopytov dokážu identifikovať s dobrou presnosťou len časť cieľov vyhľadávania. Ako naznačili aj ďalšie práce [21, 16, 50], časové atribúty preto nie sú dobrými atribútmi pre identifikáciu vyhľadávacích epizód, pokiaľ sú použité samostatne, ale hodnotne rozširujú kvalitu ostatných prístupov berúcich do úvahy súlad viacerých atribútov.

Lexikálna podobnosť dopytov

Porovnanie dopytov na základe lexikálnych vlastností umožňuje vyhodnotiť napríklad reformulácie dopytov a ich rozširovanie alebo konkretizáciu [51]. Používatelia v mnohých prípadoch tvoria dopyty, ktoré úzko súvisia s predchádzajúcimi, pretože každým novým dopytom sa snažia viac špecifikovať svoj cieľ vyhľadávania. Pridávajú tak do dopytov viaceré nové slová, pričom základný tvar a obsah dopytu nemenia. Na základe toho môžu algoritmy pre lexikálne porovnávanie textu využiť zdielané slová na určenie reformulácie dopytov a ich následne zaradenie do rovnakej vyhľadávacej epizódy.

Na výpočet miery podobnosti medzi dopytmi pomocou uvedených atribútov sa využívajú rôzne algoritmy. Viacerí autori [33, 36, 21] využili algoritmus Levenstheinovej vzdialenosti medzi dvoma reťazcami. Algoritmus Levenshteinovej vzdialenosti určuje, koľko je potrebných zmien v prvom reťazci na to, aby sme ho pretransformovali na druhý reťazec [26]. Jones a ďalší [21] identifikovali, že Levenshteinova vzdialenosť je štatisticky najpresnejší lexikálny atribút pre určenie podobnosti medzi dopytmi. Toto tvrdenie obhájili aj tým, že uvedená presnosť sa výrazne nezmenila, ani keď autori vo svojom testovacom datasete eliminovali duplicitné dopyty. Ich výsledky sa zakladali na normalizovanej forme Levenshteinovej vzdialenosti na intervale $< 0, 1 >$ definovanej ako

$$NLD(A, B) = \frac{LD(A, B)}{max(LD(A_n, B_n))} \tag{2.1}$$

kde A a B predstavujú vstupné reťazce, $max(LD(A_n, B_n))$ predstavuje maximálnu hodnotu vzdialenosti nad všetkými pre nás dostupnými reťazcami a funkcia LD určuje Levenshteinovu vzdialenosť medzi reťazcami A a B.

Ako sme už spomínali, používatelia častokrát iba pridávajú nové slová do dopytov, aby konkretizovali svoj cieľ. Na identifikáciu podobnosti medzi dopytmi, kedy sa pridávajú len nové slová, môžeme použiť aj jednoduchší prístup porovnania ako je miera spoločných slov [36, 49]. Autori okrem prieniku podobných slov využili aj ďalšie algoritmy ako kosínusová podobnosť alebo Jaccardova vzdialenosť. Kosínusová podobnosť definuje podobnosť medzi dvoma vektormi pomocou priestoru zvierajúceho kosínusový uhol medzi nimi a je definovaná ako

$$similarity(A, B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.2}$$

kde A a B sú vektory slov jednotlivých dopytov.

V ukážke 2 môžeme vidieť reálny príklad hodnoty kosínusovej podobnosti pre dopyty *Bentley autobazár* a *Bentley autobazár Bratislava*.

Ukážka 2. *Oba dopyty – Bentley autobazár a Bentley autobazár Bratislava – rozložíme na vektory slov podla toho, ktoré slová z celkovej množiny možných slov obsahujú. Celková množina slov prestavuje zjednotenie všetkých slov z oboch dopytov – Bentley, autobazár, Bratislava. Preto, vektor pre prvý dopyt Bentley autobazár je* $[1, 1, 0]$ *a druhý dopyt Bentley autobazár Bratislava má vektor rovný* $[1, 1, 1]$*. Na základe vzťahu 2.2 je kosínusová podobnosť uvedených dopytov 0.816.*

Ako je možné vidieť, podobnosť dopytov je výrazne vysoká a udáva správnu náväznosť dopytov, pretože sa jedná o reformuláciu dopytu a upresnenie konkrétneho cieľu – nájdenie autobazárov predávajúcich značku Bentley v Bratislave. Výkonnosť algoritmov na porovnávanie reťazcov sa dá vylepšiť pomocou predspracovania slov v dopytoch na ich korene (angl. *stemming*) [35]. Najpoužívanejším algoritmom je Porterov algoritmus pre korene slov (angl. *Porter stemming algorithm*). Na základe toho by dopyty *Running in California* a *Places to Run in California* boli pretransformované na *Run in California* a *Places to Run in California* a ich výsledná spoločná podobnosť by bola výrazne vyššia. Pokiaľ uvažujeme dopyty v slovenčine alebo češtine, problém rozloženia slov na ich korene je komplikovaný výskumný problém a je výrazne komplikovanejší ako rozklad v prípade angličtiny kvôli charakteru jednotlivých jazykov.

Lexikálna podobnosť dopytom má ale aj svoje nedostatky, pretože pokiaľ by sme ale aplikovali kosínusovú podobnosť alebo Levenshteinovu vzdialenosť na dopyt *Cucumber recipes books* a *Cucumber books*, zistíme, že sa výrazne podobajú. Avšak, v prípade prvého dopytu sa používateľ mohol zaujímať o knihy s receptami a v druhom

dopyte sa používateľ zaujímal o knihy pre testovací rámec *Cucumber*. Lexikálne atribúty by ale identifikovali, že dané dva dopyty sú výrazne podobné, avšak cieľ, ktorý sa za nimi skrýva, je v skutočnosti výrazne odlišný.

Na základe uvedeného problému je lexikálna analýza dopytov vhodná len pre identifikáciu reformulácie dopytu a čiastočne krátkodobých cieľov počas vyhľadávania, pretože nedokáže zachytiť sémantické vzťahy nielen medzi jednotlivými dopytmi, ale ani vzťahom dopytu k záujmom používateľa. Preto sa lexikálna analýza a časové atribúty využívajú hlavne v spolupráci s inými, komplexnejšími atribútmi kontextu používateľa [21, 50].

Ďalšie atribúty, ktoré vedia prispieť k pochopeniu sémantických vzťahov medzi dopytmi sú výsledky vyhľadávania spojené s aktivitou používateľa. Pokiaľ uvažujeme, že dopyt je viacznačný, tak vyhľadávač poskytne používateľovi zoznam výsledkov, ktoré sa viažu na každú oblasť významu daného dopytu. Na to, aby vyhľadávač zistil, ktorý význam je relevantný pre daného používateľa, môže sa pozrieť, či niektorí z výsledkov používateľ nevidel pri predchádzajúcich dopytoch. Nájdenie dopytov, ktoré daný používateľ zadal a zdieľali s aktuálnym dopytom niektoré výsledky, môže viesť k odhaleniu nadväzujúcich dopytov. Avšak, opäť musíme brať do úvahy, že aj prechádzajúce dopyty mohli byť viacznačné a zdieľané výsledky sú len odpoveďou vyhľadávača na rozsiahly informačný priestor. Na to, aby bolo možné odfiltrovať viacznačnosť dopytov a nájsť relevantný zdroj významu dopytu pre používateľa, musíme uvažovať aktivitu používateľa počas vyhľadávania a jeho interakciu s výsledkami.

Používateľ aktivitou určuje svoje preferencie pri jednotlivých výsledkoch a prezrádza, ktorý význam dopytu je preňho relevantný. Pokiaľ používateľ klikne na výsledok vyhľadávania a strávi na ňom určitý čas, ktorý môžeme uvažovať ako reprezentatívny čas záujmu používateľa o obsah výsledku, tak obsah daného výsledku môže výrazne prispieť k opisu významu dopytu pre používateľa. Následne, ak sa tieto relevantné výsledky zobrazia pre ďalšie dopyty, môžeme uvažovať, že tieto dopytu spolu súvisia.

2.2.2 Rozšírenie významu dopytov

Pre rozšírenie významu dopytov je nutné rozšíriť ich informáciu z iných zdrojov (angl. *query expansion*). Táto technika sa používa na obohatenie reprezentácie krátkych dopytov o nové slová, ktoré sú relevantné pre dopyt a pravdepodobne sa vyskytnú aj vo výsledných dokumentoch [35], pretože podľa štatistík [45], dopyty používateľov sú zvyčajne krátke a neobsahujú viac ako tri slová.

Viacerí autori [42, 10] sa zamerali na rozšírenie významu dopytov o informácie z výsledkov vyhľadávania. Shen a ďalší [42] rozšírili dopyt o nadpisy z prvých 50

výsledkov vyhľadávania. Na extrakciu relevantných slov použili metódu TF-IDF a následne porovnali vektory slov pre dva dopyty na základe kosínusovej podobnosti. Pokiaľ podobnosť prekročila určitú hranicu, dané dva dopyty autori zaradili do rovnakej vyhľadávacej epizódy.

Cui a ďalší [10] identifikovali, že využitie všetkých výsledkov pre rozšírenie dopytu môže výrazne zasiahnuť do relevancie podobnosti dopytov, pretože mnohé výsledky vyhľadávania nie sú relevantné, ale obsahujú len podobný text, ktorý významovo nemusí byť v súlade s dopytom. Preto brali pri rozšírení dopytov do úvahy len výsledky, na ktoré používatelia klikli. Dôvod pre uváženie len kliknutých výsledkov je jednoduchý – výsledky vybraté používateľmi sú najrelevantnejšie pre ich dopyt [45]. Na základe toho autori extrahovali pre každý výsledok zoznam kľúčových slov a rozšírili dopyt. Autori prezentovali bipartitný graf na obrázku 2.3, kde vzťah medzi termom z dokumentu D – t_D a termom z dopytu Q – t_Q existoval len v tom prípade, ak pre aspoň jeden dopyt obsahujúci t_Q existoval jeden kliknutý dokument t_D. Váha jednotlivých hrán je určená na základe pravdepodobnosti výskytu termu t_D pre term t_Q nasledovne

$$P(t_D, t_Q) = \sum_{\forall D_i \in S} P(t_D, D_i) \frac{freq(t_Q, D_i)}{freq(t_Q)} \tag{2.3}$$

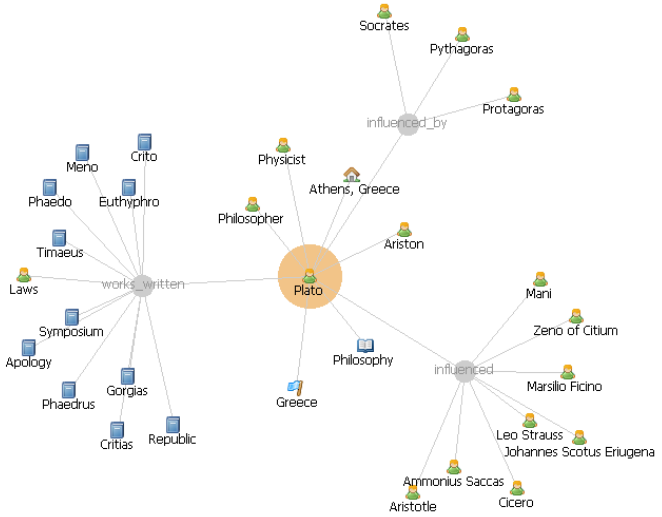
kde S je zoznam kliknutých výsledkov pre dopyty obsahujúce term t_Q, $P(t_D, D_i)$ je normalizovaná relevancia termu t_D pre dokument D_i ku všetkým váham termov pre daný dokument, $freq(t_Q, D_i)$ je počet dopytov, kde sa term t_Q a dokument D_i nachádzajú spolu a $freq(t_Q)$ je počet dopytov, ktoré obsahujú term t_Q.

Ďalší autori sa namiesto výsledkov vyhľadávania obrátili na otvorené encyklopédie. *Wikipedia*[1], *Freebase*[2] a *Probase*[3] sú otvorené encyklopédie, ktoré ponúkajú rôzne metadáta pre určenie konceptov a sú tak schopné rozšíriť význam dopytu.

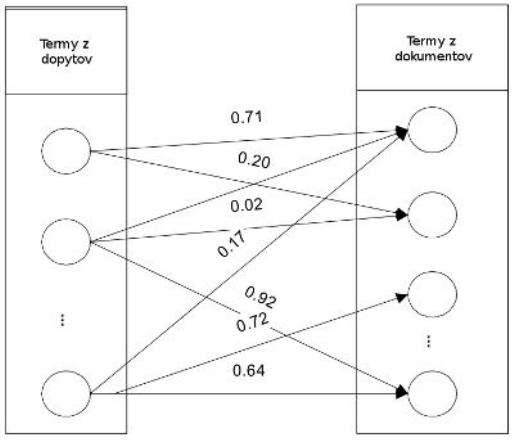
Existuje viacero prístupov reprezentácie konceptov a znalostí, ktoré rozdeľujeme na [33]

- Založené na vzťahoch, kde sú znalosti modelované pomocou grafu konceptov a význam vzťahov je určený na základe ciest v grafe, ako je možné vidieť na obrázku 2.2,
- Založené na informačnom obsahu, ktorý reprezentuje koncept,
- Založené na vektoroch, kde každý koncept je modelovaný ako vektor slov.

[1]Wikipedia: https://www.wikipedia.org/
[2]Freebase: https://www.freebase.com/
[3]Probase: http://research.microsoft.com/en-us/projects/probase/



Obr. 2.2: Graf reprezentujúci vzťahy entít v rámci encyklopédie Freebase (zdroj: http://thinkbase.cs.auckland.ac.nz/)



Obr. 2.3: Graf vzťahu termov z dopytu a termov extrahovaných z dokumentov [10]

Lucchese a ďalší [33] použili metódu wikifikácie, kde rozšírili význam dopytu o koncepty pomocou článkov z Wikipedia a Wiktionary[4]. Autori prestavili wikifikáciu termu t z dopytu $\overrightarrow{C}(t) = (c_1, c_2, c_3, \ldots, c_W)$ ako vektor relevancií termu voči článkom vo Wikipedia a Wiktionary. Relevanciu termu t autori určili na základe miery TF-IDF. Výsledná wikifikácia dopytu potom predstavuje sumu všetkých vektorov termov, z ktorých dopyt pozostáva.

$$\overrightarrow{C}(Q) = \sum_{\forall t \in Q} \overrightarrow{C}(t) \tag{2.4}$$

Porovnanie na základe ich významu je potom možné pomocou kosínusovej podobnosti $\overrightarrow{C}(Q)$ vektorov ich dopytov.

Hua a ďalší [17] identifikovali, že rozšírenie dopytov len na úrovni jednotlivých termov môže spôsobiť nekonzistenciu vo význame v prípade konceptov s viacslovným pomenovaním. Príkladom môže byť dopyt *tiger woods*. Wikifikácia podľa prechádzajúceho prístupu potom rozšíri význam dopytov o články o tigroch a dreve, pretože dopyt rozdelí na dva termy – *tiger* a *woods*. Koncepty asociované s dopytom sú potom zviera a drevo, pričom cieľom vyhľadávania bol golfový hráč Tiger Woods.

Autori sa pokúsili odstrániť tento problém pomocou encyklopédie *Probase*. V ich prípade Probase mapoval najdlhší možný názov konceptu a správne identifikoval koncept dopytu *tiger woods* ako golfový hráč. Autori porovnali aj ďalšie dopyty a zistili, že encyklopédie ako Probase dokážu veľmi dobre rozlíšiť aj viacznačnosť dopytov.

2.2.3 Aktivita používateľa

Používateľ sa počas vyhľadávania naviguje medzi výsledky vyhľadávania a prezerá si obsah konkrétnych výsledkov. Všetky akcie, ktoré používateľ vykoná počas vyhľadávania, nám dokážu implicitne povedať niečo o tom, aké má daný používateľ záujmy, znalosti a prezradia nám, aký význam a vzťah má daný používateľ k svojmu dopytu. Napríklad, pre viacznačný dopyt, ktorému vyhovujú výsledky z viacerých tématických oblastí, nám používateľ kliknutím na výsledok vyhľadávania naznačuje, že výsledok z danej témy je z jeho pohľadu najrelevantnejší pre jeho aktuálny dopyt.

