Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Mladen Karan

RAČUNALOM POTPOMOGNUTA IZGRADNJA I SEMANTIČKO PRETRAŽIVANJE ZBIRKI PITANJA I ODGOVORA

DOKTORSKI RAD

Zagreb, 2017.

Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Mladen Karan

RAČUNALOM POTPOMOGNUTA IZGRADNJA I SEMANTIČKO PRETRAŽIVANJE ZBIRKI PITANJA I ODGOVORA

DOKTORSKI RAD

Mentor: izv. prof. dr. sc. Jan Šnajder

Zagreb, 2017.

University of Zagreb
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Mladen Karan

COMPUTER-AIDED CONSTRUCTION AND SEMANTIC SEARCH OF QUESTION AND ANSWER COLLECTIONS

DOCTORAL THESIS

Supervisor: Associate Professor Jan Šnajder, PhD

Zagreb, 2017

Doktorski rad izrađen je na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva, na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave, u Laboratoriju za analizu teksta i inženjerstvo znanja (TakeLab).

Mentor: izv. prof. dr. sc. Jan Šnajder

Doktorski rad ima: 141 stranica

Doktorski rad br.: _____

O mentoru

Jan Šnajder diplomirao je, magistrirao i doktorirao u polju računarstva na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER), 2002., 2006. odnosno 2010. godine. Od 2002. godine radio je kao znanstveni novak, od 2011. godine kao docent, a od 2016. godine kao izvanredni profesor na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave FER- a. Usavršavao se na Institutu za računalnu lingvistiku Sveučilišta u Heidelbergu, Institutu za obradu prirodnog jezika Sveučilišta u Stuttgartu, Nacionalnome institutu za informacijske i komunikacijske tehnologije u Kyotu te Sveučilištu u Melbourneu. Sudjelovao je na nizu znanstvenih i stručnih projekata iz područja obrade prirodnog jezika i strojnog učenja. Voditelj je uspostavnog projekta HRZZ-a i projekta provjere koncepta HAMAG-BICRO-a te je istraživač na projektu UKF-a. Autor je ili suautor više od 100 znanstvenih radova u časopisima i zbornicima međunarodnih konferencija u području obrade prirodnog jezika i pretraživanja informacija te je bio recenzentom za veći broj časopisa i konferencija iz tog područja. Nositelj je šest predmeta na FER-u te je bio mentorom ili sumentorom studentima na više od 100 preddiplomskih i diplomskih radova. Član je stručnih udruga IEEE, ACM, ACL, tajnik Hrvatskoga društva za jezične tehnologije te suosnivač i tajnik posebne interesne skupine za obradu prirodnog jezika za slavenske jezike pri udruzi za računalnu lingvistiku (ACL SIGSLAV). Član je Znanstvenog centra izvrsnosti za znanost o podacima i kooperativne sustave te je pridruženi urednik časopisa Journal of Computing and Information Technology (CIT). Dobitnik je Srebrne plakete "Josip Lončar" 2010. godine, stipendije Hrvatske zaklade za znanost 2012. godine, stipendije Japanskog društva za promicanje znanosti 2014. godine te stipendije australske vlade Endeavour 2015. godine.

About the Supervisor

Jan Šnajder has received his BSc, MSc, and PhD degrees in Computer Science from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 2002, 2006, and 2010, respectively. From September 2002 he was working as a research assistant, from 2011 as Assistant Professor, and from 2016 as Associate Professor at the Department of Electronics, Microelectronics, Computer and Intelligent Systems at FER. He was a visiting researcher at the Institute for Computational Linguistics at the University of Heidelberg, the Institute for Natural Language Processing at the University of Stuttgart, the National Institute of Information and Communications Technology in Kyoto, and the University of Melbourne. He participated in a number of research and industry projects in the field of natural language processing and machine learning. He is the principal investigator on a HRZZ installation grant project and a HAMAG-BICRO proof-of-concept project, and a researcher on a UKF project. He has (co-) authored more than 100 papers in journals and conferences in natural language processing and information retrieval, and has been reviewing for major journals and conferences in the field. He is the lecturer in charge for six courses at FER and has supervised and co-supervised more than 100 BA and MA theses. He is a member of IEEE, ACM, ACL, the secretary of the Croatian Language Technologies Society, the co-founder and secretary of the Special Interest Group for Slavic NLP of the Association for Computational Linguistics (ACL SIGSLAV). He is a member of the Centre of Research Excellence for Data Science and Advanced Cooperative Systems and the associate editor of the Journal of Computing and Information Technology. He has been awarded the Silver Plaque "Josip Lončar" in 2010, the Croatian Science Foundation fellowship in 2012, the fellowship of the Japanese Society for the Promotion of Science in 2014, and the Endeavour Fellowship of the Australian Government in 2015.

Zahvala

Janu, koji je sa mnom prolazio sve. Koji je žrtvovao i svoje privatno vrijeme, nerijetko vikendom ili tijekom blagdana, kako bi meni pomagao. Koji je strpljivo odgovarao na moja mnogobrojna pitanja, koja nisu uvijek bila previše bistra. Janu, koji je vjerovao u mene kad ni ja sam nisam, s kojim sam rastao i još uvijek rastem kao istraživač i kao osoba.

Baki Mariji, koja mi je još od djetinjstva uvijek pružala veliku potporu, a to je nastavila činiti i tijekom poslijediplomskog studija. Baki, koja je neobično dobro upoznata s procedurom stjecanja doktorata, te je pomno i aktivno pratila svaki korak. Baki, koja me svojim smislom za humor, vitalnošću i optimističnim stavom prema životu svaki dan iznova iznenadi i inspirira.

Roditeljima, posebno majci, koja mi je kao profesor hrvatskog uspjela prenijeti svoju ljubav prema jezicima, što je neizravno utjecalo na odabir područja kojim se ovaj rad bavi. Mami, koja me naučila da je mašta jednako važna u životu kao i znanje. Mami bujne mašte, od koje svaki put mogu čuti neku novu priču uz kekse i sok.

Domagoju, drugu i suborcu na polju doktorskog studija. Čudnovatom čovjeku velikih mišića i velikog srca. Čovjeku koji je nebrojeno puta podmetnuo leđa i odradio razne zadatke umjesto mene, kako bih ja imao više vremena za doktorat. Domagoju, jedinom članu TakeLaba koji bi ovo znao prevesti na japanski!

Goranu, romantičnoj duši i najpragmatičnijem čovjeku kojeg znam. Goranu, koji je svojevremeno odradio mnogi projekt i fakultetsku obavezu kako bi mene rasteretio, a i sada se vrlo brzo javlja na chat kad ga zazovem. Forza Fiume!

Martinu, pustolovu koji je uvijek bio maksimalno raspoložen za kavu, rasprave o pitanjima iz dubokog učenja i odrađivanje kojeg termina labosa više.

Profesorici Dalbelo, koja mi je omogućila da postanem (i ostanem :) član TakeLaba i kasnije asistent. Koja drži labos na okupu sve ove godine. Koja mi je, unatoč vlastitim životnim izazovima, bila stalna potpora od samog početka.

Bubaču, Điju, Perkiju, Tomiju i Vuci – prijateljima koji su pokušali od mene napraviti socijalno biće ili barem spriječiti da postanem skroz asocijalan. Do neke mjere im je to uspjelo te su mi i više nego jednom pomogli naći bolji put.

Svima ostalima koji su bili dio mog života u ovom intenzivnom razdoblju. Svima koji su trpjeli moje frustracije, promjene raspoloženja, razmišljanje na glas i beskrajno odgađanje drugih obaveza zbog doktorata. Svima koji su se sa mnom u dobrim trenucima iskreno veselili, a u onim drugima bili mi potpora.

Sažetak

Rad se bavi nizom zadataka definiranih nad zbirkama često postavljanih pitanja (engl. *Frequently Asked Question Collections* – FAQ-zbirke). Takve zbirke sačinjavaju dokumenti koji se sastoje od pitanja i odgovora na to pitanje. Ovakav način strukturiranja informacija često koriste veliki pružatelji usluga, kao što su telekomunikacijski operateri, banke, javna i državna uprava, internetske trgovine i sl. U praksi, ove su zbirke tipično izgrađene specifično za neku domenu te sadrže ograničen broj informacijskih potreba. Ova specifična svojstva nekad se mogu iskoristiti za poboljšanje razvijanih modela i postupaka koji djeluju nad FAQ-zbirkama.

Cilj istraživanja bio je razvoj rješenja niza zadataka koji su ključni za uspješno korištenje FAQ-zbirki. Zadatci uključuju sve važnije poglede upravljanja FAQ-zbirkom, od njene izgradnje i održavanja do semantičkog pretraživanja. Rješavanje ovih zadataka vrlo je složeno zbog kratkoće tekstova i višeznačnosti prirodnog jezika zbog kojih se javlja "leksički jaz" između korisničkih upita i dokumenata. Pri istraživanju poseban je naglasak bio na razmatranju strojno potpomognutih postupaka koji bi što više smanjili količinu ljudskog rada potrebnu za upravljanje FAQ-zbirkom.