Správanie používateľov počas vyhľadávania je komplikované, pretože pokiaľ neuvažujeme len dopyt, ale aj ostatné akcie používateľa pri vyhľadávaní, existuje veľké množstvo kombinácii akcií používateľa, ktoré zahŕňajú kliknutie na výsledok, prečítanie krátkeho úryvku pri výsledku (angl. *snippet*), čas strávený prezeraním konkrétneho výsledku a mnoho ďalších [29]. Správanie používateľa teda môže predstavovať kombináciu týchto akcií a za rôznymi kombináciami sa môžu skrývať rôzne

[4] http://www.wiktionary.org/

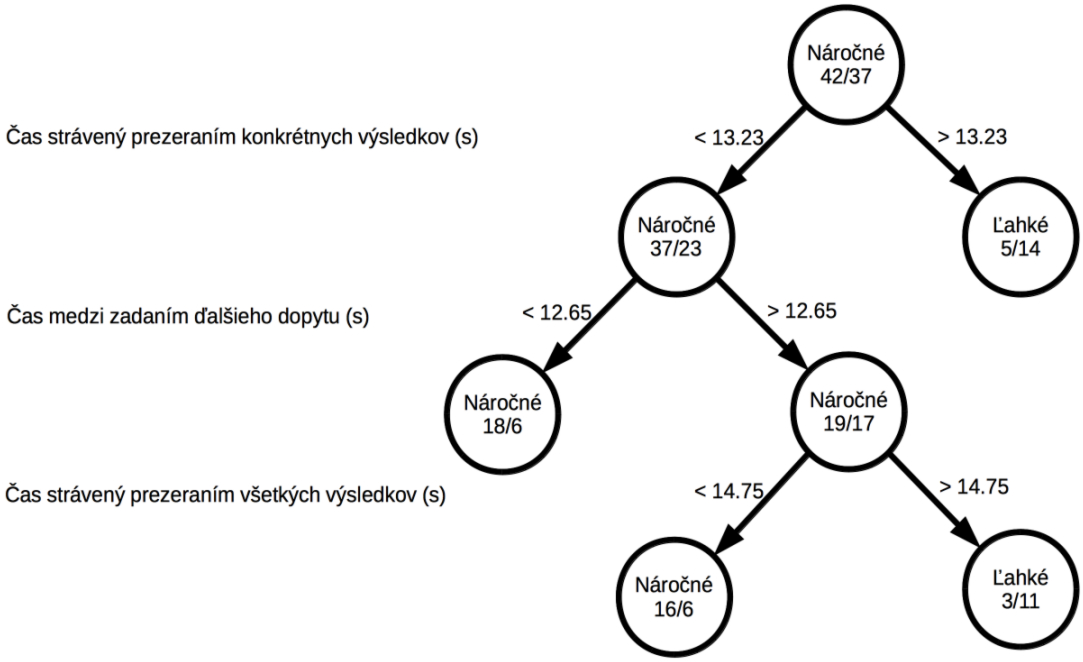
zámery používateľa pre dosiahnutie cieľu vyhľadávania. Ku komplexnosti aktivity používateľa počas vyhľadávania pridáva aj fakt, že hlavne v prípade exploratívneho vyhľadávania, kedy používateľ prehľadáva informačný priestor s cieľom oboznámiť sa s určitou témou, sa správanie používateľa líši na začiatku a aj na konci vyhľadávania. Preto môžeme rozdeliť fázy vyhľadávania na tri časti [30]

- začiatok vyhľadávania – pred zadaním ďalšieho dopytu,
- stred procesu vyhľadávania, kedy používateľ získava prehľad v danej informačnej oblasti,
- koniec vyhľadávania, kedy používateľ dosiahol cieľ vyhľadávania.

2.2.3.1 Scenáre správania používateľa

Na základe tohto rozdelenia vieme zaradiť pozorované správanie používateľa do scenárov vyhľadávania. Na začiatku používateľ prehľadáva viaceré výsledky, pretože jeho cieľom je získať širší obraz o danej téme. Začiatok vyhľadávania sa preto prevažne vyznačuje viacerými navštívenými stránkami a dlhším časom stráveným prezeraním stránok (angl. *dwell time*). Tématické zaradenie jednotlivých stránok sa prelína a používateľ sa venuje viacerým oblastiam, hlavne v prípade viacznačných dopytov, s cieľom získať prehľad v určitej oblasti. Hlavne v tejto fáze môžeme získať najviac metadát o význame dopytu pre používateľa. V prípade viacznačného dopytu používateľ nechce získať všeobecný prehľad o všetkých významoch dopytu, ale sústreďuje sa prevažne na konkrétnu tému. Navštevovaním viacerých stránok a časom na nich stráveným nám udáva, ktorá stránka sa významovo zaraďuje do jeho témy záujmu. Ak teda v správaní používateľa pozorujeme, že na niektorých stránkach strávil veľké množstvo času a iné stránky len otvoril a v zápätí ich zavrel, môžeme tieto nerelevantné stránky na základe krátkeho času úplne odfiltrovať a znevýhodniť pri analýze aktivity používateľa. Ako postupujú používatelia ďalej pri vyhľadávaní, ich cieľ sa začína viac špecifikovať a ich správanie sa mení – sústreďujú sa už na konkrétne výsledky s úzko-spojenou témou a prezerajú si bližšie väčšinou menšie množstvo výsledkov, pretože už získali určitý prehľad v danej oblasti a vedia odhadnúť, ktoré výsledky sú naozaj relevantné k ich téme. Ich cieľ už nie je získanie prehľadu, ale rozšírenie získaných informácii [30, 31].

Ak uvažujeme, že cieľ používateľov je nájdenie konkrétnej informácie, môžeme predpokladať, že ich dopyty budú výrazne lexikálne podobné, pretože častokrát používatelia hľadajú danú informáciu v zhone a reformuláciou sa snažia trafiť ten správny dopyt, ktorý odfiltruje výsledky obsahujúce hľadanú informáciu. Z ohľadom na aktivitu môžeme predpokladať, že používatelia budú voliť viacero výsledkov, na ktorých strávia menej času hľadajúc danú informáciu. Výsledné relevantné výsledky



Obr. 2.4: Rozhodovací strom atribútov pre určenie náročnosti cieľu vyhľadávania pri prvom dopyte [30]. Čísla pod identifikovanou úrovňou náročnosti značia koľko úloh bolo v skupine je reálne náročných/jednoduchých

môžu byť v prípade ich informačnej potreby len niektoré výsledky, na ktorých strávili priemerne dlhší čas než na ostatných počas ich vyhľadávania alebo na nich označovali a kopírovali text [18]. Ďalší autori [30] však identifikovali, že vzory správania pri častej a rýchlej reformulácii dopytov môžu indikovať, že používateľ je zmätený z náročnosti svojho informačného cieľa a prvotné výsledky neponúkajú žiadne relevantné informácie, preto sa snaží reformulovať svoj dopyt. Autori identifikovali na základe ich atribútov aj opis správania používateľa v prvej fáze vyhľadávania a zostrojili rozhodovací strom na obrázku 2.4, ktorý opisuje nakoľko jednotlivé hodnoty atribútov zatrieďujú náročnosť cieľa. Na základe toho je potrebné brať do úvahy, že v prípade správania používateľa existuje viacero možných scenárov a akcie používateľa sú mnohokrát výrazne viacznačné a líšia sa od konkrétneho používateľa. Ako je teda možné vziať do úvahy správanie používateľov, keď stroj v mnohých prípadoch nevie bojovať z viacznačnosťou ich akcií?

Viaceré prístupy [8, 3, 27, 15, 9, 18, 30, 31] sa zamerali na využitie interakcie

používateľa s výsledkami vyhľadávania pre identifikáciu vyhľadávacích epizód. Autori využili implicitnú spätnú väzby od používateľa vo forme kliknutí na výsledky a vyhľadávacie epizódy identifikovali na základe výsledkov, ktoré dopyty zdieľajú. Kliknutie v kombinácii s časom prezerania výsledku nám vie prezradiť nakoľko je výsledok pre používateľa relevantný, pretože ak používateľ klikne na výsledok a za jednu alebo dve sekundy sa vráti späť na zoznam výsledkov, môžeme predpokladať, že výsledok nebol veľmi nápomocný v jeho informačnej potrebe. Implicitná spätná väzba od používateľa sa ale viaže nielen na kliknuté výsledky, ale aj výsledky, ktoré používateľ nevybral. Pokiaľ používateľ zo zoznamu výsledkov klikne na tretí výsledok, môžeme predpokladať, že prvé dva výsledky, ktoré vyhľadávač identifikoval ako relevantné, nemusia byť významovo v súlade s dopytom pre daného používateľa alebo sa venujú inej téme v prípade viacznačného dopytu. Tieto výsledky môžeme následne pri analýze správanie znevýhodniť alebo vôbec neuvažovať pri výpočte podobností dopytov.

2.2.3.2 Atribúty aktivity používateľa

Výskumy naznačujú, že medzi najdôležitejšie atribúty ovplyvňujúce skutočnosť, že používateľ klikne na výsledok, sú práve pozícia výsledku a kontext – ostatné dokumenty [15]. Výsledky, ktoré sú relevantné pre používateľa, môžu byť prebité ostatnými výsledkami, ktoré sa na základe vysokej textovej podobnosti nachádzajú na prvých miestach. Používateľ si ich nemusí všimnúť, pretože sa k nim jednoducho neposunie a pokračuje namiesto toho v špecifikovaní dopytu [39, 41]. Generická relevancia výsledkov výrazne ovplyvňuje správanie používateľa pri prehliadaní výsledkov, avšak uvážením tohto problému a zvýšením relevancie pre výsledky na nižších pozíciach, ak na ne používateľ klikol, je čiastočným riešením, ktoré funguje len v prípade, ak používateľ tieto výsledky naozaj uvidí a nezvolí radšej cestu reformulácie dopytu.

Preto je potrebné pre zlepšenie presnosti identifikácie vyhľadávacích epizód využiť súhru viacerých atribútov. Ak upravíme relevanciu výsledkov na základe aktivity používateľa a ďalších atribútov, vieme lepšie odhadnúť podobnosť dopytov na základe zdieľaných výsledkov, pretože hlavne v prípade viacznačných dopytov ako *Cucumber*, môžeme naraziť na výsledky z rôznych oblastí, pričom len jedna oblasť je relevantná pre vyhľadávaciu epizódu používateľa. Pre upravenie relevancie výsledkov vyhľadávania viacerí autori [8, 3] využili atribúty, ktoré zohľadňujú

- pozíciu v zozname výsledkov,
- frekvenciu predchádzajúcich kliknutí na výsledok,
- počet výsledkov, ktoré boli kliknuté pred aktuálnym,

- počet výsledkov, ktoré boli kliknuté po aktuálnom,
- čas strávený prezeraním výsledkov a
- čas strávený prezeraním konkrétneho výsledku.

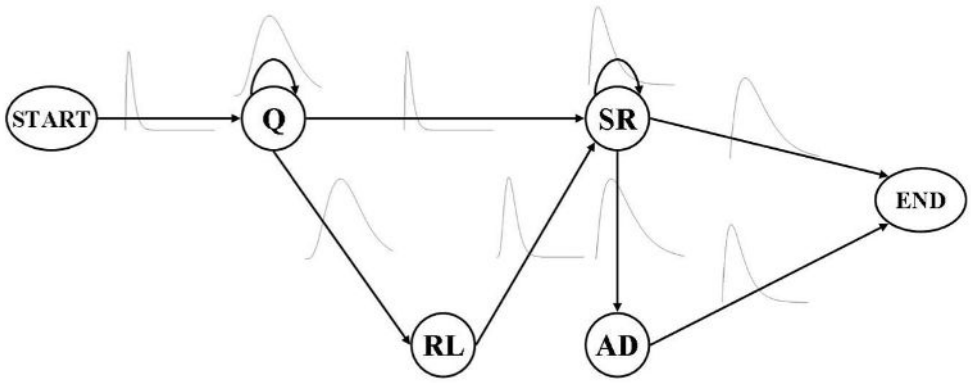
Na základe experimentov autori [8] identifikovali frekvenciu kliknutí na výsledok a jeho pozíciu ako najvýznamnejšie atribúty. Avšak, sami autori uznávajú, že dané atribúty, hlavne v prípade frekvencie kliknutí, sú významné len v prípade použitia s ďalšou sadou atribútov, pretože v opačnom prípade zavádzajú príliš veľa šumu. Používateľ môže kliknúť na výsledok, avšak hneď na prvý pohľad zistí, že sa netýka toho, čo hľadá a vráti sa späť na výsledky vyhľadávania, čiže času strávený prezeraním nám dokáže viac priblížiť relevantnosť výsledku pre používateľa. Práve pri pozorovaní správania používateľov počas vyhľadávaní ďalší autori identifikovali čas strávený prezeraním konkrétnych výsledkov a počet kliknutých výsledkov ako najrelevantnejšie atribúty [30]. Tieto atribúty boli najrelevantnejšie počas všetkých troch fáz vyhľadávania, t.j. na začiatku, v strede aj na konci.

Pokiaľ používateľ klikol na výsledok a čas prezerania tohto výsledku prekročil určitú definovanú hranicu, môžeme povedať, že daný výsledok je relevantný pre používateľa v prípade aktuálneho dopytu. Uvedená formulácia pripomína pravidlá, na základe ktorých je možné určovať, či je výsledok relevantný alebo nie, t.j. defacto zatrieďovať výsledky do dvoch skupín. Na základe toho môžeme uvedený problém relevancie výsledku považovať za problém strojového učenia a na reprezentáciu uvedených pravidiel môžeme použiť rozhodovacie stromy. Podobný prístup aplikovali pre určovanie náročnosti cieľu vyhľadávania autori v práci [30] a pomocou algoritmu rekurzívneho delenia identifikovali najrelevantnejšie atribúty ich modelu a definovali, že v ich prípade model identifikuje cieľ ako náročný, ak používateľ strávi prezeraním výsledkov pre jeden dopyt viac ako 36 sekúnd, priemerný čas prezerania konkrétnych výsledkov je viac ako 9,9 sekúnd a používateľ klikol aspoň na jeden výsledok. Rovnaký postup je možné aplikovať na relevanciu výsledkov. Ak určíme, že výsledok je relevantný pre používateľa na základe uvedených pravidiel, môžeme povedať, že nový dopyt, ktorý obsahuje tento výsledok, je podobný s daným dopytom.

Aj ďalší autori identifikovali, že implicitná spätná väzba od používateľov dokáže s výrazne lepšou presnosťou identifikovať podobnosť medzi dopytmi ako lexikálne prístupy [15, 8, 16]. Preto vytvorili model len na základe atribútov zahrňujúcich aktivitu používateľa, neuvažujúc žiaden čas strávený prezeraním výsledkov ani aktivitu na zvolených výsledkoch. Využili čisto len vzory v správaní používateľa pri kliknutiach na rôzne typy výsledkov vyhľadávania za rôznych podmienok a svoj model založili na atribútoch

- kliknutie na výsledok vyhľadávania,

- kliknutie na sponzorovaný výsledok,
- kliknutie na odporučený výsledok
- kliknutie na pravopisná opravu dopytu (angl. *spelling suggestion*),



Obr. 2.5: Príklad Skrytého Markovoveho modelu pre správanie používateľa počas vyhľadávania [16]

Na základe týchto atribútov vytvorili graf, ktorý pozostával z uzlov reprezentujúcich akciu používateľa a hrana predstavovala presun medzi jednotlivými akciami. Ďalší autori využili komplexnejšiu reprezentáciu cieľu vyhľadávania. Cieľ vyhľadávania reprezentovali ako sekvenciu za sebou idúcich akcií, čo im pomohlo vytvoriť model založený na pravdepodobnostiach prechodu jednotlivými akciami a využili pri tom Skryté Markovove modely [16]. Ako je možné vidieť na obrázku č. 2.5, akcie používateľa sú reprezentované ako prechody medzi jednotlivými stavmi. Ako aj v prípade prechádzajúcich prístupov, aj v tomto prístupe autori uvažovali čas ako jeden z najdôležitejších atribútov. Čas zakomponovali do modelu ako čas prechodu medzi jednotlivými stavmi. Pokiaľ stav prechodu medzi kliknutím na výsledok a zadaním nového dopytu je dlhší ako pár sekúnd, vieme povedať, že používateľa tento výsledok asi zaujímal a upravíme pravdepodobnosť kliknutia na daný výsledok a jemu podobné. Znázornenie časových prechodov je možné na obrázku č. 2.5 pomocou šedých kriviek. V tomto prípade je ako prvá akcia používateľa dopyt. Následne s určitou pravdepodobnosťou môže buď zmeniť svoj dopyt (Q), klikne na výsledok (SR) alebo si zvolí odkaz na podobné vyhľadávanie (RL) a v rámci každého prechodu model zaznamená čas prechodu medzi prechádzajúcim a aktuálnym stavom. Veľkou výhodou tohto prístupu je, že je veľmi dobre škálovateľný a je možné ho teda nasadiť do priamej prevádzky bez potreby prepočítavania modelu strojového učenia ako je v prípade iných prístupov. Model založený na stavoch potrebuje vidieť do histórie len

niekoľko predchádzajúcich krokov a vie automaticky určiť pravdepodobnosti ďalších akcií.

2.2.4 Diskusia

Ako už aj predchádzajúce prístupy zhrnuli, aktivita používateľa má oveľa kvalitnejšiu množinu metadát pre identifikáciu vyhľadávacích epizód a cieľu používateľa. Široký a viacznačný informačný priestor, ktorý sa skrýva za dopytom používateľa, nám sám používateľ dokáže obmedziť svojími akciami, pretože nám napovedajú, približne ktorý význam daného dopytu mal používateľ na mysli. Výraznou výhodou lexikálnych prístupov je ale veľmi rýchla identifikácia reformulácii dopytov, ktorá nemusí byť úplne presná v prípade aktivity používateľa. Používateľ nemusí vôbec kliknúť na žiaden výsledok a posunúť sa rovno k reformulácii dopytu a z obsahu našich metadát o jeho aktivite nevieme určiť, či ďalší dopyt súvisí s prechádzajúcim alebo nie.

Preto sa v tomto prípade naskytuje možnosť využiť hybridný model, ktorý bude využívať atribúty aktivity používateľa a spojí ich s atribútmi dopytov. Existuje viacero možností ako tieto dva prístupy zakomponovať do modelu

Založeného na množine atribútov
: Tento hybridný prístup pozostáva zo spojenia lexikálnych atribútov a atribútov aktivity používateľa pre model strojového učenia.

Založeného na prepojeniach dopytov cez výsledky
: Tento prístup berie do úvahy prepojenia medzi dopytmi na základe výsledkov, ktoré dopyty zdieľajú. Na základe lexikálnych atribútov a aktivity používateľa upraví relevanciu výsledkov pre dopyty pre konkrétneho používateľa a prepojí dopyty na základe výsledkov, ktoré zdieľajú, pričom váha prepojenia je súčet relevancií spoločných výsledkov.

Oba spomenuté prístupy sú výrazne podobné, ale ich rozdiel spočíva v myšlienke. Zatiaľ čo prvý prístup uvažuje zoznam atribútov a počíta podobnosť medzi dopytmi, druhý hľadá cestu medzi dvoma dopytmi. Na základe toho je druhý prístup flexibilnejší, pretože na základe vypočítanej relevancie je možné zahrnúť uvažované atribúty ako atribúty modelu strojového učenia, ale keďže výsledky spolu s ich relevanciou pre používateľa reprezentujú určitý graf, kde váha prepojenia dopytov a výsledkov je určená na základe relevancie výsledkov vypočítanej pre konkrétneho používateľa. Preto pre druhý prístup je možné použiť modely strojového učenia, ale aj iné prístupy ako komplexné grafové algoritmy pre nájdenie podobnosti medzi dopytmi a uvedenú podobnosť použiť samostatne alebo ako ďalší atribút modelu strojového učenia.