Za provođenje istraživanja izgrađena su tri skupa podataka na engleskom jeziku, te je korišten od prije izgrađen četvrti skup podataka na hrvatskom jeziku. Provedeno je predistraživanje na skupu podataka za hrvatski jezik u kojem je pokazano da su neki od predloženih postupaka semantičkog pretraživanja FAQ-zbirke dovoljno jezično neovisni za primjenu na proizvoljan jezik. Za engleski jezik provedena su dva predistraživanja kojima je potvrđeno da su skupovi reprezentativni za uvjete kakvi se javljaju u praktičnim primjenama te da su prikladni za daljnje istraživanje, koje je provedeno samo za engleski jezik.

Prvi istražen zadatak jest strojno potpomognuta izgradnja FAQ-zbirke. Potrebno je, uz skup korisničkih upita i dokumentaciju o postojećim proizvodima i uslugama, izgraditi FAQ-zbirku koja će biti namijenjena zadovoljavanju najčešćih informacijskih potreba korisnika. Izgradnja se provodi u dva koraka. Prvo se korisnički upiti grupiraju u grupe takve da upiti pojedine grupe adresiraju istu informacijsku potrebu. Za ovo je predložen postupak grupiranja s ograničenjima temeljen na aktivnom učenju. Nakon toga dohvaćaju se potencijalno relevantni tekstovi iz dokumentacije za svaku od informacijskih potreba pronađenih u prvom koraku. Dohvat je ostvaren pomoću klasičnih postupaka za pretraživanje informacija. Navedeni koraci olakšavaju i ubrzavaju izgradnju FAQ-zbirke jer automatiziraju značajan dio posla.

Drugi zadatak jest postupak za otkrivanje pitanja koja nisu pokrivena FAQ-zbirkom. Ovakva se pitanja pojavljuju kada se, nakon dužeg vremena korištenja, pojavi nova informacijska potreba korisnika koja nije prisutna u FAQ-zbirci. Predložen je postupak za otkrivanje nedostajućih pitanja temeljen na nadziranom strojnom učenju. Rješavanje ovog zadatka omogućava naknadnu nadopunu FAQ-zbirke potrebnim pitanjima i odgovorima te tako poboljšava njeno

pokrivanje informacijskih potreba korisnika.

Konačno, najvažniji doprinos rada jest niz modela za semantičko pretraživanje FAQ-zbirke. Predložene su dvije vrste modela, koji se temelje na nadziranome strojnom učenju rangiranja. Prva vrsta modela, uz same riječi u tekstu, koristi niz lingvistički motiviranih značajki kao što su oznake vrste riječi ili jezgrene funkcije nad sintaktičkim stablima. Druga vrsta modela temelji se na konvolucijskoj neuronskoj mreži, koja radi izravno sa semantičkim vektorskim reprezentacijama riječi. Pokazano je da obje vrste modela daju zadovoljavajuće rezultate uz podatke označene strategijom usmjerenom na parafraze. Iz toga proizlazi da predloženi modeli pružaju u praksi značajna poboljšanja točnosti pretraživanja u usporedbi s nenadziranim alternativama, ali uz razmjerno malen rad uložen u označavanje.

Ključne riječi: Često postavljana pitanja, odgovaranje na pitanja, pretraživanje informacija, grupiranje s ograničenjima, aktivno učenje, učenje rangiranja, konvolucijska neuronska mreža, semantičko pretraživanje.

Summary

Computer-Aided Construction and Semantic Search of Question and Answer Collections

This thesis focuses on several tasks concerning frequently asked questions (FAQ) collections. FAQ collections are composed of documents containing a question and a corresponding answer. This way of structuring information is often used by large-scale service providers, such as telecom-operators, banks, state-administration, online stores etc. The benefit of FAQ collections for large service-providers is threefold. First, they enable a majority of users to get answers to their questions quickly and accurately, thereby increasing the quality of user experience. Second, due to the users getting their answers more quickly, fewer users contact the customer service contact center, which decreases the number of user queries that need to be dealt with manually by customer service agents. Third, apart from being used directly by the users themselves, FAQ collections can also be used by the customer service agents. In such a scenario, agents could have access to an internal FAQ collection with questions asked by various users in the past. This enables the agents to reuse previously answered user queries, in order to formulate answers to new user queries more efficiently.

This thesis addresses a number of tasks important for the successful use of FAQ collections, ranging from constructing and managing FAQ collections to semantic search over FAQ collections. The challenge in solving these tasks lies in the shortness of texts and ambiguity of natural language, which give rise to a "lexical gap" between queries and documents. An additional goal of the thesis was to investigate machine-aided approaches that would minimize the amount of human effort required for FAQ collection management.

In practical applications, FAQ collections are often small and contain a limited number of unique information needs. Such properties were taken into account in this research, as they can be leveraged for constructing and searching FAQ collections.

To evaluate the effectiveness of the proposed methods a suitable FAQ retrieval dataset is necessary. In the context of this research, a FAQ retrieval dataset consists of three components. First, a FAQ collection with documents to be searched. Second, a set of user queries targeting those documents. Third, a set of relevance judgements that define a set of relevant documents for each query. Furthermore, to make the datasets as realistic as possible, several additional requirements were enforced when creating the datasets.

1. Documents should be in the form of FAQ pairs;
2. The dataset should be domain-specific, as typical FAQ collections in practical applications are focused on a very narrow domain;
3. The dataset should contain sufficient data for applying supervised machine learning met-

hods;

4. The dataset should have query-level redundancy, which is defined as each query having a number of its paraphrases. This last requirement could equally well pertain to other kinds of information retrieval datasets, as even in non FAQ-related applications, it will often be the case that the users will phrase the same question in different ways. However, FAQ collections are typically small enough to allow for manual annotation of query paraphrases. Annotated paraphrases can later be used to improve models for constructing and searching FAQ collections.

As none of the datasets available in the literature fully satisfied these requirements, with the exception of the VipFAQ dataset in Croatian, three new English datasets were built.

The first dataset – VerizonFAQ – was built by manually annotating queries and relevance judgements on FAQ pairs crawled from the web site of a large US Telecom company. The second dataset is FAQIR, built in a similar fashion, but using FAQ pairs obtained from the "maintenance & repairs" subcategory of the YahooAnswers community question answering web site. An interesting property of this dataset is the annotation scheme that was used. The relevance judgements for each query and FAQ pair are not binary (relevant or non-relevant), but rather on a scale of 1-4 with the following semantics:

1. *Relevant* – a FAQ pair is the adequate answer to the query;
2. *Useful* – while not exactly covering the information need of the query, the FAQ pair does provide useful information;
3. *Useless* – while covering the main topic of the query the FAQ pair offers no additional useful information to actually provide an answer to it;
4. *Non-relevant* – the FAQ pair is not relevant in any way for the query.

A third dataset – StackFAQ – was built on data obtained from the "Web Apps" category of the StackExchange web site. Unlike for the previous two datasets, the relevance judgements for this dataset were not annotated manually, but were derived automatically from users' up-votes.

Preliminary experiments were conducted on the Croatian VipFAQ dataset. A supervised machine learning approach was used, employing an SVM classifier on a wide range of textual similarity features. The model, which takes as input a pair consisting of a query and a FAQ pair, must classify the pair as *relevant* or *non-relevant*. To use this model for information retrieval, the candidate FAQ pairs for a given query are ranked by the confidence score of the SVM for the *relevant* class. The experimental evaluation demonstrated that the benefits of using retrieval methods based on supervised machine learning can be substantial for this task. Moreover, the experiments demonstrated that some of the proposed semantic search methods are sufficiently language-independent to be applied to any language.

Furthermore, two sets of preliminary experiments were conducted on the English datasets. These experiments ensured that the datasets are representative of real-world conditions that arise

in practical applications of FAQ collections and are thus suitable for the rest of this research. The first set of experiments was conducted on the VerizonFAQ dataset, and investigated the influence of manually and automatically derived query expansion rules on retrieval performance. Although the query expansion rules did not significantly affect the performance, the experiment revealed that the VerizonFAQ dataset is unrealistically simple. Consequently, any conclusions based on this dataset would be very unrealistic and therefore this dataset was not used in the rest of the research. The second set of experiments was conducted on the FAQIR dataset, and was meant to provide a baseline information retrieval performance for the more complex retrieval models. Several unsupervised retrieval models were used, including the BM25, a simple vector space retrieval model, and a model based on aggregating word embeddings. The experiments also included a small ensemble of the three models, which emerged as the best unsupervised approach. These experiments confirmed that the FAQIR dataset is realistic and suitable for use in the rest of this research.