3 Strojové učenie

Pre riešenie problému v informatike potrebujeme algoritmus. Keďže algoritmus predstavuje presné vyjadrenie generického postupu riešenia problému, vieme ho veľmi jednoducho prepísať do zdrojového kódu. Problém však nastáva v prípade, ak nemáme reálne overený algoritmus na riešenie problému, pretože problém nemusí byť riešiteľný v reálnom výpočtovom čase alebo riešenie daného problému nemá pevné stanové pravidlá, ktoré by boli vyjadriteľné pomocou algoritmu. Typickým príkladom problému, ktorý nie je možné riešiť konvenčným prístupom vo forme algoritmu, je označenie emailovej správy ako nevyžiadanej pošty (angl. *spam*). Nevyžiadanej pošty je dnes veľké množstvo, preto namiesto algoritmu môžeme využiť dáta. Pomocou množiny dát, o ktorej vieme do akej kategórie patrí, t.j. či sa jedná o nevyžiadanú poštu alebo nie, vytvoríme program, ktorý sa naučí vzory v dátach a dokáže pomocou hľadania daných vzorov automaticky zaradiť nové dáta do príslušnej kategórie [1]. Tento prístup sa nazýva strojové učenie.

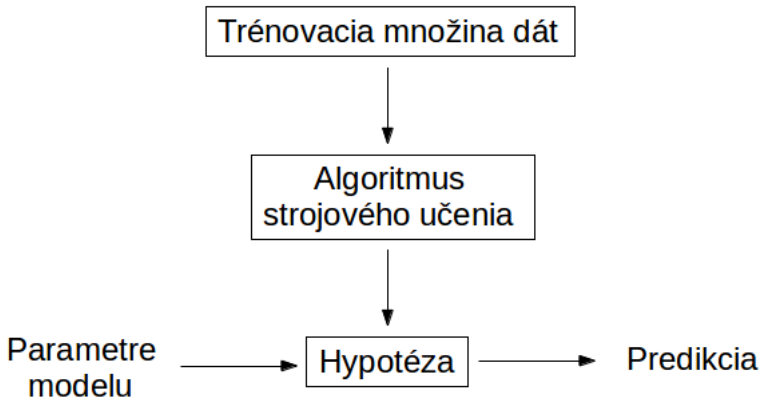
Definícia 3. *Strojové učenie (angl. Machine Learning) je odvetvie umelej inteligencie, ktoré sa zameriava na konštrukciu a štúdium systémov, ktoré sú schopné učiť sa z dát. Cieľom tohto odvetvia je tvorba algoritmov, ktoré sa učia určovať vlastnosti alebo predikovať udalosti na základe učenia sa zo záznamov z minulosti. Učenie sa väčšinou zameriava na pozorovanie záznamov z minulosti a nachádzanie spoločných vzorov alebo iných atribútov, ktoré dokážu čo najpresnejšie určiť budúci stav. [1]*

Jedným z prístupov strojového učenia je analýza a spracovanie dát, o ktorých je známe, do akej kategórie patria. Na základe tejto analýzy potom stroj vie aproximovať zaradenie pre nové dáta [1]. Aproximácia zaradenia nemusí byť úplne presná, ale na základe identifikácie súvislostí a vzorov v dátach na veľkom množstve trénovacích dát sa veľkosť chyby zmenšuje.

Strojové učenie využíva teóriu štatistiky na tvorbu matematických modelov, ktoré opisujú vzťah dát. Postup pri tvorbe učiaceho sa algoritmu pozostáva z [1]

- spracovania veľkého množstva dát určených na trénovanie parametrov matematického modelu,
- overenie nastavenia parametrov modelu na testovacej vzorke dát a

- optimalizácia parametrov pre znížení chyby.



Obr. 3.1: Základný prehľad postupu analýzy dát pri strojovom učení (zdroj: http://ml-class.org)

3.1 Strojové učenie ako riešenie identifikácie vyhľadávacích epizód

Viaceré prístupy [36, 40, 22, 21, 43, 16, 8, 35, 49, 11, 17] využili strojové učenie pre identifikáciu vyhľadávacích epizód na základe atribútov kontextu vyhľadávania. Medzi tieto atribúty zaraďujeme lexikálne vlastnosti dopytov, rozšírenia významov dopytov a aktivitu používateľa počas vyhľadávania. Autori sa pri definovaní modelov v prevažnej miere zamerali na využitie overených algoritmov strojového učenia.

3.1.1 GBDT – Gradient Boosted Decision Trees

Rozhodovacie stromy sú jednou zo základných metód strojového učenia pre klasifikáciu, ale v niektorých prípadoch môže ich predikcia byť výrazne slabá. Friedman a ďalší [13] uviedli metódu *Gradient Boosting* pre vylepšenie slabých modelov predikcie, typicky rozhodovacích stromov, pričom táto metóda berie do úvahy klasický problém učenia s učiteľom a namiesto tvorby jedného komplexného stromu vytvára *les* rozhodovacích stromov, ktoré predstavujú slabé modely. Výsledný model je kombináciou jednotlivých rozhodovacích stromov na základe uvedeného vzťahu

$$F(x, \beta, \alpha) = \sum_{i=1}^{n} \beta_i h(x, \alpha_i) \tag{3.1}$$

kde h predstavuje jednotlivé slabé predikcie. Sumárny výsledok jednotlivých slabých predikcii prezentuje v konečnom dôsledku lepšiu predikciu ako samotný slabý

model rozhodovacieho stromu ako celku. Metódu *Gradient Boosted Decision Trees* využili viaceré práce [36, 16].

3.1.2 SVM – Support Vector Machine

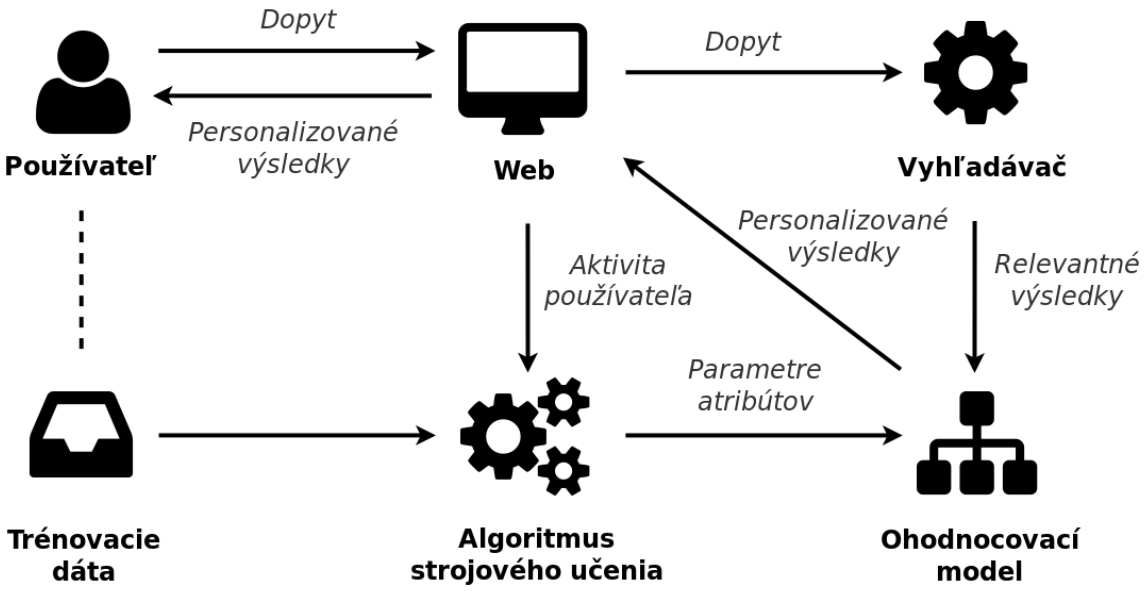
Algoritmy podporných vektorov (angl. *Support Vector Machine*) sa snažia nájsť efektívnu lineárnu hranicu pre opis dát. Hlavnou výhodou *SVM* je, že dokážu pracovať aj s veľmi zložitými nelineárnymi funkciami pomocou prevodu pôvodného priestoru dát do viacdimenzionálneho, kde je možné triedy oddeliť lineárne. Metóda *SVM* je veľmi rozšírená v oblasti strojového učenia, avšak prípade modelovania záujmov používateľa pre identifikáciu vyhľadávacích epizód a celkovej personalizácie vyhľadávania je nutné, aby sa model neustále prispôsoboval meniacim sa záujmom používateľa. Model preto musí byť flexibilný a musí dokázať zachytiť aj krátkodobé záujmy používateľa. Pre tieto potreby bol vyvinutý rozšírený algoritmus *Ranking SVM*.

3.1.3 Learning to Rank

Learning to Rank je prístup, ktorý využíva metódy strojového učenia pre lepšie identifikovanie relevantných dokumentov na základe zoznamu atribútov, ktoré dané dokumenty opisujú. Pomocou daných atribútov je potom možné rozoznať, ktoré dokumenty sú relevantnejšie oproti ostatným. Medzi tieto atribúty môže patriť kategória, ktorú preferujeme, webový portál, ktorý často navštevujeme alebo autor, ktorého práce nás zaujímajú. Na to, aby stroj mohol vyhodnotiť význam týchto atribútov, potrebuje vedieť, akých autorov máme radi a ktoré webové portály si často prezeráme.

Pokiaľ uvažujeme aktivitu používateľa ako atribút podobnosti dopytov, tak metóda *Learning to Rank* je nápomocná v prípade výpočtu relevancie dokumentov pre lepšiu identifikáciu podobnosti dopytov na základe výsledkov vyhľadávania. Pokiaľ by sme uvažovali podobnosť len na základe relevantnosti výsledkov určených podľa vyhľadávača, nebrali by sme do úvahy význam dopytu pre používateľa. Na základe aktivity používateľa reprezentovanej ako kliknutia na výsledky alebo čas strávený prezeraním obsahu konkrétnych výsledkov dokážeme odhaliť preferencie používateľa pre konkrétne výsledky a získame lepší pohľad na to, ktoré výsledku sú relevantné práve pre daného používateľa.

Na rozdiel od iných prístupov, *Learning to Rank* sa snaží prispôsobiť charakteristikám používateľa, pričom berie do úvahy relevanciu, ktorú má daný dokument pre používateľa na základe uvažovaného zoznamu atribútov. Ako je možné vidieť na obrázku 3.2, prístup sa prispôsobuje aktivitám používateľa počas vyhľadávania, pričom na základe nich vytvára ohodnocovací model.



Obr. 3.2: Prehľad architektúry Learning to Rank pre personalizáciu vyhľadávania

Na základe obrázku 3.2 je možné identifikovať fázy postupu pri definovaní modelu strojového učenia:

1. Určenie základných atribútov dokumentov a modelu používateľa pre model strojového učenia,
2. určenie parametrov modelu na základe trénovacích dát,
3. personalizácia dokumentov na základe naučených parametrov modelu pre nové dáta.

3.1.4 Ranking SVM

Algoritmus *Ranking SVM* [19] sa prevažne sústreďuje na problém ohodnotenia výsledkov vyhľadávania v prípade vyhľadávačov a všeobecne v oblasti získavania informácii (angl. *Information Retrieval*). Hlavnou výhodou tohto algoritmu je rýchle prispôsobenie sa meniacim sa atribútom, ktoré je charakteristické pre modelovanie krátkodobých záujmov používateľa počas vyhľadávania. *Ranking SVM* tento stav dosahuje neustálym upravovaním atribútov modelu na základe meniaceho sa kontextu vyhľadávania.

Pokiaľ berieme do úvahy kliknutie na výsledok vyhľadávania ako vyjadrenie používateľa o relevantnosti výsledku [40, 19], môžeme pomocou implicitnej spätnej väzby od používateľa prispôsobiť usporadúvanie výsledných dokumentov. Napriek tomu je dôležité spomenúť, že výsledok vyhľadávania nie je nezaujatý parameter [45]. Keďže výsledky vyhľadávania sú prezentované zoznamom, relevantné výsledky

pre používateľa nachádzajúce sa na spodných pozíciách zoznamu sú čiastočne znevýhodnené oproti prvým pozíciam. Preto je dôležité pri vyhodnocovaní relevancie pre používateľa brať do úvahy nielen samotnú akciu ako kliknutie alebo čas strávený prezeraním výsledku, ale využiť aj samotnú pozíciu výsledku spolu s generickou relevanciou určenou vyhľadávačom.

Joachims a ďalší [20, 45] identifikovali viaceré možné prístupy ku vyhodnocovaniu relevantnosti výsledkov podľa akcií používateľa. Na základe ich pozorovaní identifikovali, že posledné kliknutie na výsledok je vždy najrelevantnejšie ako predchádzajúce kliknutia. Prístup uvažuje pre dopyt q zoznam výsledkov

$$p_1^*, p_2^*, p_3, p_4^*, p_5, p_6, p_7^* \tag{3.2}$$

kde p_i reprezentuje výsledok vyhľadávania a p_i^* reprezentuje kliknutý výsledok vyhľadávania. Potom prístup posledného kliknutia nazvaný ako $LastClick > SkipAbove$ uvažuje, že výsledok p_7 je relevantnejší ako p_3, p_5, p_6, avšak neuvažuje upravené relevancie pre ostatné kliknuté výsledky – p_1, p_2 a p_4.

Po úprave relevancie výsledkov vyhľadávania môžeme porovnať podobnosť dopytov s uvážením významu dopytu pre používateľa. Ak zoberieme do úvahy dopyt *Java*, v prípade ktorého používateľ klikol na výsledok o programovacom jazyku, tak o ďalšom dopyte *Java source* zdieľajúcom niektoré výsledky z predchádzajúceho dopytu vieme, že sa viaže na zdrojové kódy programovacieho jazyka Java. Preto identifikuje vyhľadávaciu epizódu významnú pre daného používateľa a zároveň personalizujeme aj jeho výsledky vyhľadávania.

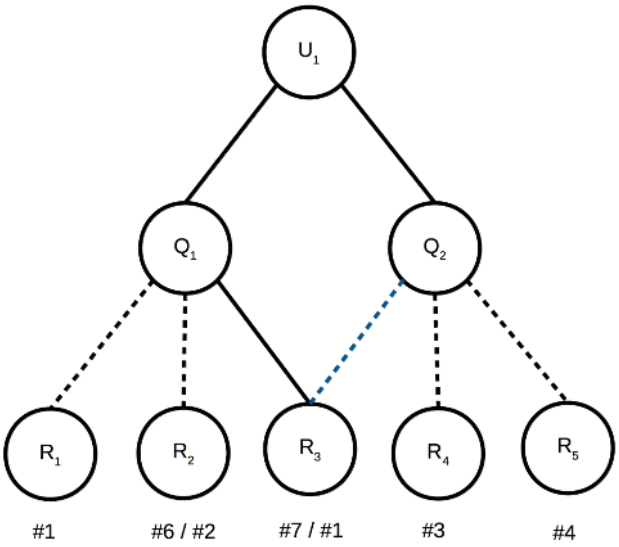
4 Návrh

Aktivita používateľa počas vyhľadávania zahŕňa zadávanie dopytu, prezeranie výsledkov, kopírovanie textu a čas strávený týmito akciami. Všetky uvedené záznamy o aktivite používateľa nám vedia napovedať niečo o tom, o ktoré výsledky vyhľadávania sa používateľ zaujíma, t.j. ktoré výsledky sú z jeho pohľadu naozaj relevantné pre jeho dopyt. Ak predpokladáme, že dva dopyty sú si podobné, keď zdieľajú výsledky vyhľadávania [32], môžeme počet zdieľaných výsledkov pokladať za mieru podobnosti dvoch dopytov. Pokiaľ rozšírime túto podobnosť o aktivitu používateľa, zvýšime tak význam podobnosti pre konkrétneho používateľa. To znamená, že ak dané dopyty zdieľajú výsledky, ktoré si používateľ naozaj aj prezrel a strávil na nich dlhší čas, môžeme predpokladať podobnosť s väčšou istotou [29, 18, 30].

Naša metóda uvažuje mieru zdieľania výsledkov vyhľadávania medzi dopytmi a aktivitu používateľa na týchto výsledkoch ako hlavný zdroj podobnosti medzi dopytmi. Náš prístup spočíva v uvažovaní prepojenia dopytov a výsledkov v obmedzení na používateľa. Na základe toho je význam prepojenia dopytu odlišný pre dvoch rôznych používateľov, pretože obaja interagovali s daným výsledkom inak – to znamená, že jeden používateľ daný výsledok zvolil a strávil na ňom dlhší čas, pričom druhý používateľ daný výsledok preskočil. Rozšírením miery podobnosti dvoch dopytov na základe zdieľaných výsledkov o aktivitu používateľa ako kliknutie na výsledky a čas strávený prezeraním výsledkov, predstavíme presnejší prístup pre získanie podobnosti dopytov.

Ďalší význam aktivity plynie priamo z obmedzenia na konkrétneho používateľa, pretože dva dopyty sa môžu mať rovnaký cieľ, avšak lexikálne sa môžu výrazne odlišovať. Ako je možné vidieť na obrázku č. 4.1, dva dopyty pre používateľa zdieľajú iba jeden výsledok. Pri jednom z dopytov sa daný výsledok nachádzal na nižších pozíciách a v prípade druhého dopytu bol výrazne relevantný. Keďže dané dva dopyty zdieľajú iba jeden výsledok a zároveň s odlišnou relevanciou, tak intuitívne uvažujeme, že dané dva dopyty nie sú veľmi podobné. Avšak, ak zahrnieme do úvahy aj aktivitu používateľa, zistíme, že používateľ sa zaujímal práve o daný menej relevantný výsledok. Preto aktivita dokáže napovedať výrazne viac o relevancii výsledku a podobnosti dopytov pre konkrétneho používateľa.