After conducting these preliminary experiments the rest of the research was focused on addressing three most prominent tasks on FAQ collections:

1. Building the FAQ collection while minimizing human effort;
2. Detecting information needs missing in the FAQ collection;
3. Semantic search over the FAQ collection.

Proposed solutions of these tasks comprise the main contributions of this thesis.

The first task considered is machine-aided construction of a FAQ collection. The task is to, given a set of user queries and documentation about existing products and services, build a FAQ collection that addresses the most frequent information needs of the users. The construction consists of two steps. First, user queries are grouped in such a way that queries from a given group address the same information need. To this end, a constrained clustering approach based on active learning was proposed. The procedure starts from an unsupervised clustering and iteratively improves it by asking the users questions and modifying the clustering based on the answers it receives. In essence, the users' answers provide constraints that improve the quality of the clustering. The experiments show that even with a reasonably low number of questions posed to the users, the resulting clustering is considerably better than the initial, unsupervised one. After the clustering, each cluster represents a group of paraphrased queries for a particular information need. In the next step, potentially relevant answer text from the documentation is retrieved for each of the information needs identified in the first step. The retrieval is performed using standard information retrieval methods operating on clusters of queries instead of individual queries. This step saves time when looking up the answer for an information need. Combined, these two steps make the construction of the FAQ collection easier and faster. In the experimental evaluation, both steps proved to have practically satisfactory performance .

The second task explored in this research is a method for detecting questions that are not

covered by a FAQ collection. Such questions appear when, after a prolonged period of use, a new information need arises among the users, which is absent from the original FAQ collection. Several methods for detecting such cases were investigated. All proposed methods are based on the underlying idea that a representation should be derived for each information need in the FAQ collection. The representation of a new, potentially missing information need is compared to representations of existing information needs, and the new need is labeled as missing if the resulting similarity is too low. The investigated models define how to derive and compare information need representations. They include unsupervised methods based on word embeddings as well as supervised methods based on SVM and Gaussian Mixture Models. Special attention is given to exploiting the query paraphrases in the available datasets, which showed considerable performance boosts. Solving this task enables additional supplementation of a FAQ collection with missing questions and answers, thus increasing the coverage of information needs.

Finally, the most important contribution of this thesis are several models for semantic search on FAQ collections. Two types of models were proposed, both based on supervised learning to rank. The first type of models uses surface similarity features between the query and document texts. Along with word based features, several linguistically motivated features are also used. A comprehensive feature list is as follows:

1. Ranks that a document had for the given query in the baseline unsupervised retrieval models;
2. Bag-of-words features – the number of times a word appears in the document. Two numbers for each word: one for the query and one for the document. As this generates a considerable number of features, filter feature selection was performed using the χ^2 method;
3. Levenshtein distance between the query and a document;
4. Four features implemented in the SEMILAR package, based mostly on greedy word alignments paired with LSA or WordNet;
5. Syntax tree kernel similarity – kernel functions defined on syntax trees of texts define a similarity metric that takes into account the syntactic information in texts. Outputs of four kernel functions were used as features: subtree kernel, subset tree kernel, partial tree kernel, and semantically smoothed partial tree krenel. The last kernel type is especially interesting, as it combines the tree kernel paradigm with word embeddings. This kernel uses the embeddings to semantically compare words in the leaves of syntax trees in order to provide a higher quality overall similarity score.

ListNET and LambdaMART machine learning algorithms were used on these feature representations of query-document pairs.

The second type of model is based on a convolutional neural network. It is trained to classify each pair of a query and a document into class *relevant* if the document is relevant for

the query, or *non-relevant* otherwise. The network operates directly on semantic vector word representations of text. The architecture of the network is as follows:

1. Two convolutional layers each derive a latent representation of the query and the document (in this case a FAQ pair), respectively. The texts are represented as matrices containing word embeddings;
2. A max-pooling layer reduces the output of the convolutions to a fixed length representation corresponding to the number of convolution filters used;
3. Outputs of the previous layer are fed into a standard fully-connected hidden layer with ReLU transfer functions;
4. Outputs of the hidden layer are passed into a fully-connected output layer with the softmax transfer function and two outputs.

Learning is performed by minimizing the cross-entropy loss using standard stochastic gradient descent over minibatches. The ranking is produced by ordering the candidate documents for a given query by the model's confidence for the *relevant* class.

It was shown that both types of models give good results that outperform baseline unsupervised ranking models. Moreover, the performance benefits of supervised models exist even when they are trained on data labeled with a paraphrase-focused strategy. This implies that it is possible to exploit paraphrases of queries present in the data (whose manual annotation is feasible in the case of FAQ collections) to reduce the need for laborious document labeling. Consequently, the models provide practically relevant search performance improvements over unsupervised alternatives, while requiring little manual annotation effort.

Keywords: Frequently asked questions, Question answering, Information retrieval, Constrained clustering, Active learning, Learning to rank, Convolutional neural network, Semantic search.

Sadržaj

Poglavlje 1.

Uvod

Rad se bavi nizom zadataka definiranih nad zbirkama pitanja i odgovora, koje u kontekstu ovog istraživanja predstavljaju zbirke često postavljanih pitanja (engl. *Frequently Asked Question Collections* – FAQ-zbirke). Razmatrani zadatci uključuju izgradnju, održavanje i pretraživanje FAQ-zbirki. Motivacija za ovo istraživanje proizlazi iz toga što su FAQ-zbirke često korištene u praksi, pa bi bolja rješenja ovih zadataka poboljšala korisničko iskustvo velikog broja korisnika. Cilj ovog rada je istražiti moguća rješenja za spomenute zadatke uz poseban naglasak na strojno potpomognute pristupe, koji bi uključivali što manje ljudskog rada. U nastavku, opisano je šire područje rada, obrazložene su specifičnosti FAQ-zbirki i podrobnije su opisani zadatci kojima se ovo istraživanje bavi.

1.1. Šire područje rada

Područje ovog rada jest analiza i pretraživanje teksta. To se područje nalazi na presjecištu više potpodručja umjetne inteligencije:

- Strojno učenje (engl. *machine learning*) – Područje koje se bavi izradom računalnih modela koji iz označenih primjera mogu naučiti rješavati probleme za koje je inače potrebna ljudska inteligencija. Modeli i postupci temeljeni na strojnom učenju jezgra su većine suvremenih postupaka analize i pretraživanja teksta;
- Obrada prirodnog jezika (engl. *natural language processing*) – Područje koje se bavi razumijevanjem i generiranjem prirodnog jezika[1] pomoću računala. Ovo područje kombinira znanja iz lingvistike sa znanjima iz umjetne inteligencije (ponajviše strojnog učenja) kako bi se izgradili računalni modeli jezičnih pojava;
- Pretraživanje informacija (engl. *information retrieval*) – Ovo područje bavi se problemom dohvata elemenata koji su relevantni za neku informacijsku potrebu. Elementi mogu biti

[1] Jezik koji se prirodno razvio kroz dugogodišnju ljudsku komunikaciju bez svjesnog djelovanja da se potakne takav razvoj. Prirodni jezici po tome su oprečni formalnim jezicima, kakvi se koriste za, npr., programiranje računala.

prikazani različitim medijima, primjerice slike, tekst ili videozapisi. U okviru ovog rada razmatraju se samo postupci koji rade s tekstom. Najuspješnije tehnike iz ovog područja najčešće kombiniraju vektorske prikaze teksta izgrađene postupcima obrade prirodnog jezika s postupcima za rangiranje na temelju strojnog učenja.

Rješavanje zadataka analize i pretraživanja teksta pomoću računala vrlo je složeno jer su ti zadaci najčešće UI-potpuni.[2] Razlog tome jest velika složenost i visoka razina višeznačnosti prisutna u ljudskom jeziku. Nadalje, složenost i višeznačnost protežu se kroz sve razine jezika, od fonologije, preko morfologije, sintakse i semantike sve to razina pragmatike i diskursa. Razrješavanje višeznačnosti jezika je za ljude trivijalno te se događa automatski. No, za računalo ovo predstavlja značajan problem. Na primjer, u rečenici "Prešao sam zebru preko crvenog.", čovjeku će odmah biti jasno da riječ "zebra" u tom kontekstu označava pješački prijelaz, a ne afričku životinju. Za računalo je ovo izazovan zadatak te postoji grana obrade prirodnog jezika koja se njime bavi – razrješavanje višeznačnosti riječi (engl. *word sense disambiguation*). Ponekad jezik može biti toliko višeznačan da je i ljudima izazovno odgonetnuti stvarno značenje teksta. Primjerice, iz rečenice "Pozdravio sam prijatelja sa brda." nije potpuno jasno je li značenje: (1) "Ja sam na brdu te sam od tamo pozdravio prijatelja." ili (2) "Pozdravio sam prijatelja koji, za razliku od mojih ostalih prijatelja, živi na brdu." Dodatno na već opisane probleme, prirodni jezik često koristi znanje o svijetu koje računalo nema. Na primjer, rečenica "On trči kao Bolt.", zapravo znači da netko trči brzo. No, da bi je se ispravno razumjelo, treba znati na koga se odnosi riječ Bolt i na temelju toga zaključiti da se radi o komentaru brzine nečijeg trčanja. Zbog ovih problema analiza i pretraživanje teksta izazovno je područje, koje koristi postupke i modele iz sva tri prethodno nabrojena područja.