Na základe toho uvažujeme v rámci našej metódy aktivitu na zdieľaných vý-

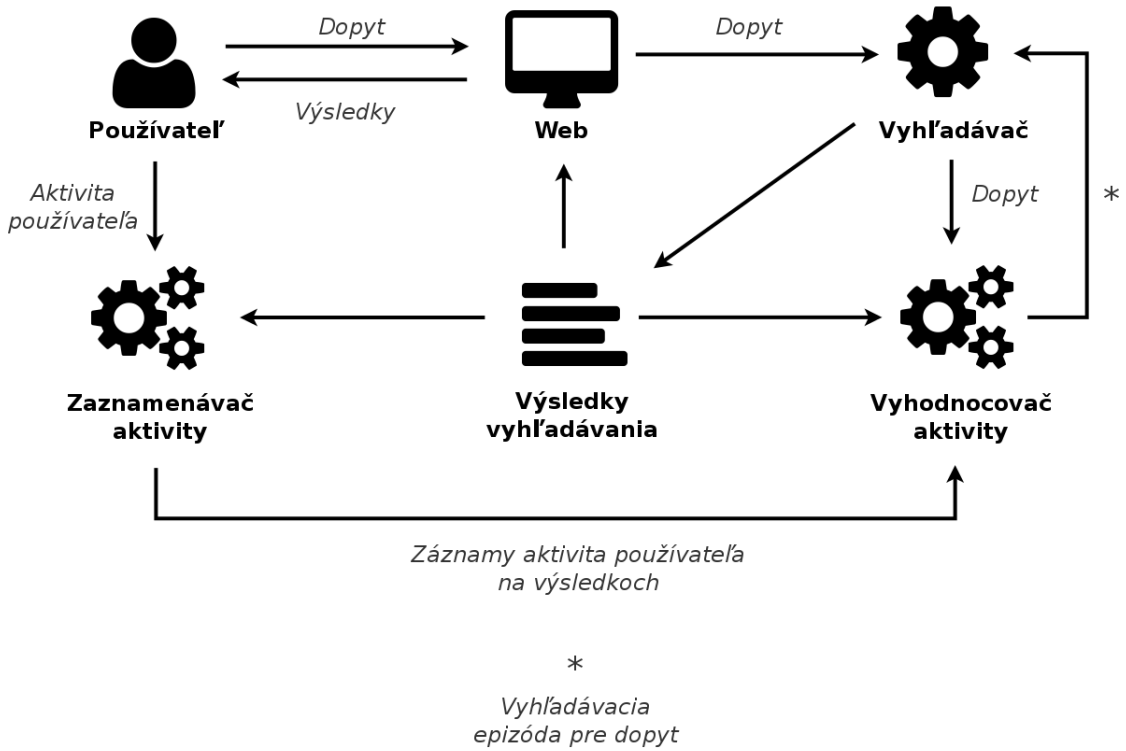


Obr. 4.1: Prepojenie dopytov pre používateľa na základe jeho akcií. Plná čiara reprezentuje kliknutie na výsledok a čiarkovaná čiara reprezentuje zaradenie medzi výsledkami vyhľadávania

sledkoch dopytov relatívne ku konkrétnym používateľom. Pri výsledkoch a aktivite používateľa uvažujeme viaceré atribúty, ktoré prispievajú k zvýšeniu podobnosti dopytov. Medzi dané atribúty zahrňujeme

- pozíciu výsledku v zozname výsledkov,
- kliknutie na výsledok,
- čas strávený prezeraním výsledku,
- počet zdielaných výsledkov a
- počet spoločných kliknutých výsledkov.

Ako je možné vidieť aj na obrázku č. 4.2 predstavujúceho prehľad našej metódy, interakcia používateľa s výsledkami vyhľadávania je zaznamenávaná počas vyhľadávania a vyhodnocovaná pri každej zmene dopytu alebo novej akcii používateľa. Výpočet podobnosti medzi dopytmi následne prebieha na základe zdieľaných výsledkov medzi dopytmi. Vyberieme N existujúcich epizód a ich posledné dopyty porovnáme s aktuálnym a aktuálny dopyt zaradíme do epizódy najrelevantnejšieho dopytu.



Obr. 4.2: Prehľad našej metódy, ktorá zaznamenáva aktivitu používateľa a na základe tejto aktivity zaradí dopyt do vyhľadávacej epizódy

4.1 Definícia vyhľadávacej epizódy

Definícia 4. *Vyhľadávacia epizóda (angl. search session) prestavuje všetku aktivitu používateľa s rovnakým alebo podobným cieľom v určitom časovom intervale.*

Na základe toho reprezentujeme vyhľadávaciu epizódu ako

$$S = \{A_1, A_2, A_3, ...A_n\} \tag{4.1}$$

kde A_i reprezentuje akciu používateľa, ktorá môže byť buď zadanie dopytu alebo kliknutie na výsledok. Každá akcia je ohraničená časom a na základe toho časové rozpätie akcie definujeme ako rozdiel časov dvoch akcií

$$t(A_n) = T_{A_{n+1}} - T_{A_n} \tag{4.2}$$

kde T_{A_n} reprezentuje čas, kedy bola akcia A_n vytvorená. Na základe toho definujeme celú dĺžku vyhľadávacej epizódy ako

$$t(S) = \sum_{A \in S} t(A) \tag{4.3}$$

4.2 Postup metódy

Naša metóda sa zameriava na využitie strojového učenia pre zaradenie dopytov do vyhľadávacích epizód. Na základe toho reprezentujeme podobnosť dopytov ako dvojice, ktorých podobnosť je vyjadrená pomocou zoznamu vlastností a na základe modelu strojového učenia zostaveného z existujúcich dopytov zaraďujeme nové dopyty do vyhľadávacích epizód. Model strojového učenia pozostáva zo zoznamu vlastností, ktorým sa podrobnejšie venujeme v kapitole 4.2.2. Model sa učí význam podobných vlastností dopytov a na základe naučených hraníc vlastností určí, či sa dva dopyty podobajú alebo nie. Výsledkom jeho učenia je teda model, ktorý dokáže pre dva dopyty zistiť, či patria do rovnakej vyhľadávacej epizódy.

4.2.1 Tvorba dvojíc dopytov

Zdroj znalostí pre algoritmus strojového učenia reprezentujeme ako dvojice dopytov

$$(Q_1, Q_2) = \{F_1, F_2, F_3, F_4, \dots F_n\} \tag{4.4}$$

kde Q_1 a Q_2 reprezentujú dopyty a $F_{1\dots n}$ reprezentujú hodnoty vlastností pre dané dopyty. Uvedené dvojice vytvárame pre všetky známe dopyty. Na základe toho naša metóda vždy spracováva

$$\binom{n}{2} = \frac{n!}{2(n-2)!} = \frac{n(n-1)}{2} \tag{4.5}$$

dvojíc dopytov, kde n reprezentuje počet dopytov považovaných za zdroj znalostí pre algoritmus strojového učenia a zároveň uvedený vzťah opisuje aj výpočtovú zložitosť kombinácií daných dopytov.

4.2.2 Výpočet vlastností dvoch dopytov

Po vytvorení dvojíc dopytov naša metóda vypočíta pre každú dvojicu mieru podobnosti na základe hodnoty ich spoločných vlastností. Zoznam týchto vlastností a vzájomnej podobnosti slúži ako vstup pre model strojového učenia. Na základe hodnôt jednotlivých vlastností sa model strojového učenia naučí, pri akej hladine jednotlivých vlastností dva dopyty patria do spoločnej vyhľadávacej epizódy. Pre výpočet spoločných vlastností dvoch dopytov sme zvolili uvedené vlastnosti:

- *čas* – rozdiel času medzi dvomi dopytmi reprezentovaný ako počet sekúnd,
- *normalizovanú Levenshteinovu vzdialenosť* dvoch dopytov z kapitoly (2.1),

- *kosínusovú podobnosť* a *Jaccardov koeficient podobnosti* [2], ktoré hľadajú mieru podobnosti medzi vektormi slov z dopytov a
- podobnosť dopytov na základe výsledkov vyhľadávania a aktivity používateľa podľa vzťahu 4.7.

Uvedené vlastnosti reprezentujú vektor vlastností vo vzťahu 4.4. Čas ako vlastnosť určuje podobnosť dopytov pomocou predikátu

$$T(Q_1, Q_2) = \begin{cases} |t(Q_2) - t(Q_1)| > 26min. : 0 \\ |t(Q_2) - t(Q_1)| \leq 26min. : 1 \end{cases} \tag{4.6}$$

ktorý určí, že dva dopyty sú podobné ak rozdiel ich času zadania do vyhľadávača je menej ako 26 minút. Tento interval identifikovali autori [32] ako najúspešnejšiu hodnotu tohto temporálneho atribútu.

4.2.3 Podobnosť dopytov na základe aktivity používateľa

Na základe zdieľaných výsledkov medzi dopytmi a aktivity používateľa na nich definujeme podobnosť dopytov ako

$$RS(Q_1, Q_2) = |R_s| + |R_c| + \sum_{R \in R_c} log(t(R) + 1) \frac{pos(R_{Q_1})}{pos(R_{Q_2})} \tag{4.7}$$

kde R_s reprezentujú zdieľané výsledky a R_c zdieľané kliknuté výsledky medzi dopytmi Q_1 a Q_2. Atribút $t(R)$ reprezentuje čas strávený prezeraním výsledku R v sekundách, *pos* vyjadruje pozíciu výsledku v zozname výsledkov a $pos(R_{Q_1})$ a $pos(R_{Q_2})$ vyjadruje pozíciu výsledku R v rámci konkrétnych dopytov. Pomer

$$\frac{pos(R_{Q_1})}{pos(R_{Q_2})} \tag{4.8}$$

vyjadruje posun pozície výsledku medzi dvomi dopytmi a preferuje tak výsledky, ktoré sa z nízkych pozícií presunuli v nasledujúcom dopyte na vysoké. Podobný prístup aplikuje v prípade výsledkov, ktoré v rámci prvého dopytu boli na vysokých pozíciach, ale v druhom dopyte sa posunuli na nižšie. Takéto výsledky vzťah znevýhodní.

4.2.4 Zaradenie dopytu do vyhľadávacej epizódy

Posledný krok našej metódy predstavuje zaradenie dopytu do konkrétnej vyhľadávacej epizódy. Pre nový dopyt z histórie vyhľadávacích epizód vyberieme N vyhľadávacích epizód, ktoré patria danému používateľovi. Tieto epizódy predstavujú kandidátov, do ktorých môžeme zaradiť nový dopyt. Pomocou vytvoreného modelu

strojového učenia identifikujeme podobnosť medzi novým dopytom a poslednými dopytmi z vyhľadávacích epizód. Epizódy, ktoré model identifikoval ako podobné zoradíme podľa vzťahu 4.7 a vyberieme tú epizódu, pre ktorú ma vzťah najvyššiu hodnotu. Nový dopyt zaradíme do danej epizódy.

4.3 Diskusia k návrhu metódy

Jednou z hlavných častí vzťahu 4.7 je pomer pozícii výsledkov a čas strávený prezeraním konkrétneho výsledku, pretože tieto atribúty reprezentujú aktivitu používateľa a posun vo význame dopytov. Ak uvažujeme, že používateľ klikne na výsledok, ktorý sa nachádza na nižších pozíciách, pretože dopyt je viacznačný a pre ďalší dopyt je tento istý výsledok tiež relevantný, ale nachádza sa na vyššej pozícii, uvedený pomer pozícií výsledkov 4.8 tento výsledok zvýhodní pri výpočte podobnosti. Myšlienka za uvedeným pomerom podporuje nielen reformulácie dopytov, ale aj hľadanie významu viacznačných dopytov pre konkrétneho používateľa.

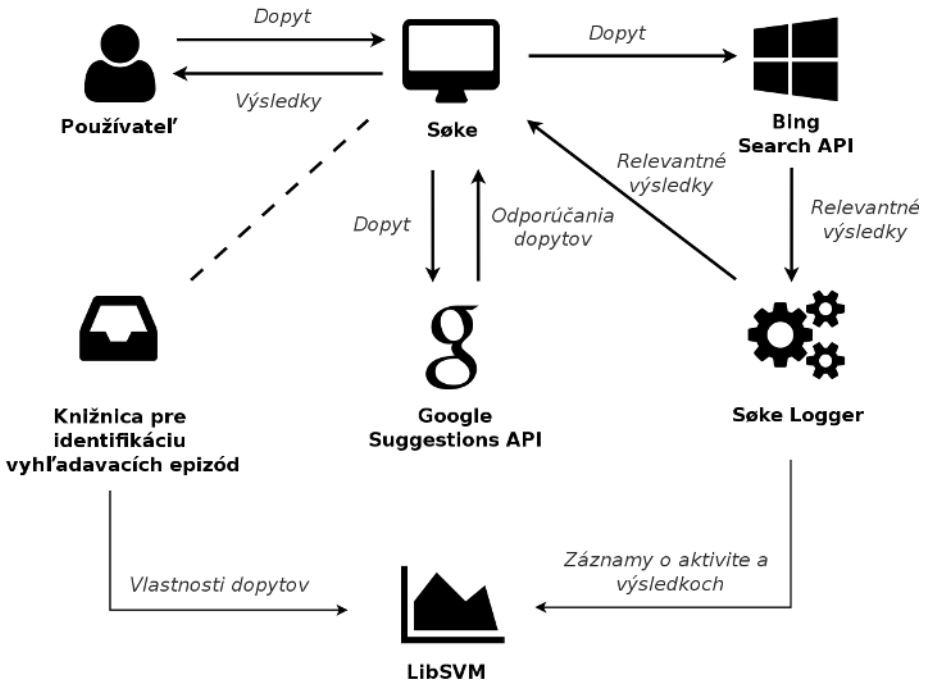
Náš prístup využíva strojové učenie, a preto sme reprezentáciu vyhľadávacích epizód prispôsobili pre tvorbu modelu strojového učenia. Napriek tomu, hlavnú časť našej metódy predstavuje spôsob určenia podobnosti medzi dopytmi reprezentovaný vzťahom 4.7. Na základe toho môžeme tento vzťah využiť pre výpočet podobnosti aj v prípade iných prístupov ako sú napríklad grafové algoritmy alebo iné algoritmy strojového učenia ako zhlukovanie (angl. *clustering*). V prípade grafových algoritmov môžeme uvažovať prepojenie dopytov pomocou výsledkov a vzťah 4.7 ako váhu hrán prepojenia. V rámci zhlukovania by sme invertovaním vzťahu 4.7 mohli určovať vzdialenosť dvoch dopytov, a tým pádom aj mieru ich podobnosti.

5 Realizácia

Realizáciu našej metódy sme rozdelili na dve časti, ktoré pozostávajú z

- vyhľadávača, ktorý v reálnom čase zaznamenáva aktivitu používateľov a na základe ich spätnej väzby zaraďuje dopyty do vyhľadávacích epizód a
- knižnice pre identifikáciu vyhľadávacích epizód pomocou našej metódy.

Prehľad celej architektúry je možné vidieť na obrázku 5.1. Prepojenie medzi knižnicou a vyhľadávačom nie je realizované, pretože knižnica bola zatiaľ navrhnutá ako simulátor vyhľadávacích epizód pre potreby overenia nami navrhnutého modelu. Knižnica môže byť navrhnutá ako rozšírenie pre vyhľadávač, ktoré bude fungovať na rovnakom architektonickom modeli ako zaznamenávač výsledok vyhľadávania alebo aktivity používateľa, pričom bude rozširovať natrénovaný model o nové dáta o aktivite používateľa a v reálnom čase zaraďovať dopyty do vyhľadávacích epizód bez potreby spätnej väzby používateľa.



Obr. 5.1: Architektúra nástrojov použitých v realizácie našej metódy

5.1 Søke – platforma pre tvorbu anotovaných vyhľadávaní

Søke je open-source vyhľadávač [1] postavený na rámci pre vývoj webových aplikácii Ruby on Rails [2] a databáze PosgreSQL [3]. Implementáciu webovej aplikácie rozširujú dva moduly pre

- *Google Suggestions* – modul pre interakciu so službou Google Suggestions [4] a
- *Bing* – modul pre interakciu so službou Bing Search API pre získanie výsledkov vyhľadávania [5].

Rozhranie vyhľadávača je možné vidieť na obrázku 5.2. V rámci rozhrania môžu používatelia ručne zaradiť každý nový dopyt do príslušnej vyhľadávacej epizódy a ponúknuť tak priamu spätnú väzbu o následnosti ich dopytov. Vyhľadávač sleduje celú ich aktivitu počas vyhľadávania. Ako prvý krok zaznamenáva dopyt, ktorý používatelia zadajú. Pri zadávaní dopytov používateľom ponúka nápovedy, ktoré sú získané cez službu *Google Suggestions* pomocou spomínaného modulu. Výsledky vyhľadávania sú nájdené pomocou služby *Bing Search API*. Pre všetky nájdené výsledky sa zaznamenávajú atribúty ako

- pozícia,
- URL adresa,
- krátky úryvok textu stránky a
- názov stránky.

Keď používateľ klikne na konkrétny výsledok, čas kliknutia na daný výsledok sa zaznamená a priradí sa konkrétnemu výsledku.

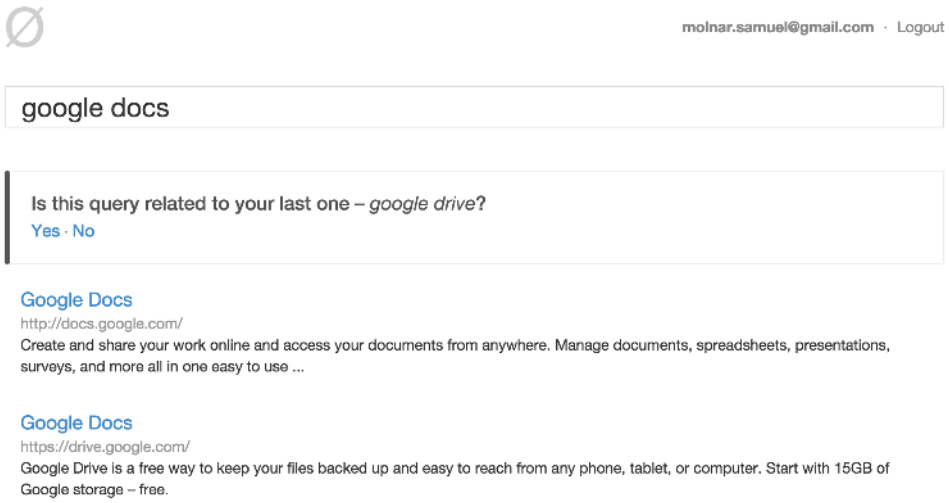
Po zadaní dopytu nasleduje najdôležitejší krok vyhľadávania. Každý nový dopyt v našom systéme predstavuje novú vyhľadávaciu epizódu, až pokým používateľ nevyjadrí explicitne inak. Možnosť ponúknuť explicitnú spätnú väzbu a zaradiť dopyt do vyhľadávacej epizódy zobrazí vyhľadávač používateľom ako výzvu nad výsledkami vyhľadávania. Zaradenie je rozdelené do dvoch krokov.

[1] Søke: https://github.com/smolnar/soke
[2] Ruby on Rails: http://rubyonrails.org/
[3] PostgreSQL: http://www.postgresql.org/
[4] Google Suggestions: https://developers.google.com/web-search/docs/
[5] Bing Search API: https://datamarket.azure.com/dataset/bing/search

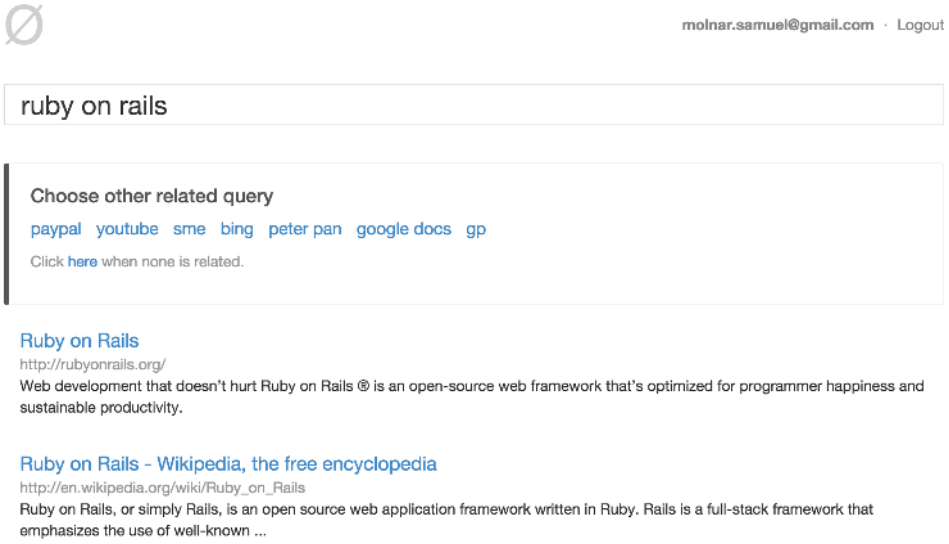
Obr. 5.2: Rozhranie vyhľadávača Søke

1. **Prvý krok** – ako je možné vidieť na obrázku 5.3, v rámci prvého kroku zaradenia dopytu používateľ vyjadrí, či jeho aktuálny dopyt je relevantný s jeho predchádzajúcim dopytom. Pokiaľ sa vyjadrí, že jeho aktuálny dopyt súvisí s prechádzajúcim, okno overenia sa vytratí a nový dopyt sa zaradí do vyhľadávacej epizódy prechádzajúceho dopytu. Pokiaľ ale jeho aktuálny dopyt nesúvisí s prechádzajúcim dopytom, nasleduje druhý krok zaradenia.

2. **Druhý krok** – v rámci druhého kroku zaradenia má používateľ vybrať zo zoznamu dopytov, ku ktorým je aktuálny dopyt relevantný. Ako je možné vidieť na obrázku 5.4, používateľ vidí niekoľko dopytov z rôznych nedávnych vyhľadávacích epizód. Môže zaradiť dopyt do jednej z nich alebo explicitne vyjadriť, že dopyt reprezentuje úplne novú epizódu.



Obr. 5.3: Rozhranie pre zaradenie dopytu do prechádzajúcej vyhľadávacej epizódy



Obr. 5.4: Rozhranie pre zaradenie dopytu do inej ako predchádzajúcej vyhľadávacej epizódy

5.2 Knižnica pre simuláciu identifikácie vyhľadávacích epizód

Naša knižnica je postavená na knižnici *LibSVM* [6] a jej implementácii *rb-libsvm* [7] v programovacom jazyku Ruby. Knižnica spracováva dáta z vyhľadávača Søke, ktoré sú reprezentované vo formáte JSON. Hlavnými časťami knižnice sú moduly

- *Sampler*, ktorý obsahuje viacero stratégii pre výber vyhľadávacích epizód pre overenie,
- *Metrics*, ktorý implementuje algoritmy pre určenie spoločných vlastností medzi dopytmi ako Jaccardov koeficient podobnosti alebo kosínusová podobnosť,
- *Model*, ktorý obsahuje viacero stratégii pre výpočet modelu, ako napríklad *LibSVM* a
- *Evaluator*, ktorý orchestruje ostatné moduly a vykonáva overenie.

V rámci modulu *Metrics* sme využili viacero dostupných implementácii algoritmov. Pre výpočet Jaccardovho koeficientu využívame knižnicu *jaccard* [8] a algoritmus Levenshteinovej podobnosti sme prebrali z existujúcich open-source implementácii[9], pričom sme ho upravili, aby vyhovoval normalizovanej verzii definovanej v 2.1.

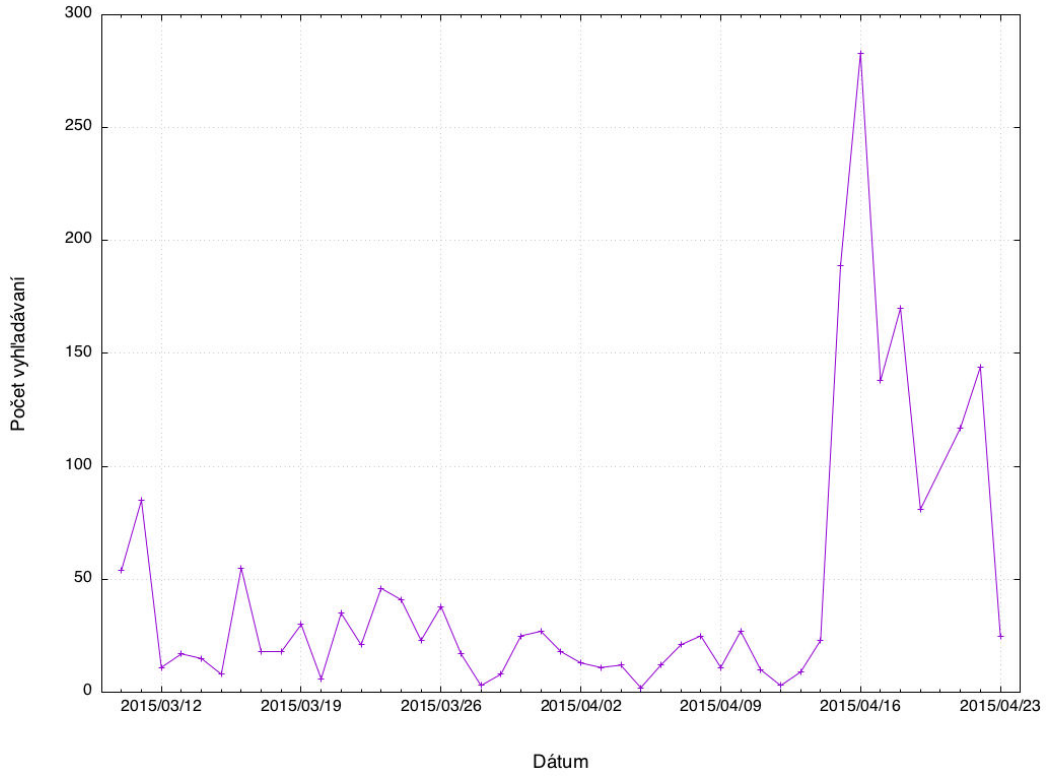
[6] LibSVM: http://www.csie.ntu.edu.tw/ cjlin/libsvm/
[7] rb-libsvm: https://github.com/febeling/rb-libsvm
[8] jaccard: `https://github.com/francois/jaccard`
[9] Levenshtein: `http://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance`

6 Overenie

Overenie našej metódy sme realizovali na dátach, ktoré sme zaznamenali pomocou vyhľadávača Søke počas jeho prevádzky od 10. marca 2015 až do 23. apríla 2015. Nad dátami sme realizovali štúdiu, aby sme zistili, ako používatelia vyhľadávajú a aký je pomer počtu dopytov a zdieľaných výsledok v rámci ich vyhľadávacích epizód. Na uvedenej štúdie sme usúdili vhodnosť daných dát pre naše overenie.

6.1 Používateľská štúdia vyhľadávača Søke

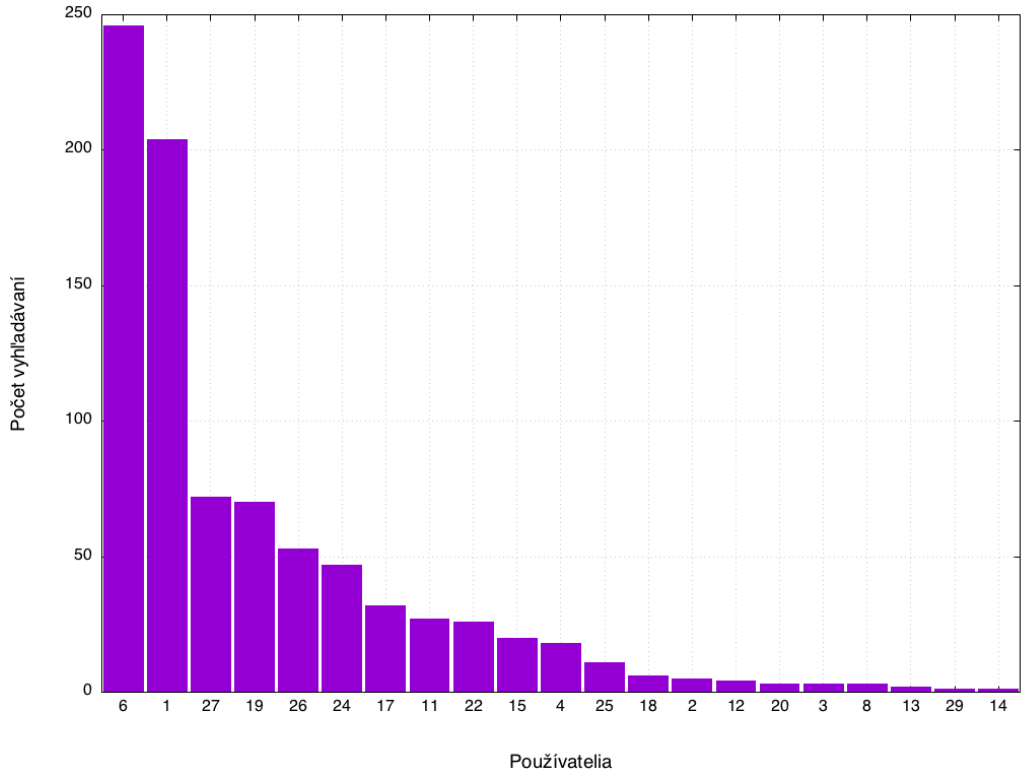
Cieľom našej používateľskej štúdie bola analýza správania používateľov pri vyhľadávaní a odhalenie charakteru ich dopytov a miery zdieľania výsledkov medzi dopytmi.



Obr. 6.1: Počet dopytov počas jednotlivých dní od 10. marca 2015 do 23. apríla 2015

Vyhľadávač bol uvedený do prevádzky 10. marca 2015. Od spustenia do dňa

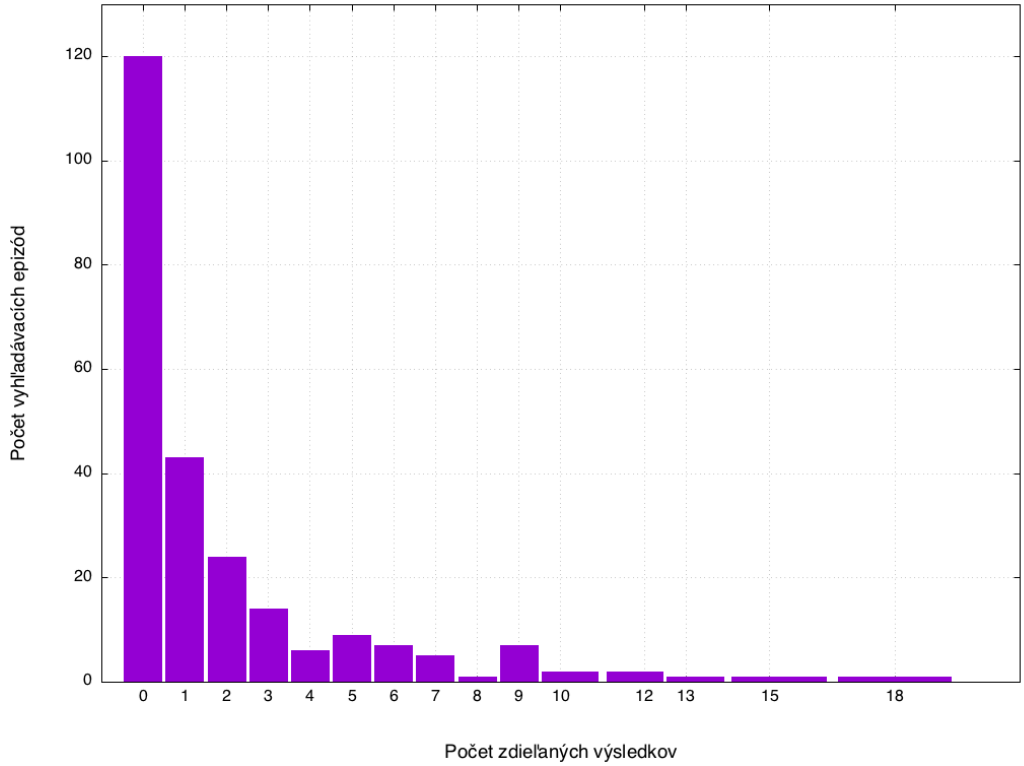
vyhodnotenia (23. apríl 2015) sa v ňom zaregistrovalo 30 používateľov. Spolu títo používatelia vyhľadávali 2 261-krát a použili pri tom 2 099 unikátnych dopytov. Na obrázku 6.2 je možné vidieť histogram aktivity jednotlivých používateľov podľa počtu ich dopytov a na obrázku 6.1 je možné vidieť rozloženie počtu dopytov naprieč obdobím prevádzky vyhľadávača. V období okolo 15. marca 2015 sme zaznamenali zvýšenie aktivity a počtu dopytov, pretože sme oslovili väčšiu skupinu študentov, aby náš vyhľadávač používala ako svoj základný. Ako si ale môžeme všimnúť, ich aktivita pomaly klesala a väčšina nových používateľov pomaly prestala vyhľadávač používať. Pre zobrazené dopyty bolo vrátených spolu 22 656 webových stránok, pričom systém obsahoval 18 621 unikátnych stránok. Zo všetkých vyhľadávaní používatelia zaradili 854 do vyhľadávacích epizód, pričom tak ohodnotili 464 epizód. Medián dĺžok vyhľadávacích epizód predstavoval 2.28 minúty a priemer 144.6 minút, čo značí, že dataset pozostáva prevažne z epizód s krátkodobým zámerom a obsahuje pár vyhľadávacích epizód prekračujúcich niekoľkohodinovú dĺžku.



Obr. 6.2: Histogram počtu vyhľadávaní pre konkrétnych používateľov (meraný v dňoch vyhodnotenia)

Pre určenie vhodnosti dát pre naše overenie sme sa zamerali na identifikovanie dvoch štatistických meraní – počet zdieľaných výsledkov medzi dopytmi a počet kliknutí na výsledky v rámci vyhľadávacích epizód. Pre určenie počtu zdieľaných výsledkov sme z existujúcich epizód v dátach vybrali len tie, ktoré obsahujú aspoň dva dopyty. Takýchto epizód sme identifikovali 243. Na obrázku 6.3 je možné

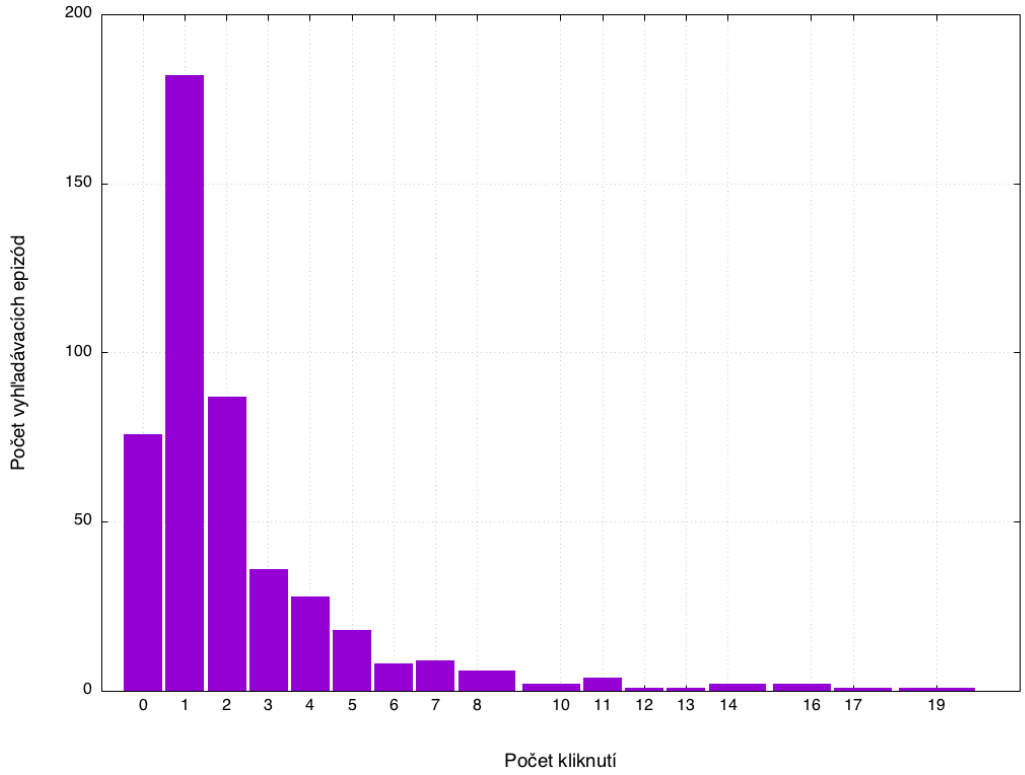
vidieť pomer počtu výsledkov, ktoré sa zdieľali medzi susednými dopytmi v rámci vyhľadávacích epizód. Z celkového počtu 243 epizód, 123 epizód zdieľalo aspoň jeden výsledok medzi dopytmi. Podľa uvedenej štatistiky by naša metóda vedela osamote identifikovať len približne jednu polovicu zo všetkých dopytov v rámci uvedených dát. Preto sme si potvrdili aj predpoklad, že naša metóda pôsobí skôr ako doplnok k už existujúcim riešeniam ako je kosínusová podobnosť alebo iné lexikálne a temporálne vlastnosti dopytov.