Ovaj rad primjenjuje postupke iz analize i pretraživanja teksta za rješavanje niza zadataka definiranih nad zbirkama često postavljanih pitanja i odgovora. Ovi zadatci kao i motivacija za njih detaljno su opisani u sljedećim odjeljcima.

1.2. FAQ-zbirke

Zbirke često postavljanih pitanja i odgovora (engl. *frequently asked questions* – FAQ) često se koriste za olakšavanje pristupa informacijama. Najčešći korisnici takvih zbirki su klijenti velikih tvrtki koja se bave uslužnim djelatnostima. Samo neki od brojnih primjera iz ove skupine su telekomunikacijski operateri, pružatelji komunalnih usluga (struja ili plin), tijela javne i državne uprave, banke, internetske trgovine i sl. Korisnost zbirki često postavljanih pitanja i odgovora dolazi do to većeg izražaja što je broj klijenata veći.

Zbirke često postavljanih pitanja i odgovora (u nastavku rada *FAQ-zbirke*) su tekstne zbirke

[2] engl. *AI complete* – zadaci koji se smatraju toliko teškima da bi njihovo rješavanje bilo ekvivalentno stvaranju *jake UI* – umjetne inteligencije koja je jednako inteligentna kao čovjek.

koje se sastoje od više *FAQ-parova*. Jedan FAQ-par uključuje (1) FAQ pitanje, koje izražava neku informacijsku potrebu i (2) FAQ odgovor, koji daje odgovor na informacijsku potrebu izraženu FAQ pitanjem. Prilikom izgradnje FAQ-zbirke posebna pažnja posvećuje se tome da FAQ-parovi pokrivaju informacijske potrebe koje se često javljaju u korisničkim upitima. Tako se postiže da FAQ zbirka koja nije prevelika zapravo pokriva značajan dio upita korisnika. Korisnost takve zbirke očituje se na tri načina:

1. Brži pristup informacijama – Ako se korisnički upit nalazi u FAQ-zbirci, korisnik može pretragom preko web-sučelja vrlo brzo i bez potrebe za kontaktiranjem korisničke službe dobiti prihvatljiv odgovor na svoje pitanje. To korisniku štedi vrijeme i tako povećava kvalitetu korisničkog iskustva;
2. Rasterećenje korisničke službe – Ovo je izravna posljedica prethodno navedene prednosti. Budući da značajan broj korisnika može pronaći prihvatljiv odgovor na svoj upit u FAQ-zbirci, smanjuje se broj korisnika koji će kontaktirati korisničku službu. Zbog smanjenog opterećenja povećava se kvaliteta usluge korisničke službe te se smanjuje vrijeme čekanja za one korisnike koji je ipak kontaktiraju;
3. Povećanje učinkovitosti korisničke službe – Osim za prikazivanje informacija korisnicima usluga, FAQ-zbirka može se koristiti i za prikazivanje informacija agentima korisničke službe. Ovakva FAQ-zbirka sadržavala bi pitanja koja su korisnici prije često postavljali korisničkoj službi. Agenti korisničke službe koji bi imali pristup takvoj zbirci mogli bi brže davati odgovore korisnicima te bi dani odgovori bili usklađeniji na razini svih agenata korisničke službe.

U praktičnim primjenama FAQ-zbirke su obično malene – reda veličine nekoliko stotina ili par tisuća FAQ-parova. Također, najčešće su domenski specifične, tj. usko usredotočene na konkretnu domenu iz koje će dolaziti korisnički upiti. Na takvim FAQ-zbirkama moguće je osmisliti specijalizirane postupke analize podataka koji iskorištavaju njihova specifična svojstva. Iz ovih razloga u ovom je radu naglasak upravo na malim, domenski specifičnim zbirkama.

1.3. Zadaci pretraživanja FAQ-zbirki

Prilikom korištenja FAQ-zbirki postoji više zadataka koje je potrebno riješiti. Za svaki od njih poželjno je smanjiti potreban ljudski posao kroz računalom potpomognutu automatizaciju. Za neke od zadataka to nije moguće, ali je moguće smanjiti ljudski trud potreban za postizanje zadovoljavajućih rješenja. Zadatci nad FAQ-zbirkama koji su od praktične važnosti su sljedeći:

1. Izgradnja FAQ-zbirke – Zadatak uključuje stvaranje nove FAQ-zbirke. Očito rješenje je zapošljavanje domenskog stručnjaka koji bi krenuo od prazne FAQ-zbirke te sastavio nove FAQ-parove koje bi smatrao prikladnima. Međutim, ovo rješenje je skupo i dugotrajno. Nadalje, domenski stručnjak može pogrešno procijeniti koje informacijske potrebe

je prikladno uključiti u zbirku. Cjeloviti postupak izgradnje FAQ-zbirke do sada još nije uspješno automatiziran. Ipak, jedna mogućnost ublažavanja navedenih nedostataka jest strojno potpomognuta poluautomatska izgradnja zbirke. Uz odgovarajuće preduvjete, domenski stručnjak može dobiti pomoć računala pri izgradnji zbirke. Ovakva vrsta pristupa jedna je od tema ovog rada, koja je detaljno razmotrena u poglavlju 7.;

2. Otkrivanje upita nepokrivenih zbirkom – Ovaj zadatak bavi se otkrivanjem informacijskih potreba za koje ne postoji odgovarajući FAQ-par u FAQ-zbirci. Prednost rješavanja ovog zadatka je to što bi ono omogućilo naknadno nadopunjavanje FAQ-zbirke prikladnim FAQ-parom, čime do većeg izražaja dolaze već navedene prednosti FAQ-zbirki. Ovaj zadatak moguće je zadovoljavajuće riješiti potpuno automatskim postupkom, što je također jedna od tema ovog rada istražena u poglavlju 8.;
3. Otkrivanje podvostručenih upita – Umjesto pronalaženja informacijskih potreba koje nedostaju, u ovom slučaju je zadatak pronaći informacijske potrebe za koje u FAQ-zbirci postoji više od jednog FAQ-para. Zadatak je do neke mjere sličan prethodnom. Rješavanje ovog zadatka omogućuje da se uklone duplikati iz FAQ-zbirke. Duplikati u osnovi ne predstavljaju ozbiljan problem u FAQ-zbirkama. No, oni ipak predstavljaju suvišne FAQ-parove koji usporavaju postupak pretraživanja. Dodatno, takvi FAQ-parovi mogu u nekim slučajevima negativno utjecati na rezultate pretraživanja zbirke;
4. Otkrivanje nepotrebnih upita – Zadatak je usko povezan sa prethodna dva zadatka. Potrebno je pronaći informacijske potrebe za koje u FAQ-zbirci postoji FAQ-par, ali one zapravo nisu česte. Takav FAQ-par je vrlo rijetko relevantan za korisnički upit. S druge strane, za sve ostale korisničke upite on može negativno utjecati na rezultate pretraživanja. Zbog toga je takav FAQ-par, nakon što ga se otkrije, najbolje ukloniti iz zbirke;
5. Pretraživanje FAQ-zbirki – Za korisnički upit, koji izražava korisnikovu informacijsku potrebu, potrebno je rangirati sve FAQ-parove iz zbirke tako da oni koji su najrelevantniji za korisnikovu informacijsku potrebu budu što bliže vrhu rangiranog popisa. Ovaj zadatak najvažniji je od svih navedenih zadataka. Ovo je složen zadatak zbog problema leksičkog jaza (v. odjeljak 9.1.). Niz postupaka za rješavanje ovog problema, specifično za male, domenski specifične FAQ-zbirke, razmatran je u sklopu ovog rada u poglavlju 9.

Vrijedi napomenuti da bi se zadatke 2.–4. moglo objediniti u jedan veliki zadatak – *održavanje FAQ-zbirke*. Takav zadatak sveo bi se na otkrivanje svih mogućih slučajeva kada je FAQ-zbirku potrebno ažurirati, što je slučaj koji se najčešće javlja kada prođe duže vrijeme od izgradnje FAQ-zbirke, pa se aktualno popularne informacijske potrebe promijene.

Svi spomenuti zadaci svode se na primjenu postupaka za računalnu analizu i pretraživanje teksta. Zbog već opisanih izazova u ovom području ovi zadaci su vrlo teški za riješiti na računalu. Najbolji rezultati u ovom području postignuti su primjenom nekog od postupaka nadziranog ili nenadziranog strojnog učenja. Zbog toga su modeli i postupci razmatrani u ovom

radu usmjereni na primjenu strojnog učenja.