Obr. 6.3: Pomer počtu zdieľaných výsledkov v rámci vyhľadávacích epizód (meraný počas dní vyhodnotenia)

Ako druhú mieru vhodnosti sme zvolili počet kliknutých výsledkov v rámci vyhľadávacích epizód. Na základe štatistiky sme zistili, že 388 zo 464 epizód obsahuje aspoň jedno kliknutie na výsledok vyhľadávania. Na obrázku 6.4 je možné vidieť rozdelenie počtu kliknutí medzi epizódami.

Uvedená štúdia preukázala, že podobné dopyty častokrát majú podobné výsledky. Hľadali sme dôvod, prečo niektoré epizódy nezdieľajú žiadny výsledok medzi susednými dopytmi. Po analýze uvedených dopytov sme zistili, že pre niektoré neboli nájdené žiadne výsledky alebo sa jednalo o reformulácie, pretože prvotný dopyt používateľa bol príliš všeobecný a druhý dopyt ho výrazne špecifikoval. Identifikovali sme taktiež, ze 17 epizód obsahovalo čas medzi zadaním svojich dopytov menej ako 5 sekúnd. Na základe toho môžeme uvažovať, že niektoré dopyty boli preklepy, používateľ nebol spokojný s výsledkami alebo sa pri písaní dopytu pomýlil. V prípade



Obr. 6.4: Pomer počtu kliknutých výsledkov vyhľadávania v rámci vyhľadávacích epizód (meraný počas dní vyhodnotenia)

niektorých dopytov sme si všimli, že používatelia sa snažili len skúšať ako systém funguje a zaraďovali tieto skúšobné dopyty aj do epizód. Taktiež musíme brať do úvahy charakter vyhľadávača *Bing*, ktorý mohol uvedené dopyty a relevantnosť výsledkov upravovať pomocou vlastného algoritmu. Keďže vlastnosti, ktoré uvažujeme v rámci nášho modelu dokážu zachytiť reformulácie a lexikálne podobnosti medzi dopytmi a naša metóda má rozširovať ich silu, tak sme usúdili, že dataset svojou mierou zdieľania výsledkov v ohodnotených epizódach je vhodný pre naše overenie. Taktiež má veľký význam v našich dátach fakt, že sa jedná o pravý zlatý štandard, pretože zaradenie dopytov prichádza priamo od konkrétnych používateľov a nie anotátorov, ktorí len zaraďujú dopyty do epizód na základe tušenia, že dané dva dopyty spolu súvisia. Náš dataset obsahuje priame spojenia, ako ich vníma používateľ a aký význam majú dané dopyty práve pre neho.

6.2 Experiment

Overenie našej metódy sme realizovali nad anotovanými dátami od 10. marca 2015 do 23. apríla 2015. Sústredili sme sa na neriadený experiment, kde sme neuvažovali žiadne obmedzenia alebo predpísané úlohy pre používateľov a vyzvali sme ich,

aby vyhľadávač používali prirodzene. Počas tohto obdobia používatelia vytvorili a zaradili svoje dopyty celkovo do 464 vyhľadávacích epizód.

Ako mieru pre overenie sme si zvolili *Precision*, *Recall* a F_β, ktoré predstavujú štandardnú metriku pre overenie vyhľadávacích epizód a ich základné reprezentácie boli autormi prispôsobené ako [14]

$$P = \frac{TP}{TP + FP} = \frac{N_{shift\&correct}}{N_{shift} + N_{contin}} \tag{6.1}$$

$$R = \frac{TP}{TP + FN} = \frac{N_{shift\&correct}}{N_{true\_shift}} \tag{6.2}$$

$$F_\beta = \frac{(1 + \beta^2) * P * R}{\beta^2 * P + R}, \beta = 1.5 \tag{6.3}$$

kde $N_{shift\&correct}$ reprezentuje počet správne identifikovaných dopytov do príslušných epizód, $N_{shift} + N_{contin}$ predstavuje súčet všetkých epizód identifikovaných modelom ako správne a N_{true_shift} predstavuje počet epizód identifikovaných ľudskými anotátormi. Uvedené miery sme rozšírili ešte o presnosť (angl. *Accuracy*) definovanú ako

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.4}$$

pričom určuje celkovú úspešnosť v identifikácii vyhľadávacích epizód, pretože uvažuje nielen správne zaradené dopyty do vyhľadávacích epizód, ale aj správne nezaradené dopyty. Na základe miery *Precision* určujeme presnosť metódy zisteným, koľko z identifikovaných epizód je identifikovaných správne a miera *Recall* nám prezrádza, koľko zo všetkých epizód sme identifikovali.

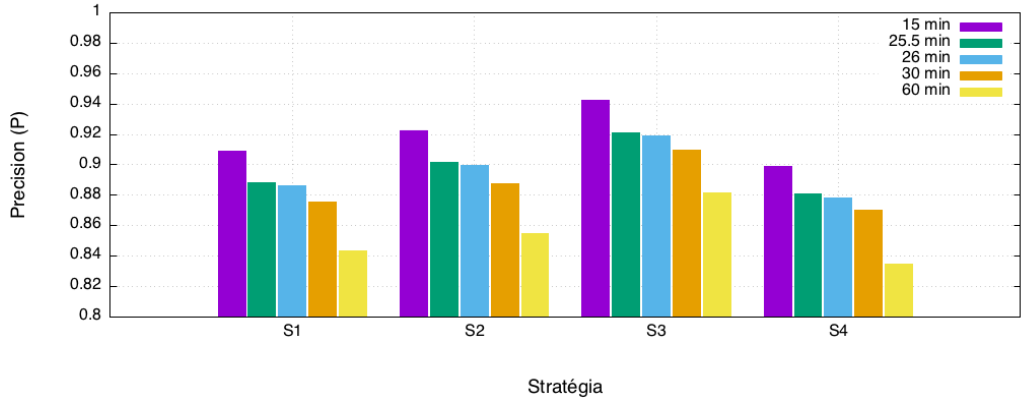
Ako prvý krok nášho experimentu sme si definovali stratégie, ktorými filtrujeme dáta. Naša používateľská štúdia odhalila, že približne len polovica z ohodnotených vyhľadávacích epizód obsahuje spoločné výsledky. Na základe toho sme si stanovili cieľ porovnať výkonnosť našej metódy na všetkých vyhľadávacích epizódach oproti dátam, ktoré obsahujú aktivitu používateľov a zdieľané výsledky. Definované stratégie filtrovali epizódy na základe toho, či obsahovali

- viac ako jeden dopyt ($S1$ – 233 epizód, 729 dopytov),
- viac ako jeden dopyt a aspoň jedno kliknutie na výsledok ($S2$ – 210 epizód, 673 dopytov),
- viac ako jeden dopyt a aspoň dve kliknutia ($S3$ – 173 epizód, 588 dopytov).

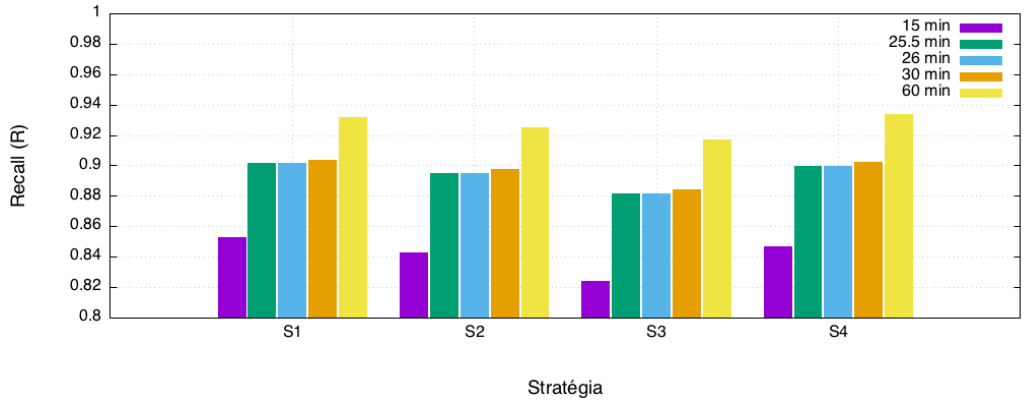
Medzi uvedené stratégie sme zaradili aj filtrovanie epizód od používateľov, ktorí nemali viac ako tri vyhľadávacie epizódy ($S4$ – 225 epizód, 706 dopytov), aby sme zistili, ako výrazne vplývajú vágne dopyty a preklepy na náš model. Pre každú

stratégiu sme rozdelili dáta na trénovaciu a testovaciu množinu v pomere 75%. Pri overení konkrétneho testovacieho dopytu sme porovnali daný dopyt s poslednými piatimi epizódami od daného dopytu a vyhodnotili jeho zaradenie.

V rámci prvého kroku overenia sme sa zamerali na otestovanie existujúcich riešení pre identifikáciu vyhľadávacích epizód. Najjednoduchším a zároveň najrozšírenejším prístupom je identifikácia vyhľadávacích epizód podľa času [50, 21]. Časovú segmentáciu sme realizovali vo viacerých intervaloch. Uvedené intervaly sme rozdelili na 15, 30, 25.5, 26 a 60 minút [21, 32, 7]. Na obrázkoch 6.5, 6.6, 6.7 a 6.8 je možné vidieť výsledky pre konkrétne časové intervaly a stratégie výberu vyhľadávacích epizód.

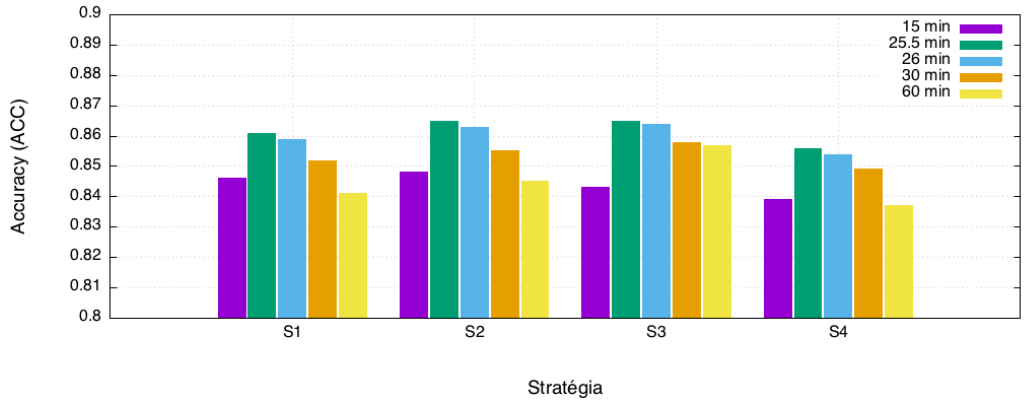


Obr. 6.5: Výsledky pre mieru Precision pre konkrétne stratégie a časové okná na mierke 0.8 až 1.0

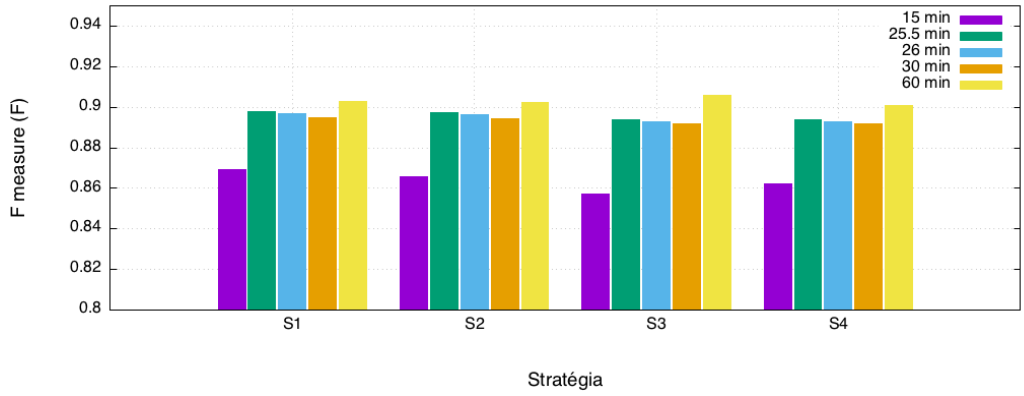


Obr. 6.6: Výsledky pre mieru Recall pre konkrétne stratégie a časové okná na mierke 0.8 až 1.0

Ako si môžeme všimnúť, presnosť jednotlivých intervalov od 25.5 minút až po 30 minút sa líši len veľmi málo, čo potvrdzuje tvrdenie autorov z [21], že na veľkosti intervalu v rámci jednej hodiny nezáleží a výsledky štandardu 25.5 minút nie sú o nič odlišnejšie od náhodne zvoleného ohraničenia. Významný je aj pomer *Precision* a *Recall* pre 15 minútový interval, pretože zatiaľ čo hodnota *Precision* je výrazne



Obr. 6.7: Výsledky pre mieru Accuracy pre konkrétne stratégie a časové okná na mierke 0.8 až 0.9



Obr. 6.8: Výsledky pre mieru F_β pre konkrétne stratégie a časové okná na mierke 0.8 až 0.95

vysoká, t.j. všetky epizódy, ktoré interval identifikoval ako správne sú správne, ale na základe hodnoty *Recall* vidíme, že viac ako 20% epizód prehliadol. Preto je čas úspešný výrazne v prípade krátkodobých záujmov ako nájdenie kina, podniku alebo konkrétnej ulice, ktoré väčšinou pozostávajú z dvoch alebo troch dopytov v rámci 5 až 15 minút.

Tabuľka 6.1 zhŕňa vývoj miery *Accuracy* pre konkrétne stratégie. Ako si môžeme všimnúť, čím sa zvyšuje interval pre komplexnejšie stratégie, čas zvyšuje svoju úspešnosť pri hľadaní komplexnejších epizód a zvyšuje tak hodnotu miery *Recall*. Čím viac ale zvyšuje interval, tým viac sa aj pri identifikácii pomýli a zhorší mieru *Precision* a *Accuracy*, pretože zaradil do vyhľadávacej epizódy dva dopyty, ktoré spolu nesúvisia.

V rámci ďalšieho kroku nášho overenia sme sa zamerali na porovnanie našej metódy (označenej ako *activity*) voči časovému atribútu (označenému ako *base*). Ako základný čas sme si zvolili interval 26 minút, pretože sa osvedčil ako najúspešnejší v prechádzajúcom kroku overenia. Na obrázkoch 6.9, 6.10, 6.11 a 6.12 je možné vidieť

Tabuľka 6.1: Tabuľka presností pre konkrétne intervaly a stratégie.

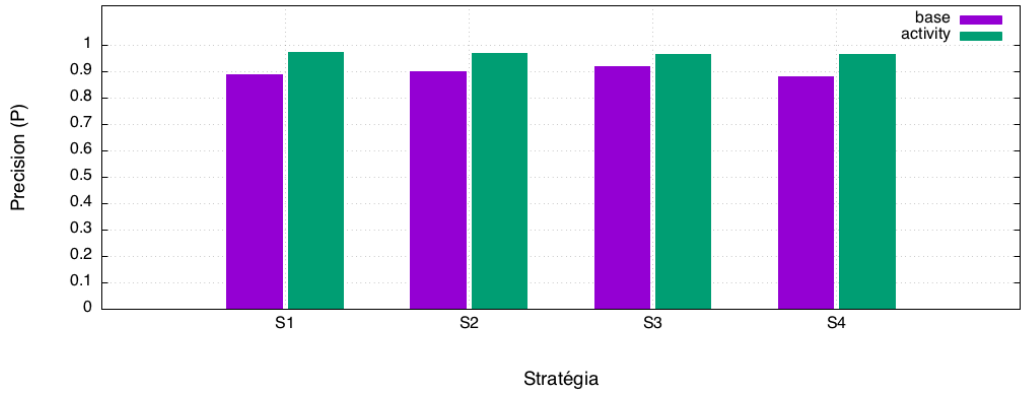
Stratégia	15 min	25.5 min	26 min	30 min	60 min
S1	0.846	0.861	0.859	0.852	0.841
S2	0.848	0.865	0.863	0.855	0.845
S3	0.843	0.865	0.864	0.858	0.857
S4	0.839	0.856	0.854	0.849	0.837

porovnanie jednotlivých mier pre našu metódu a časový atribút.