Ovaj se rad bavi značajnijim zadatcima u korištenju FAQ-zbirke. Posebice, rad je usredotočen na izgradnju zbirke, otkrivanje nepokrivenih upita i pretraživanje zbirke. Pri rješavanju navedenih zadataka uzeta su u obzir specifična svojstva malih, domenski specifičnih FAQ-zbirki koja bi se mogla iskoristiti za poboljšanje učinkovitosti predloženih rješenja. Dok su predistraživanja provedena za hrvatski i engleski jezik, glavni dio istraživanja proveden je samo za engleski jezik. Tijekom cijelog istraživanja posebna je pažnja posvećena korištenju jezično neovisnih pristupa. Zato se rezultati istraživanja mogu, uz male prilagodbe, primijeniti na proizvoljan jezik. Izvorni znanstveni doprinosi ovog rada su kako slijedi.

1. Postupak za strojno potpomognutu izgradnju zbirke pitanja i odgovora temeljen na postupcima grupiranja tekstnih podataka i pretraživanja tekstnih informacija;
2. Postupak za strojno potpomognutu nadgradnju zbirke pitanja i odgovora temeljen otkrivanju nepostojećih odgovora u zbirci primjenom metoda strojnog učenja
3. Model za semantičko pretraživanje zbirke pitanja i odgovora temeljen na jezično neovisnim statističkim značajkama.

1.4. Struktura rada

Rad u nastavku strukturiran je na sljedeći način. Poglavlje 2. daje sažet pregled temeljnih postupaka iz područja strojnog učenja, obrade prirodnog jezika i pretraživanja informacija koji su korišteni u ovom istraživanju. Ostatak rada pretpostavlja da je čitatelj upoznat s pojmovima iz tog poglavlja. Slijedi detaljan opis korištenih referentnih skupova podataka u poglavlju 3. Zatim, niz predistraživanja koja su provedena za hrvatski i engleski jezik opisan je u poglavljima 4., 5. i 6. Nakon toga opisan je predloženi postupak za izgradnju zbirke u poglavlju 7. te je obrađen predloženi postupak za otkrivanje informacijskih potreba nepokrivenih FAQ-zbirkom u poglavlju 8. Slijedi opis modela za semantičko pretraživanje zbirke, koji je izložen u poglavlju 9. Konačno, poglavlje 10. zaključuje rad i predlaže smjerove za nastavak istraživanja.

Poglavlje 2.

Temeljni postupci i mjere

Većina pristupa istraženih u ovom radu temelji se na strojnom učenju. Ovo poglavlje daje kratak pregled postupaka strojnog učenja s kojima se potrebno upoznati kako bi se lakše moglo razumjeti nastavak rada. Postupci su podijeljeni na temeljne postupke iz strojnog učenja, temeljne postupke za pretraživanje informacija i postupke za učenje rangiranja. Nadalje, budući da je ispravno vrednovanje postupaka vrlo važno u strojnom učenju, zadnji odjeljak posvećen je opisu korištenih mjera vrednovanja.

2.1. Temeljni postupci iz strojnog učenja

Strojno učenje vrlo je moćan pristup rješavanju zadataka koji zahtijevaju ljudsku inteligenciju, pa je stoga i velik dio ovog rada temeljen na strojnom učenju. Pristupi strojnog učenja se mogu podijeliti na nadzirane i nenadzirane. Nadzirani pristupi uče rješavati neki zadatak na temelju podataka za koje je označen željeni izlaz postupka. Nenadzirani pristupi nemaju na raspolaganju oznaku željenog izlaza, već se temelje na pronalaženju uzoraka u podacima. Tipičan primjer nadziranog pristupa je klasifikacija, dok je tipičan primjer nenadziranog pristupa grupiranje. Ovaj rad koristi obje vrste pristupa, a odabrani su oni pristupi koji su se u literaturi pokazali kao vrlo dobri na sličnim problemima. U nastavku, dan je kratak opis korištenih pristupa.

2.1.1. Stroj potpornih vektora

Klasifikacijski model temeljen na stroju potpornih vektora (engl. *support vector machine* – SVM) prvi put su predložili Cortes i Vapnik (1995). Radi se o nadziranom klasifikacijskom postupku. Model pretpostavlja da je oblik granice između klasa u prostoru značajki predstavljen s linearnom funkcijom (hiperravninom) u prostoru značajki, definiranom kao $\mathbf{w}^T\mathbf{x} + w_0$. Za razliku od svih do tada predloženih modela, SVM je jedinstven po tome što je u funkciju gubitka ugrađena maksimizacija margine između razreda. Margina je definirana kao najmanja

udaljenost nekog primjera za učenje od granice između klasa. Posljedica toga je da, od svih mogućih granica između razreda, optimizacijski algoritam koji uči klasifikacijski model pronalazi onu koja će najbolje generalizirati. Moderne varijante ovog postupka obično koriste formulaciju tzv. meke margine, gdje je dopušteno da pojedini primjeri budu unutar margine između razreda, ili čak s njene pogrešne strane. No, za takve slučajeve model se kažnjava kroz funkciju gubitka. Učenje modela traži $(\mathbf{w}, w_0)$ koji minimiziraju, uz određena ograničenja, sljedeći izraz:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

Lijevi član predstavlja maksimizaciju margine (širina margine je $\frac{2}{\|\mathbf{w}\|}$) dok desni predstavlja funkciju gubitka koja kažnjava pogreške modela. Funkcija gubitka koja se koristi je funkcija zglobnice (engl. *hinge loss*). Ona je definirana za primjer x_i kao $\xi_i = L(\mathbf{x_i}) = max(0, 1 - y_i(\mathbf{w}^T\mathbf{x_i} + w_0))$. Pri tome će primjeri x_i koji su s prave strane margine imati gubitak jednak 0, dok će oni unutar margine ili s njene pogrešne strane imati gubitak proporcionalan tome koliko su daleko od ispravne strane margine. Ovaj optimizacijski problem može se svesti na kvadratno programiranje te riješiti za to prikladnim algoritmima (Joachims, 1998; Platt, 1998).

Iako osnovni model SVM pretpostavlja linearnu granicu između primjera, može se postići da model postane nelinaran. Ovo je najlakše napraviti pomoću nelinearnog preslikavanja originalnih primjera u prostoru značajki u prostor značajki više dimenzije. Dodatno, uz prikladnu reformulaciju gornjeg optimizacijskog problema moguće je koristiti i *jezgreni trik*. Time se izbjegava eksplicitno preslikavanje primjera, već se ono radi implicitno kroz jezgrenu funkciju koja predstavlja skalarni produkt u prostoru više dimenzije. Zbog toga model SVM može raditi čak i u beskonačno dimenzijskim prostorima značajki, što mu omogućava uspješno modeliranje granice vrlo složenog oblika.

Zbog visoke složenosti modela SVM, vrlo je važno napraviti regularizaciju kako se model ne bi prenaučio. Za model SVM to se postiže pomoću dva hiperparametra. Prvi je hiperparametar C – kazna koju model dobiva za krivo klasificiran primjer. Drugi hiperparametar je γ, koji predstavlja širinu jezgrenih funkcija te se uzima u obzir samo ako se koristi radijalna bazna jezgrena funkcija (engl. *radial basis function*). Vrijednosti ovih hiperparametara potrebno je optmirati na izdvojenom skupu za provjeru postupkom pretrage po rešetci (engl. *grid search*).

Osim modela SVM za binarnu klasifikaciju postoji i jednorazredni SVM, predložen u (Schölkopf i dr., 2001), gdje su za učenje dostupni primjeri samo jednog razreda. Zadatak modela jest za neviđene primjere odrediti pripadaju li također u taj razred. Dodatno, postoje brojne druge formulacije SVM-a za, primjerice, regresijske zadatke (Drucker i dr., 1997) ili rangiranje (Joachims, 2002).

2.1.2. Hijerarhijsko aglomerativno grupiranje

Hijerarhijsko aglomerativno gurpiranje (engl. *hierarchical agglomerrative clustering* – HAC) klasičan je nenadziran postupak za grupiranje podataka. Postupak kreće od N primjera od kojih je svaki u početku jednočlana grupa. Potom se u svakoj iteraciji određuju dvije najsličnije grupe i spajaju. Na taj način broj grupa u koje su primjeri svrstani se smanjuje za jedan u svakoj iteraciji. Postupak se ponavlja dok nije dosegnut željen broj grupa ili dok iznos mjere sličnosti između najsličnije dvije grupe ne padne ispod nekog praga. Da bi postupak radio, potrebno je definirati sličnost dviju grupa. To se može napraviti na više načina (Punj i Stewart, 1983). Ovaj jednostavan algoritam u praksi daje dobre rezultate, ali nije skalabilan zbog kvadratnog broja sličnosti koje je potrebno izračunati.

2.1.3. Postupak K-srednjih vrijednosti

Ovo je također jedan od klasičnih postupaka za grupiranje podataka. Postupak svrstava ukupno N primjera u K grupa tako što, nakon slučajne inicijalizacije centara grupa, iterira do konvergencije sljedeća dva koraka:

1. Svaki primjer $\mathbf{x_i}$ svrstaj u onu grupu j čiji centar $\mathbf{c_j}$ je najbliži $\mathbf{x_i}$;
2. Za svaku grupu j izračunaj novi centroid $\mathbf{c_j}$ kao prosjek svih $\mathbf{x_i}$ koji su trenutno u njoj.

Postupak staje kada se centroidi grupa prestanu mijenjati kroz iteracije. Iako jednostavan, ovaj postupak u praksi daje dobre rezultate te je vrlo skalabilan. Dodatna poboljšanja mogu se postići pažljivim odabirom pozicija na kojima će biti početni centri grupa (Pena i dr., 1999).

2.1.4. Grupiranje pomoću Gaussovih mješavina

Ovaj postupak grupiranja pretpostavlja da su primjeri koje treba grupirati u prostoru značajki zapravo izvučeni iz mješavine multivarijatnih Gaussovih razdioba. Uz zadan broj grupa K, koji predstavlja broj komponenata Gaussove mješavine, koristi se postupak maksimizacije očekivanja (engl. *expectation maximization* – EM) za pronalaženje parametara modela koji imaju najveću izglednost s obzirom na podatke. Parametri modela su vektor srednjih vrijednost i kovarijacijska matrica za svaku od komponenata mješavine te pripadne težine kojima se komponente kombiniraju u mješavini. Na ovaj način svaka naučena komponenta mješavine predstavlja jednu grupu. Primjeri se potom svrstavaju u grupe tako da se svaki primjer svrsta u onu grupu za koju mu naučen model Gaussove mješavine daje najveću vjerojatnost.

2.1.5. Spektralno grupiranje

Postupci spektralnog grupiranja temelje se na smanjenju dimenzija matrice sličnosti između primjera. Rastavom na vlastite vektore i vlastite vrijednosti te matrice dobiva se niskodimenzijski

vektorski prikaz za svaki primjer. Na tako reduciranim vektorskim prikazima primjera provodi se neki od postupaka za grupiranje, najčešće postupak K srednjh vrijednosti. Može se pokazati da je opisan postupak zapravo istovjetan rješavanju problema pronalaženja optimalnog normaliziranog reza na grafu sličnosti između primjera (Shi i Malik, 2000; Ng i dr., 2002). Postupci grupiranja iz ove skupine nemaju nikakvih pretpostavki o obliku grupa, pa su zato posebno prikladni na skupovima podataka gdje su grupe isprepletene ili vrlo nepravilnog oblika.

2.2. Temeljni postupci za pretraživanje informacija

Ovaj odjeljak opisuje korištene temeljne postupke za pretraživanje informacija. Mnogi od njih grade na pristupima iz strojnog učenja opisanima u prethodnom odjeljku. Svi pristupi za pretraživanje informacija iz ovog odjeljka su nenadzirani te se koriste kao referentni postupci za vrednovanje naprednijih pristupa pretraživanju.

2.2.1. Postupak temeljen na vektorskom prostoru

Ovaj postupak za pretraživanje temelji sa na prikazivanju teksta pomoću vektora utežanih načelom *frekvencija – inverzna frekvencija u dokumentima* (engl. *term frequency – inverse document frequency* – tf-idf). Ovi vektorski prikazi teksta imaju po jedan element za svaku riječ koja postoji u rječniku. Za vektorski prikaz konkretnog dokumenta d iz skupa dokumenata D vrijednost elementa koji odgovara riječi w računa se pomoću sljedećeg izraza:

$$\text{tf-idf}(w) = TF(w)IDF(w)$$

Pri tome je TF komponenta jednaka broju pojavljivanja riječi w u dokumentu d, dok se IDF komponenta računa prema sljedećem izrazu:

$$IDF(w) = \log \frac{|D|}{|D_w|}$$

Ovdje je sa D_w označen podskup skupa dokumenata D koji se sastoji od onih dokumenata koji sadrže riječ w.

Valja primijetiti da je IDF komponenta za riječi neovisna o korisničkom upitu, pa u implementaciji može biti unaprijed izračunata. Također, tf-idf vektori su tipično vrlo rijetki, što dodatno doprinosi učinkovitosti ovog postupka.

Pretraživanje informacija ovim postupkom se ostvaruje tako da se tf-idf vektorski prikazi upita usporede s vektorskim prikazima dokumenata koristeći kosinusnu sličnost vektora. Kada se pritom računa vektorski prikaz korisničkog upita, postupak je identičan kao i za dokumente, samo se kao $TF(w)$ uvrštava broj pojavljivanja riječi w u upitu. Vrijedi napomenuti da ovaj

postupak pretraživanja može umjesto tf-idf vektorskih prikaza teksta koristiti i bilo kakve alternativne vektorske prikaze, o čemu će biti još govora u nastavku.

2.2.2. BM25

BM25 je vrlo poznat postupak za pretraživanje informacija koji su predložili Robertson i dr. (1995). Postupak razmatra korisnički upit Q i potencijalno relevantni dokument D kao skupove riječi $\{q_1, ..., q_N\}$ i $\{d_1, ..., d_M\}$. Dokumente model rangira padajućim redoslijedom po iznosu sljedećeg izraza:

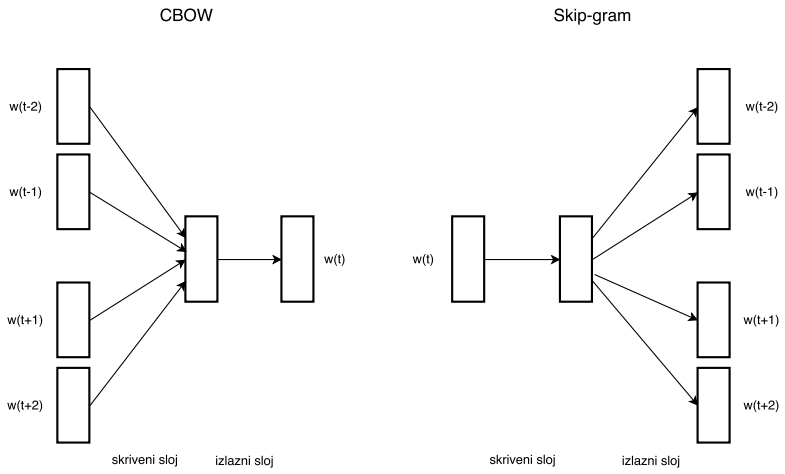
$$rel(Q,D) = \sum_{q \in Q} IDF(q) \frac{f_{q,D}(k_1+1)}{f_{q,D} + k_1(1 - b + b\frac{|D|}{L})} \tag{2.2.1}$$

pri čemu $f_{q,D}$ označava broj pojavljivanja riječi q u dokumentu D, a L prosječnu duljinu dokumenta. Veličine k_1 i b su hiperparametri modela, a $IDF(q)$ označava inverznu frekvenciju po dokumentima za riječ q koja je već opisana kod modela pretraživanja temeljenog na vektorskom prostoru. Konačan oblik izraza 2.2.1 rezultat je dugog niza pokusa te pokušaja i pogrešaka. Zato je ovaj izraz, iako jednostavan, vrlo učinkovit za pretraživanje informacija.

2.2.3. Latentna semantička analiza

Latentna semantička analiza postupak je za izgradnju semantičkih vektorskih prikaza riječi i dokumenata prvi put predložen u (Deerwester i dr., 1990). Postupak radi nad skupom dokumenata predstavljenim matricom A. Stupci matrice predstavljaju dokumente, dok retci predstavljaju riječi. Element A_{ij} ove matrice definiran je kao broj pojavljivanja riječi i u dokumentu j. Postupak LSA računa singularnu dekompoziciju ove matrice tako da pronalazi ortogonalne matrice U i V za koje vrijedi $A = UDV^T$. Nakon toga se radi smanjenje dimenzionalnosti na način da se zanemare svi singularni vektori i vrijednosti osim prvih k. Pri tome se dobivaju reducirane matrice U', D' i V' takve da je $A \approx U'D'V'^T$. Zanimljiv rezultat je da reducirana matrica U' za svaku riječ sadrži k dimenzionalni semantički vektor. Ovi vektori su takvi da semantički slične riječi (npr. *avion* i *zrakoplov*) imaju slične vektore. Slično, reducirana matrica V' sadrži semantičke vektore dokumenata. Dokumenti koji su semantički slični, iako ne koriste iste riječi, imaju slične vektore. Svojstvo modeliranja semantičke sličnosti predstavlja značajnu prednost ovog modela u usporedbi s modelima poput BM25 koji koriste samo površinske oblike riječi.