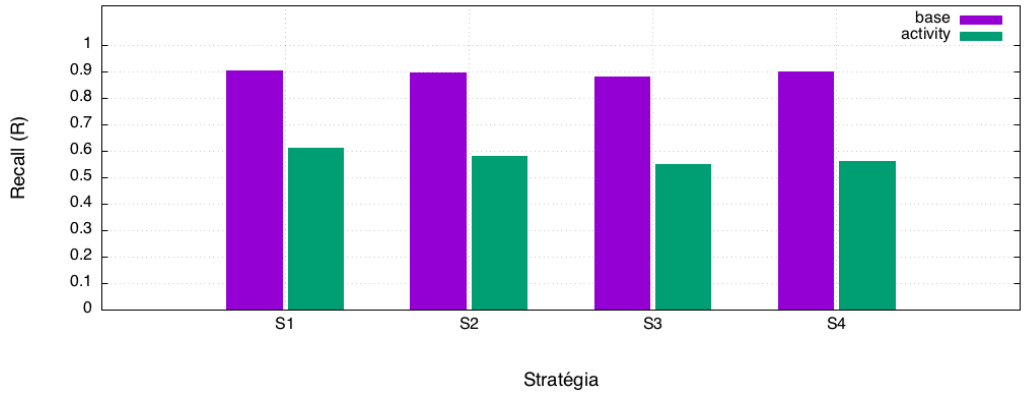
Tabuľka 6.2: Tabuľka miery Precision, Recall, Accuracy a F pre konkrétne intervaly a stratégie.

	base				*activity*			
Stratégia	P	R	ACC	$F_{1.5}$	P	R	ACC	$F_{1.5}$
S1	0.887	0.902	0.859	0.897	0.974	0.610	0.941	0.689
S2	0.899	0.895	0.863	0.896	0.970	0.579	0.936	0.661
S3	0.919	0.882	0.864	0.893	0.966	0.549	0.929	0.633
S4	0.879	0.899	0.854	0.893	0.965	0.561	0.934	0.644

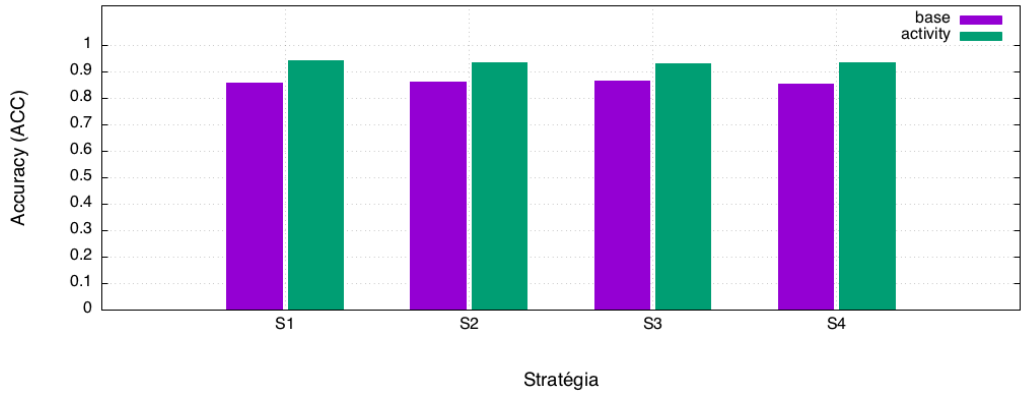
Miera *Accuracy*, t.j. presnosti našej metódy, je zobrazená na obrázku 6.11. Rozdiel úspešnosti našej metódy a časového atribútu je v priemere 8%. Tabuľka 6.2 porovnáva rozdiely v mierach medzi časovým atribútom a našou metódou. Významná miera, v ktorej dominuje časový atribút je *Recall*. Táto miera určuje, že z celkového počtu epizód časový atribút identifikoval viac epizód ako naša metóda. Preto je naša metóda presnejšia v identifikácii na základe miery *Precision*, ale dokázala identifikovať správne menej vyhľadávacích epizód. Vyššia hodnota miery *Recall* v prípade časového atribútu bola spôsobená hlavne charakterom nášho datasetu, pretože pozostáva, ako sme zistili aj v rámci používateľskej štúdie, prevažne z epizód s krátkodobým zámerom. Najvýznamnejšou mierou pre porovnanie našej metódy je miera *Accuracy*. Pomocou tieto miery určujeme porovnanie správne identifikovaných epizód (podobnosť aj rozličnosť dopytu) oproti všetkým – správnym a nesprávne identifikovaným. V prípade tejto miery je naše metóda úspešnejšia ako časový atribút približne o 8% vo všetkých stratégiach. Preto môžeme uvažovať, že využitie výsledkov vyhľadávania a aktivity používateľa dokáže zvýšiť presnosť identifikácie vyhľadávacích epizód. Keď sa zameriame na porovnanie úspešnosti našej metódy v rámci konkrétnych stratégii výberu dát, vidíme, že rozdiel úspešnosti je len minimálny, čo nasvedčuje tomu, že naša metóda je vďaka ostatným atribútom uvažujúcim čas a lexikálne vlastnosti úspešná aj pri dopytoch, ktoré nezdieľajú výsledky vyhľadávania alebo neobsahujú žiadnu aktivitu používateľa.



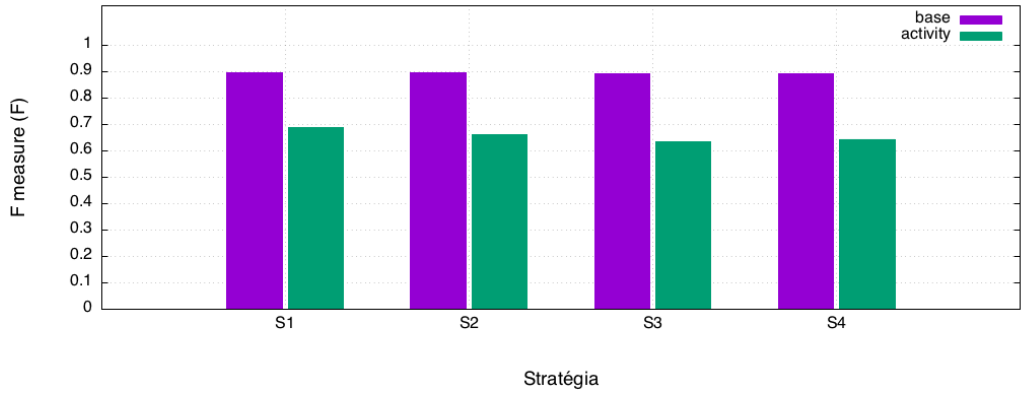
Obr. 6.9: Výsledky pre mieru Precision pre konkrétne stratégie v porovnaní s časovým atribútom a našou metódou



Obr. 6.10: Výsledky pre mieru Recall pre konkrétne stratégie v porovnaní s časovým atribútom a našou metódou



Obr. 6.11: Výsledky pre mieru Accuracy pre konkrétne stratégie v porovnaní s časovým atribútom a našou metódou



Obr. 6.12: Výsledky pre mieru F_β pre konkrétne stratégie v porovnaní s časovým atribútom a našou metódou

6.3 Diskusia

Naše overenie preukázalo, že podobné dopyty väčšinou zdieľajú rovnaké výsledky a aktivita používateľa spolu s lexikálnymi atribútmi a spoločnými výsledkami dokáže zvýšiť presnosť identifikácie vyhľadávacích epizód. Tento predpoklad sa nám podarilo overiť aj napriek nedostatku dát o aktivite používateľov približne na polovici našich vyhľadávacích epizód. Ako doplňujúce overenie by sme mohli porovnať úspešnosť našej metódy len na používateľoch, ktorí boli na základe ich dát naozaj aktívni a používali vyhľadávavač na skoro všetky ich vyhľadávania. Ak by sme uvažovali len týchto používateľov, podľa rozloženia počtu dopytov uvedeného z obrázku 6.1 by sme uvažovali vyhľadávacie epizódy len od prvých štyroch alebo piatich používateľov. Takto by sme mohli odfiltrovať náhodné dopyty alebo chyby používateľov a pravdepodobne ponúknuť model založený na relevantnejších dátach.

Priestor pre možné zvýšenie úspešnosti našej metódy leží vo využití negatívnej spätnej väzby. Pokiaľ používateľ klikol na výsledok na posledných miestach a nový dopyt obsahuje preskočené výsledky z prechádzajúceho dopytu ale nie kliknutý výsledok, môžeme predpokladať, že dopyty sa výrazne na seba podobajú, ale ich podobnosť nemá veľký význam pre konkrétneho používateľa. Náš vzorec podobnosti by sme tak rozšírili o negatívnu spätnú väzbu, pričom by sme na znevýhodnenie výsledku mohli využiť ešte jeho pozíciu v oboch dopytoch a zvýrazniť jeho nevýhodu úmerne jeho relevancii.

7 Štúdia kvality anotovania

Viacerí autori sa pri overovaní svojich prác zamerali na využitie anotátorov pre manuálne ohodnotenie náväznosti a podobnosti medzi dopytmi [38, 17, 48, 36, 32, 18, 47, 24]. Keďže anotátori sú väčšinou nezávislí a nepoznajú konkrétnych používateľov, nemajú tým pádom žiadnu informáciu o ich charaktere alebo záľubách a podobnosť medzi dopytmi hľadajú len na základe porovnávania slov a vlastnej intuície. Cieľom našej štúdie bolo overiť hypotézu, že manuálne anotovanie dopytov ľudskými anotátormi je nepresné a príliš subjektívne oproti anotáciam pridaným samotnými používateľmi, pretože nezávislí anotátori nedokážu nájsť skryté významy a následnosti medzi dopytmi už len z princípu, že nepoznajú používateľov a nevedia aký je ich vzťah k daným dopytom. Niekedy sa anotátori dokonca ani nemusia vyznať v doméne dopytov, čo môže mať za následok ešte viac zle zaradených dopytov.

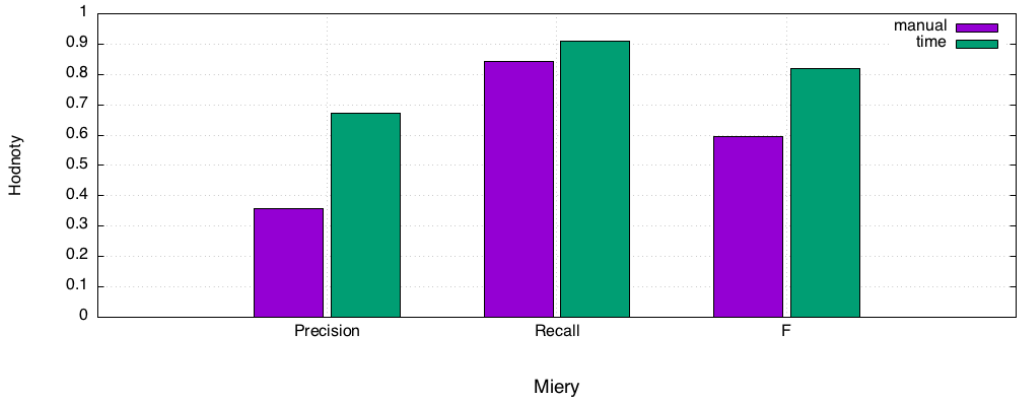
7.1 Vyhodnotenie presnosti manuálnej anotácie dopytov

Ako prvý krok vyhodnotenia sme vybrali piatich používateľov, ktorí mali najviac ohodnotených dopytov. Spolu sme tak získali 645 ohodnotených dopytov pre overenie. Tieto dopyty sme rozdelili podľa používateľov a troch nezávislých anotátárov sme požiadali o zaradenie dopytov do vyhľadávacích epizód priradením rovnakého čísla pri dopytoch, ktoré podľa nich patria do rovnakej vyhľadávacej epizódy. Toto zaradenie sme porovnali s anotáciami od používateľov. Ako mieru porovnania sme použili rovnaké miery ako v prípade kapitoly 6 – *Precision*, *Recall* a F_β na úrovni $\beta = 1.5$. Pri vyhodnotení daných mier sme postupovali nasledovne:

1. Pre uvedenú epizódu sme našli všetky používateľom ohodnotené dopyty.
2. Následne sme našli všetky dopyty z danej epizódy v zozname dopytov ohodnotenom anotátormi a číslo pri prvom dopyte epizódy sa stalo identifikátorom našej epizódy v zozname.
3. Pokiaľ dopyt zo zoznamu nemal pri sebe rovnaké číslo, určili sme ho ako *FN*.

Pokiaľ mal rovnaké číslo, určili sme ako *TP*. Ak rovnaké číslo bolo priradené pre dopyt, ktorý sa nenachádza v epizóde, určili sme ako *FP*.

Pre vyhodnotenie presnosti manuálnej anotácie (nazvanej ako *manual*) sme porovnali podobnosť daných dopytov aj pomocou časových atribútov (*time*) pri hodnote času 26 minút. Obrázok 7.1 zhŕňa porovnanie oboch metód. Ako si môžeme všimnúť, manuálne zaradenie dopytov pomocou anotátorov je horšie pre každú uvedenú mieru. Na základe toho môžeme usúdiť, že ako zlatý štandard je omnoho lepšie použiť podobnosť dopytov pomocou časových atribútov ako pomocou manuálnej anotácie.

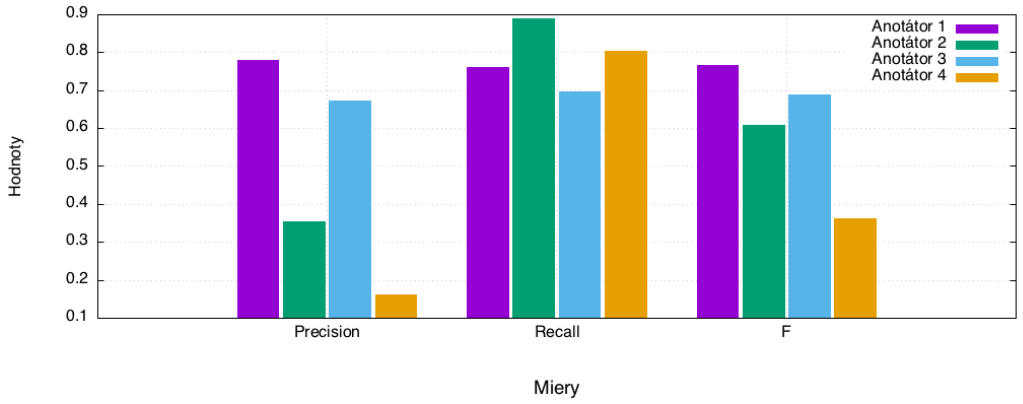


Obr. 7.1: Porovnanie manuálnej identifikácie epizód pomocou anotátorov a časových atribútov

Ďalej sme sa zamerali na identifikovanie miery zhody medzi štyrmi rôznymi anotátormi. Anotátorov sme požiadali o anotavanie celkovo 204 dopytov od jedného používateľa. Na obrázku 7.2 je možné vidieť porovnanie ich úspešnosti. Miera *Precision* sa výrazne odlišuje v prípade dvoch rôznych anotátorov a anotátori č. 2 a č. 4 sú výrazne horší ako ostatní. Anotované dopyty mali výrazne technický charakter, pričom anotátori č. 2 a č. 4 neboli v danej technickej doméne dostatočne zbehlí, a preto nedokázali identifikovať podobnosť väčšiny dopytov. Tento fakt potvrdzuje nepresnosť manuálnej anotácie externými anotátormi, pretože správna identifikácia výrazne záleží na doméne používateľa a anotátora. Ak sa anotátor nevyzná v doméne používateľa, prehliadne veľké množstvo podobných dopytov už len na základe toho, že obsahujú technické názvy, ktorým nerozumie.

Pre porovnanie zhody a vyhodnotenie chýb jednotlivých anotátorov sme sa rozhodli použiť Cohenov kappa koeficient [6] a Fleissov kappa koeficient, ktorý narozdiel od Cohenovho koeficientu dokáže spracovávať viacerých anotátorov [12]. Cohenov kappa koeficient je definovaný ako [6]

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{7.1}$$



Obr. 7.2: Porovnanie manuálnej identifikácie epizód pomocou štyroch anotátorov nad rovnakou sadou dopytov

kde $Pr(a)$ predstavuje relatívnu zhodu medzi dvomi anotátormi a $Pr(e)$ značí hypotetickú zhodu medzi anotátormi, keby každý anotátor zaradil dopyt do danej epizódy náhodne. Keďže Cohenov kappa koeficient je možné použiť len pre porovnanie dvoch anotátorov, ako mieru porovnania všetkých anotátorov sme zvoli zvolili Fleissov kappa koeficient definovaný ako [12]

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{7.2}$$

kde $\bar{P} - \bar{P}_e$ vyhodnocuje reálnu zhodu medzi anotátormi a $1 - \bar{P}_e$ vyhodnocuje možnú, t.j. náhodnú zhodu anotátorov. Fleissov kappa koeficient je nezávislý na počte anotátorov, kategórii a jednotlivých položiek, ktoré do kategórii zaraďujú anotátori. Obidva spomínané koeficienty môžu nadobúdať aj záporné hodnoty, avšak maximálna hodnota kappa koeficientov je 1.0. Tabuľka 7.1 zhŕňa interpretáciu kappa koeficientov podľa J. Landisa a G. Kocha [25]. Interpretácia hodnôt výrazne závisí od charakteru anotácii, počtu anotácii a aplikácie váhovania jednotlivých anotátorov (niektorí môžu byť spoľahlivejší a zdatnejší v doméne ako iní). V našej štúdii sme sa zamerali na porovnanie rozdielov anotátorov, a preto sme tých, ktorí sa vyznali v doméne dopytov používateľa nezvýhodňovali, aby sme zobrazili ich reálne odlišnosti v anotáciach.