Dobiveni vektori riječi dobro modeliraju semantičku sličnost riječi, dok se dobiveni vektori dokumenata mogu koristiti umjesto tf-idf vektora u modelu pretraživanja temeljenom na vektorskom prostoru.



Slika 2.1: Prikaz arhitekture modela word2vec.

2.2.4. Modeli temeljeni na neuronskim prikazima riječi

Jedna alternativa postupku LSA, koju su prvi predložili Mikolov i dr. (2013), jest postupak word2vec. U tom postupku vektori riječi grade se iz vrlo velikog korpusa korištenjem neuronskih mreža koje uče povezati riječi koje se pojavljuju u sličnim kontekstima. Postoje dvije inačice postupka ovisno o tome koji zadatak neuronska mreža uči riješiti.

1. *CBOW* – zadatak je da na temelju konteksta tj. susjednih riječi predvidjeti trenutnu riječ.
2. *Skip-gram* – zadatak je na temelju trenutne riječi predvidjeti susjedne riječi iz konteksta

Arhitektura ove dvije inačice modela prikazana je na Slici 2.1. Obje inačice koriste neuronsku mrežu s jednim skrivenim slojem. Težine koje mreža nauči na vezama između neurona možemo interpretirati kao semantički vektor riječi. Semantički slične riječi imat će slične kontekste, pa će za njih mreža naučiti slične težine, odnosno njihovi semantički vektori bit će slični. Pokazuje se da ovi vektori imaju i zanimljiva svojstva linearne kompozicionalnosti. Na primjer, ako bismo sa semantičkim vektorima riječi izračunali operaciju *kralj* − *muškarac* + *žena*, rezultantni vektor bi bio vrlo blizu semantičkog vektora riječi *kraljica*. Ovo svojstvo može se iskoristiti za izgradnju semantičkih vektora cijelih tekstova zbrajanjem semantičkih vektora pojedinih riječi u njima.

Povezan, ali napredniji, neuronski model za izgradnju semantičkih vektora tekstova predložili su Wieting i dr. (2015). Taj model koristi označen skup parafraza (Ganitkevitch i dr., 2013) kako bi optimirao semantičke vektore riječi da dobro modeliraju parafraze. Vektor većeg teksta dobiva se zadanom operacijom kompozicije nad vektorima pojedinih riječi u njemu. Funkcija gubitka koju model optimira odražava poželjno svojstvo da tako dobiveni vektori tekstova koji jesu parafraze moraju po kosinusnoj sličnosti biti sličniji nego vektori tekstova koji nisu parafraze. U (Wieting i dr., 2015) razmatrano je više različitih operacija kompozicije, no pokazalo se da već jednostavno uprosječavanje daje vrlo dobre rezultate.

Ovako dobiveni vektori tekstova mogu se iskoristiti umjesto vektora tf-idf u modelu pretraživanja temeljenom na vektorskom prostoru. Ipak, važno je napomenuti da kvaliteta gore opisanih svojstava kompozicionalnosti brzo opada kada se komponira velik broj riječi. Zato ovaj pristup modeliranju semantike teksta najbolje radi za kraće tekstove.

2.3. Postupci za učenje rangiranja

Učenje rangiranja (engl. *learning to rank*) (Liu i dr., 2009) je područje koje se bavi modelima temeljnim na nadziranome strojnom učenju koji su posebno prilagođeni da uče rješavati probleme rangiranja. Za takve modele željeni izlaz predstavljen je poredanim popisom dokumenata. U ovom odjeljku opisat ćemo tri takva algoritma.

2.3.1. ListNET

Prvi razmatrani model je ListNET, predložen u (Cao i dr., 2007). Model kao jedan primjer za učenje uzima u obzir korisnički upit i cijeli dohvaćen popis dokumenata. Za svaki par upita i dokumenta generira se vektor značajki $\mathbf{x}$ koji predstavlja ulaz u model rangiranja temeljen na linearnoj neuronskoj mreži – ona računa izlaz kao $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Izlaz mreže je zapravo ocjena relevantnosti dokumenta za dani upit. Kod učenja mreže minimizira se pogreška unakrsne entropije koja je za i-ti primjer za učenje dana sljedećim izrazom:

$$L^{(i)}(y^{(i)}, z^{(i)}) = -\sum_{\forall g \in G_k^{(i)}} P_{y^{(i)}}(g) \log P_{z^{(i)}}(g)) \tag{2.3.2}$$

Ovdje su sa y^i označene ocjene koje su dokumentima dali ljudski označivači, a sa $z^{(i)} = h(\mathbf{x^{(i)}})$ ocjene relevantnosti koje je dala neuronska mreža. U (Cao i dr., 2007) pokazano je da se iz svakog skupa ocjena relevantnosti može inducirati vjerojatnosna distribucija P koja ide po permutacijama dokumenata (engl. *distribution over permutations*). Glavna ideja učenja modela jest da se distribucija po permutacijama koju induciraju ocjene modela $P_{z^{(i)}}$ približi distribuciji po permutacijama $P_{y^{(i)}}$ induciranoj iz ocjena ljudskih označivača. Vrijedi napomenuti da bi računanje funkcije gubitka za cijeli skup dokumenata bilo presporo, pa se u implementaciji zapravo koristi distribucija po grupama permutacija s identičnih prvih k elemenata. Skup takvih grupa u gornjem je izrazu označen s G_k^i. Model se uči postupkom gradijentnog spusta.

2.3.2. LambdaMART

Postupak LambdaMART, predložen u (Wu i dr., 2010), je kombinacija dvaju postupaka – LambdaRank i MART, koje je prvo potrebno sažeto opisati. Postupak LambdaRank predložili su

Burges i dr. (2007) i temelji se na aproksimaciji gradijenta funkcije gubitka koja je zapravo mjera NDCG, opisana u odjeljku 2.4. Aproksimirani gradijenti nazivaju se λ-gradijenti i sastoje se od dvije komponente definirane po svim parovima dokumenata[1] dohvaćenog popisa. Prva komponenta je RankNET (Burges i dr., 2005) funkcija gubitka – pogreška unakrsne entropije izračunata nad logističkom funkcijom razlike ocjena relevantnosti dokumenata u svakom paru. Druga komponenta aproksimiranog gradijenta je povećanje NDCG koje bi na popisu dokumenata (poredanih po ocjenama relevantnosti) uzrokovala zamjena dokumenata iz para. Prva komponenta ovisi samo o paru, no druga komponenta ovisi o cijelom dohvaćenom popisu dokumenata i mjeri koja se razmatra (u ovom slučaju NDCG). Za razliku od same mjere NDCG, ovako postavljena funkcija gubitka glatka je i derivabilna, što omogućuje primjenu postupaka temeljenih na gradijentnom spustu.

Drugi postupak čija načela su ugrađena u LambdaMART je postupak MART (višestruka aditivna regresijska stabla – engl. *multiple additive regression trees*). Taj je postupak prvi put opisan u (Friedman, 2001), gdje se kombinira više regresijskih stabla odluke kako bi se kroz radni okvir poticanja (*engl.* boosting) naučila aproksimacija funkcije. Ako je $\mathbf{x}$ vektor značajki dobivenih iz upita i dokumenta, funkcija ocjene relevantnosti modelirana pomoću M stabala dana je sljedećim izrazom:

$$h(\mathbf{x}) = \sum_{m=1}^{M} \rho_m h(\mathbf{x}, \alpha_{\mathbf{m}}) \tag{2.3.3}$$

gdje su $\alpha_{\mathbf{m}}$ parametri m-tog stabla, a ρ_m njegova težina u modelu poticanja. Algoritam se temelji na tome da se gradijenti u odnosu na funkciju gubitka u svim pozicijama u prostoru aproksimiraju na temelju konačno mnogo točaka iz skupa za učenje. Aproksimacija gradijenta modelira se standardnim regresijskim stablom odluke kod kojeg se koristi postupak najmanjih kvadrata za određivanje optimalnih podjela čvorova. Takva aproksimacija gradijenta dodaje se u model kao još jedno stablo uz pripadnu težinu. Tako se svaka iteracija algoritma može interpretirati kao pomak u prostoru svih mogućih rješenja $h(\mathbf{x})$ u smjeru negativnog gradijenta. Ovakav način učenja modela zove se gradijentno poticanje (engl. *gradient boosting*).

Konačno, postupak LambdaMART temelji se na kombinaciji postupka LambdaRank i MART. Oni su kombinirani tako što se koristi MART postupak gradijentnog poticanja za učenje modela, a kao gradijenti se u njemu koriste λ-gradijenti definirani za funkciju gubitka kakva se koristi u postupku LambdaRank. Ovaj pristup je brz te ima prednosti modela temeljenih na poticanju (fleksibilnost, interpretabilnost) i onih temeljenih na postupku LambdaRank (empirijski optimalna rješenja) (Wu i dr., 2010).