Ako prvý krok sme porovnali dvoch anotátorov, ktorí sa vyznali v doméne dopytov oproti dvom anotátorom, z ktorých sa jeden v doméne nevyznal. Daných anotátorov sme porovnali pomocou Cohenovho kappa koeficientu. Prvá dvojica, ktorá sa vyznala v doméne dopytov, sa zhodla na úrovni $\kappa = 0.656$. Druhá dvojica, ktorá pozostávala z anotátora mimo domény, sa zhodla na úrovni $\kappa = 0.551$. Rozdiel zhody spolu s úspešnosťou naznačuje, že anotátor bez doménových znalostí dopytov sa viac pomýli pri dopytoch, pretože bez potrebných znalostí a súvislostí v danej doméne nedokáže správne identifikovať súvislosť medzi dopytmi. Opäť sme tak potvrdili náš

Tabuľka 7.1: Tabuľka interpretácie kappa koefiecientov podľa J. Landisa a G. Kocha [25].

Kappa koeficient	**Sila zhody**
< 0.00	Veľmi slabá zhoda
$0.00 - 0.20$	Slabá zhoda
$0.21 - 0.40$	Priemerná zhoda
$0.41 - 0.60$	Mierna zhoda
$0.61 - 0.80$	Značná zhoda
$0.81 - 1.00$	Skoro perfektná zhoda

predpoklad zobrazený aj v tabuľke 7.2, že kvalita anotácii výrazne závisí od znalostí domény. Pre overenie miery zhody všetkých anotátorov sme použili Fleissov kappa koeficient, pričom miera zhody medzi všetkými anotátormi dosiahla úroveň $\kappa = 0.518$. Anotátori boli tvorení z jednej polovice z expertov v doméne a druhá polovica sa v doméne nevyznala. Uvedená úroveň naznačuje, že anotátori sa zhodli len mierne. Pokiaľ si túto zhodu porovnáme s Cohenovým kappa koeficientom dvoch anotátorov, ktorí sa vyznajú v doméne, vidíme, že rozdiel v ich zhode je výrazný. Z našich zistení tak vyplýva, že pri výbere anotátorov pre konkrétny zoznam dopytov je potrebné komplexne analyzovať doménu dopytov, vybrať len anotátorov, ktorí sú najviac vzdelaní v danej doméne, a využiť ich vzájomnú zhodu spolu s časovým atribútom pre zvýšenie kvality zaradenia dopytov do epizód.

7.2 Diskusia

Dáta, ktoré sme poskytli anotátorom, obsahovali len dopyty bez ďalších iných metadát. Súčasťou reálnych dát pre anotátorov v praxi bývajú väčšinou aj dátumy, výsledky vyhľadávania a iné metadáta. Pre potreby nášho overenia sme dané metadáta anotátorom neposkytli a podľa nášho úsudku, jediný významný atribút pre anotátora by mohol byť dátum a čas zadania dopytu, pretože pokiaľ sa anotátor nevyzná v doméne, nebude rozumieť ani obsahu výsledkov a ručné prezeranie všetkých výsledkov pri počte dopytov v analyzovaných dátach v praxi nie je možné. Preto uvažujeme, že v prípade použitia všetkých uvedených dát by sa rozdiel v úspešnosti manuálnej anotácie výrazne nezmenil, len by sa priblížil úspešnosti časových atribútov.

Naše overenie taktiež ukázalo, že manuálna anotácia nie je vhodná ako zlatý štandard pre overenie vyhľadávacích epizód a je dokonca horšia ako časové atribúty, pretože anotátori sa častokrát mýlia, keď nepoznajú pravý význam dopytu pre používateľa alebo sa nevyznajú v doméne dopytov. Taktiež sme zistili, že anotátori sa odlišujú aj v prístupe porovnávania relevancie dopytov, pretože niektorí sa sústre-

ďovali na krátkodobé podobnosti dopytov a iní hľadali podobnosti dopytov aj dlhšie do histórie. Overenie pomocou koeficientov kappa preukázalo, že anotátori musia byť vyberaní pre anotovanie dopytov na základe ich znalosti domény konkrétnych dopytov, inak môžu prehliadnuť súvislosti v dopytoch len na základe toho, že nepoznajú doménové názvy použité v dopytoch a výsledkom ich anotácie sú tak nesprávne zaradené dopyty.

Výsledky nášho overenia nastolujú obavu, či manuálna anotácia dopytov externými anotátormi je tým správnym prístupom pre tvorbu zlatého štandardu pre overovanie metód identifikácie vyhľadávacích epizód. Pre potvrdenie našich predpokladov je potrebné realizovať komplexnejšie overenie s väčším počtom anotátorov a nastavením váh pre vyhľadávacie epizódy. Na základe toho by sme boli schopní ponúknuť lepší pohľad na rozdiely v hodnoteniach tak, že krátke epizódy pozostávajúce z jedného alebo dvoch dopytov by mali menšiu váhu oproti komplexnejším epizódam, pretože čím viac sa anotátor pomýli pri komplexnejšej epizóde, tým dokáže nájsť menšiu následnosť v dopytoch a identifikovať komplexnejšie ciele používateľov. Pokiaľ by túto komplexnú následnosť anotátor nenašiel, ale overovaná metóda áno, interpretujeme výsledok metódy ako nesprávny a jednoduchšie metódy uvažujúce menej komplexné epizódy sa budú javiť vo výsledkoch overenia úspešnejšie ako uvedená metóda. Najideálnejším riešením je získanie zaradenia dopytov priamo od používateľov, ale v prípade viacerých datasetov pri ich veľkosti a anonymite tento prístup nie je možný. Čiastočným riešením, aj na základe nášho porovnania zhody anotárorov, je získanie viacerých manuálnych anotácií pre zoznam dopytov od viacerých anotátorov, ktorí sa vyznajú v danej doméne, a využiť ich vzájomnú zhodu pre čiastočné spresnenie anotácii.

8 Záver

V rámci našej práce sme sa zamerali na využitie výsledkov vyhľadávania a aktivity používateľa počas vyhľadávania na zlepšenie identifikácie vyhľadávacích epizód. Náš prístup uvažuje mieru zdieľania výsledkov a čas strávený prezeraním konkrétnych výsledkov ako hlavný atribút podobnosti dopytov. Taktiež uvažujeme pomer pozícií zdieľaných výsledkov medzi dopytmi ako mieru posunu výsledku medzi dvomi dopytmi, pričom posun medzi relevantnejšie výsledky uvažujeme ako zvýšenie vzájomnej podobnosti dopytov. Prístup sme primárne navrhli ako model strojového učenia, a preto sme dopyty reprezentovali ako dvojice, ktorých atribúty určovali mieru ich podobnosti.

Náš prístup sme overili na dátach nami vytvoreného vyhľadávača Søke. Používatelia nášho vyhľadávača explicitne zaraďovali svoje dopyty do epizód pri zadaní nového dopytu. Na základe toho sme získali pravý zlatý štandard dát, ktorý vyjadruje presné podobnosti medzi dopytmi ako ich chápu používatelia. Na získaných dátach sme overili viaceré hodnoty časových atribútov, ktoré sa používajú ako základná metóda pre porovnanie úspešnosti nových metód pre identifikáciu vyhľadávacích epizód. Z uvedeného overenia sme identifikovali 26 minútový interval ako najúspešnejší časový atribút. Úspešnosť našej metódy sme porovnali s daným časovým intervalom a na základe miery *Accuracy*, ktorá určuje celkovú úspešnosť identifikácie, sme zistili, že naša metóda je v priemere o 8% lepšia ako zvolený časový interval.

V druhej časti overenia sme sa zamerali na štúdiu anotovania dopytov. Cieľom našej štúdie bolo overiť na dátach z vyhľadávača Søke hypotézu, že manuálna anotácia pomocou externých anotátorov nad dopytmi, ktoré daní anotátori nevytvárali, je menej presná ako anotácie od samotných používateľov vyhľadávača. Vybrané dopyty sme dali ohodnotiť nezávislým anotátorom, z ktorých sa niektorí vyznali v doméne daných dopytov a niektorí nie. Naše štatistické zistenia preukázali, že manuálna anotácia je dokonca horšia ako časový atribút s dĺžkou 26 minút, a preto je lepšie použiť časový atribút ako zlatý štandard pre porovnanie presnosti identifikácie vyhľadávacích epizód. Taktiež sme zistili, že podľa Fleissovho kappa koeficientu existuje len mierna zhoda medzi anotátormi, ktorí majú rôznu kvalitu znalostí domény dopytov. Pomocou Cohenovho kappa koeficientu sme porovnali rozdiel v zhode medzi anotátormi, ktorí sa vyznajú v doméne a ktorí nie. Na základe uvedeného koefi-

cientu sme opäť potvrdili, že anotátori, ktorí sa nevyznajú v doméne, majú menšiu úspešnosť pri identifikácii vyhľadávacích epizód, a preto pri manuálnej identifikácii pomocou anotátorov úspešnosť zaradenia dopytu do správnej epizódy výrazne závisí od doménových znalostí anotátora.

V ďalšej práci by sme sa radi zamerali na vylepšenie metódy uvažovaním negatívnej spätnej väzby, kedy výsledky, ktoré sú nezaujímavé pre používateľa, ale sú stále relevantné alebo na vyšších pozíciách pre nový dopyt uvažujeme ako menej relevantné pri výpočte podobnosti dopytov. Overenie našej metódy by sme radi realizovali na väčšej vzorke vyhľadávaní, ktoré by obsahovali aj dlhšie a komplexnejšie vyhľadávacie epizódy, pretože ako sme aj zistili v rámci používateľskej štúdie nášho vyhľadávača, medián dĺžky vyhľadávacích epizód predstavoval 2.28 minúty. Našu štúdiu anotovania by sme radi rozšírili o väčší počet anotátorov a porovnali odlišné domény dopytov anotované rôznymi anotátormi a overili tak na štatisticky signifikantnejšej vzorke dát presnosť manuálnej anotácie dopytov.

Literatúra

[1] Alpaydın, E.: *Introduction to Machine Learning Second Edition*. 2010, ISBN 9780262012430, 517 s.

[2] Baeza-Yates, R.; Ribeiro-Neto, B.: *Modern information retrieval*, ročník 9. ACM press, 1999, ISBN 020139829X, 513 s., 9780201398298.

[3] Billerbeck, B.; Demartini, G.; Firan, C.; aj.: Exploiting click-through data for entity retrieval. *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, 2010: s. 803 – 804.

[4] Brin, S.; Page, L.: The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems (1998)*, ročník 54, č. 1999-66, 1998: s. 1–17.

[5] Broder, A.: A taxonomy of web search. *ACM SIGIR Forum*, ročník 36, č. 2, September 2002: str. 3, ISSN 01635840.

[6] Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics*, , č. 2, 1996.

[7] Catledge, L.; Pitkow, J.: Characterizing browsing strategies in the World-Wide Web. *Computer Networks and ISDN systems*, 1995: s. 1065—-1073.

[8] Chapelle, O.; Sun, G.; Tech, G.: Global Ranking by Exploiting User Clicks. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009: s. 35–42.

[9] Chen, E.: Context-Aware Query Suggestion by Mining Click-Through and Session Data. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08 (2008)*, 2008: s. 875–883.

[10] Cui, H.; Wen, J.-R.; Nie, J.-Y.; aj.: Probabilistic query expansion using query logs. *Proceedings of the eleventh international conference on World Wide Web - WWW '02*, 2002: s. 325–332.

[11] Daoud, M.; Tamine-lechani, L.: Learning user interests for a session-based personalized search. 2008: s. 57–64.

[12] Fleiss, J. L.: Measuring nominal scale agreement among many raters. *Psychological Bulletin*, ročník 76, č. 5: s. 378–382.

[13] Friedman, J.: Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, ročník 29, 2001: s. 1189–1536.

[14] Gayo-Avello, D.: A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences*, ročník 179, č. 12, Máj 2009: s. 1822–1843, ISSN 00200255.

[15] Guo, F.; Li, L.; Faloutsos, C.: Tailoring click models to user goals. *Proceedings of the 2009 workshop on Web Search Click Data - WSCD '09*, 2009: s. 88–92.

[16] Hassan, A.; Jones, R.; Klinkner, K.: Beyond DCG: User behavior as a predictor of a successful search. *Proceedings of the third ACM international conference on Web search and data mining (2010)*, 2010: s. 221–230.

[17] Hua, W.; Song, Y.; Wang, H.; aj.: Identifying users' topical tasks in web search. *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13*, 2013: str. 93.

[18] Jiang, J.; He, D.; Allan, J.: Searching, Browsing, and Clicking in a Search Session: Changes in User Behavior by Task and Over Time. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14 (2014)*, 2014: s. 607–616.

[19] Joachims, T.: Training linear SVMs in linear time. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, 2006: str. 217.

[20] Joachims, T.; Granka, L.; Pan, B.; aj.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Transactions on Information Systems*, ročník 25, č. 2, Apríl 2007: s. 7–es, ISSN 10468188.

[21] Jones, R.; Klinkner, K. L.: Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, ACM, 2008, ISBN 978-1-59593-991-3, s. 699–708.

[22] Kotov, A.; Bennett, P.; White, R.: Modeling and analysis of cross-session search tasks. *Significance (2011)*, 2011: s. 5–14.

[23] Kramár, T.: *Utilizing Lightweight Semantics for Search Context Acquisition in Personalized Search*. 2013.

[24] Kramár, T.; Bieliková, M.: Detecting search sessions using document metadata and implicit feedback. *WSCD 2012 Workshop on Web Search Click Data. Held in conjuncion with WSDM 2012 (2012)*, 2012.

[25] Landis, J. R.; Koch, G. G.: The Measurement of Observer Agreement for Categorical Data Data for Categorical of Observer Agreement The Measurement. *Biometrics*, ročník 33, č. 1, 1977: s. 159–174.

[26] Levenshtein, V. I.: Binary codes capable of correcting deletions insertions and reversals. *Soviet Physics. 10*, 1966: s. 707–710.

[27] Li, Y.; Hsu, B.-J. P.; Zhai, C.; aj.: Unsupervised query segmentation using clickthrough for information retrieval. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*, 2011: s. 285—-294.

[28] Liao, Z.; Song, Y.; He, L.-w.; aj.: Evaluating the effectiveness of search task trails. *Proceedings of the 21st international conference on World Wide Web - WWW '12*, 2012: str. 489.

[29] Liu, C.; Belkin, N.; Cole, M.: Personalization of search results using interaction behaviors in search sessions. *Proceedings of the 35th international ACM*, 2012: s. 205–214.

[30] Liu, C.; Liu, J.; Belkin, N. J.: Predicting Search Task Difficulty at Different Search Stages. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14*, 2014: s. 569–578.

[31] Liu, J.; Liu, C.; Cole, M.; aj.: Exploring and predicting search task difficulty. *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*, 2012: str. 1313.

[32] Lucchese, C.; Orlando, S.; Perego, R.; aj.: Identifying Task-based Sessions in Search Engine Query Logs. *Proceedings of the fourth ACM international conference on Web search and data mining - WSDM '11 (2011)*: s. 277–286.

[33] Lucchese, C.; Orlando, S.; Perego, R.; aj.: Discovering tasks from search engine query logs. *ACM Transactions on Information Systems*, ročník 31, č. 3, Júl 2013: s. 1–43, ISSN 10468188.

[34] Ma, Z.; Pant, G.; Sheng, O. R. L.: Interest-based personalized search. *ACM Transactions on Information Systems*, ročník 25, č. 1, Február 2007: str. 5, ISSN 10468188.

[35] Metzler, D.; Dumais, S.; Meek, C.: Similarity measures for short segments of text. *Advances in Information Retrieval*, ročník 4425, 2007: s. 16–27.

[36] Ozertem, U.; Chapelle, O.: Learning to Suggest: A Machine Learning Framework for Ranking Query Suggestions. *SIGIR '12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012: s. 25–34.

[37] Ozmutlu, H. C.; Çavdur, F.: Application of automatic topic identification on Excite Web search engine data logs. *Information Processing & Management*, ročník 41, č. 5, September 2005: s. 1243–1262, ISSN 03064573.

[38] Paranjpe, D.: Learning Document Aboutness from Implicit User Feedback and Document Structure. *Proceeding CIKM '09 Proceedings of the 18th ACM conference on Information and knowledge management*, 2009: s. 365–374.

[39] Punera, K.: The Anatomy of a Click: Modeling User Behavior on Web Information Systems Categories and Subject Descriptors. *CIKM '10*, 2010: s. 989–998.

[40] Radlinski, F.; Joachims, T.: Query chains: learning to rank from implicit feedback. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005: s. 239–248.

[41] Richardson, M.; Dominowska, E.; Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. *Proceedings of the 16th international conference on World Wide Web*, 2007: s. 521–529.

[42] Shen, X.; Tan, B.; Zhai, C.: Implicit user modeling for personalized search. *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*, 2005: str. 824.

[43] Shokouhi, M.: Learning to Personalize Query Auto-Completion. 2011.

[44] Silverstein, C.; Marais, H.; Henzinger, M.; aj.: Analysis of a very large web search engine query log. *ACM SIGIR Forum*, 1999.

[45] Silvestri, F.: Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, ročník 4, č. 1 - 2, 2010: s. 1–174.

[46] Solskinnsbakk, G.; Gulla, J. A.: Contextual search navigation using semantic tag signatures. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies KNOW '11*, New York, New York, USA: ACM Press, September 2011, ISBN 9781450307321, str. 1.

[47] Sun, S.; Li, S.; Yang, M.; aj.: Utilizing variability of time and term content within and across users in session detection. *COLING (2008)*, , č. August, 2010: s. 1203–1210.

[48] Szpektor, I.; Gionis, A.; Maarek, Y.: Improving recommendation for long-tail queries via templates. *Proceedings of the 20th international conference on World wide web - WWW '11*, 2011: str. 47.

[49] Wang, H.; Song, Y.; Chang, M.; aj.: Learning to extract cross-session search tasks. *Proceedings of the 22nd international conference on World Wide Web*, 2013: s. 1353–1363.

[50] White, R. W.; Bennett, P. N.; Dumais, S. T.: Predicting Short-Term Interests Using Activity-Based Search Context. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10 (2010)*, , č. III, 2010: str. 1009.

[51] Xiang, B.; Jiang, D.; Pei, J.; aj.: Context-aware ranking in web search. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010: s. 451–458.