[1] Točnije, parovima vektora značajki od kojih je prvi izveden iz upita i prvog dokumenta, a drugi iz upita i drugog dokumenta.

2.3.3. Konvolucijske neuronske mreže za rangiranje

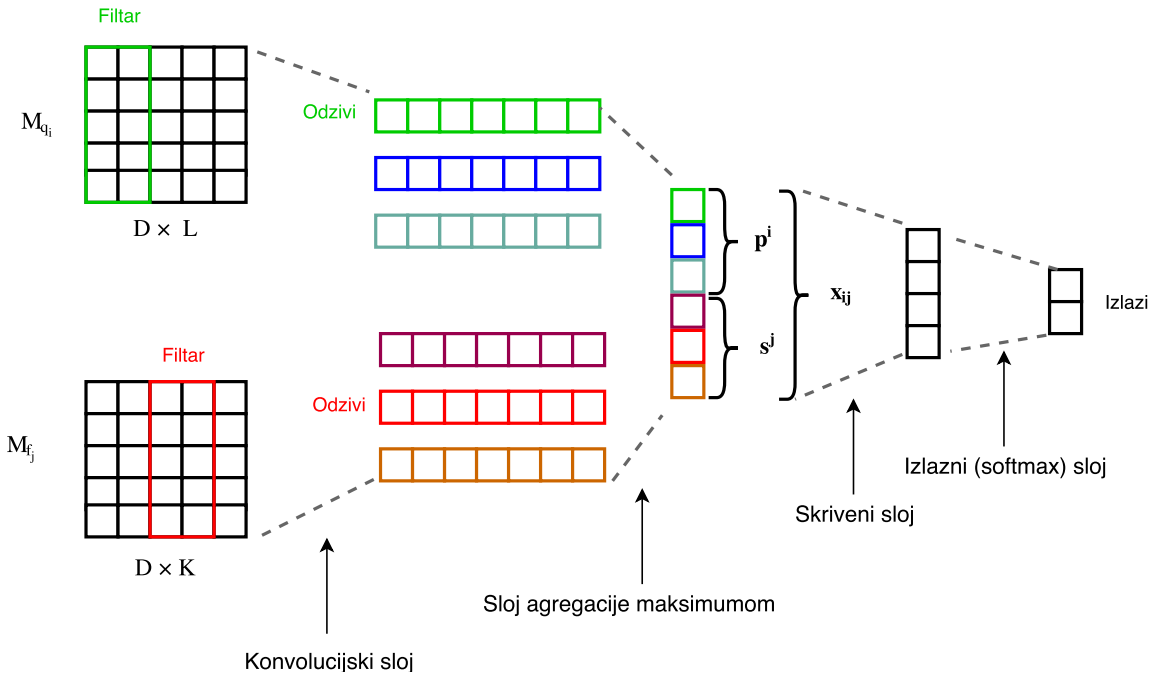
Postoji velik broj predloženih postupaka za učenje rangiranja temeljenih na neuronskim mrežama. Detaljan pregled srodnih modela iz literature može se naći u odjeljku 9.2. U okviru ovog rada najrelevantniji je model predložen u (Severyn i Moschitti, 2015), koji se pokazao vrlo uspješnim za dohvat odgovora na pitanja. Model je temeljen na konvolucijskoj neuronskoj mreži (engl. *convolutional neural network* – CNN) te problem rangiranja modelira kao klasifikacijski zadatak. Konkretno, potrebno je ostvariti preslikavanje $(q_i, d_j) \mapsto y_{ij}$, koje preslikava par upita i dokumenta u jednu od labela razreda – *relevantan* ako je d_j relevantan za q_i, ili *nerelevantan* ako to nije.

Model rangiranja temeljen na konvolucijskoj neuronskoj mreži kakav se koristi u ovom radu prikazan je na slici 2.2. Model predstavlja pojednostavljenu inačicu modela opisanog u (Severyn i Moschitti, 2015), u kojoj je uklonjena matrica sličnosti i dodatne značajke površinske sličnosti. Ulaz u mrežu je par upita i dokumenta (u ovom radu dokument je zapravo FAQ-par), koji je predstavljen pomoću dvije matrice čiji stupci predstavljaju semantičke vektore riječi u tekstu. Matrica M_{q_i} predstavlja upit, a M_{f_j} dokument. Prvi sloj mreže primjenjuje na ove matrice više odvojenih konvolucijskih filtara.[2] Rezultat ove operacije su dva skupa mapa značajki (engl. *feature maps*): jedan za upit i jedan za dokument. Sljedeći sloj radi agregaciju maksimumom (engl. *max-pooling*). Ovaj postupak za svaku mapu značajki izdvaja jedan, najveći izlaz. Tako dobiveni izlazi nad M_{q_i} se grupiraju u vektor $\mathbf{p^i}$, a nad M_{f_j} u vektor $\mathbf{s^j}$. Ovi vektori predstavljaju semantičke prikaze upita i dokumenta, a njihov spoj – vektor $\mathbf{x_{ij}}$ – semantički prikaz para korisničkog upita i dokumenta. Vektor $\mathbf{x_{ij}}$ je ulaz u skriveni sloj koji nad ulazom provodi nelinearnu transformaciju i na svoj izlaz daje $\phi(\mathbf{w}^T\mathbf{x})$. Neuroni ovog sloja na izlazu imaju nelinearnu prijenosnu funkciju ReLU ($\phi(\mathbf{x}) = \max(0, x)$). Zadnji sloj je izlazni sloj, koji na svom izlazu ima prijenosnu funkciju softmax (Bishop, 2006) s dva izlaza – po jedan za svaki od dva moguća razreda. Model uči standardnim gradijentnim spustom kroz postupak propagacije pogreške unatrag (engl. *error backpropagation*) (Rumelhart i dr., 1988). Kao funkcija gubitka koristi se logaritam pogreške unakrsne entropije pozbrajan po svim parovima upita i dokumenta. Naučeni model rangira dokumente za zadani korisnički upit tako što ih poreda silazno po vrijednosti za klasu *relevantan* dobivenu kao izlaz funkcije softmax.

2.4. Mjere za vrednovanje

Sve mjere za vrednovanje opisane u ovom odjeljku koriste se za vrednovanje postupaka za rangiranje. Iznimka je mjera F_1, koja se dodatno može koristiti i za vrednovanje postupaka za klasifikaciju ili grupiranje. Više detalja o ovim mjerama vrednovanja može se naći u (Manning

[2] Radi sažetosti, izostavljena su objašnjenja osnovnih pojmova konvolucijskih neuronskih mreža (poput filtara i sl.), koja se mogu pronaći u (LeCun i dr., 1998) i (Kim, 2014).



Slika 2.2: Skica arhitekture konvolucijske neuronske mreže kakva se koristi za rangiranje u ovom radu.

Tablica 2.1: Popis slučajeva za mjeru F_1.

Stvarno stanje	Odluka modela	Ishod za taj primjer
pozitivan	pozitivan	stvarno pozitivan (engl. *true positive* – TP)
negativan	negativan	stvarno negativan (engl. *true negative* – TN)
negativan	pozitivan	lažno pozitivan (engl. *false positive* – FP)
pozitivan	negativan	lažno negativan (engl. *false negative* – FN)

i dr., 2008).

Mjera F_1 (engl. F_1 *measure*). Ovo je vrlo često korištena mjera u području strojnog učenja. Najčešće se koristi za vrednovanje klasifikacijskih postupaka. Ako pretpostavimo da model strojnog učenja rješava binarni klasifikacijski problem gdje treba ulazni primjer svrstati u pozitivan ili negativan razred, postoje četiri moguća ishoda, kao što prikazuje Tablica 2.1.

Sada možemo definirati veličine preciznosti $P = \frac{TP}{TP+FP}$ i odziva $R = \frac{TP}{TP+FN}$. Ove dvije mjere su komplementarne te kažnjavaju različite vrste grešaka koje model može napraviti. Preciznost kažnjava modele s previše lažno pozitivnih odluka, dok odziv kažnjava modele s previše lažno negativnih odluka. Mjera F_1 definirana je kao harmonijska sredina preciznosti i odziva, tj. $F_1 = \frac{2PR}{P+R}$. Mjeru je lako prilagoditi slučaju sa više klasa (Yang i Liu, 1999).

Srednji recipročni rang (engl. *mean reciprocal rank*) – MRR. Mjera je koja vrednuje rezultate postupka za pretraživanje na skupu korisničkih upita. Mjera je definirana kao:

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{r_q}$$

gdje je Q skup upita a r_q rang najbolje rangiranog relevantnog dokumenta za upit q. Ova mjera tipično se koristi u slučajevima kada nije važno da svi relevantni dokumenti budu dohvaćeni, već je prioritet da barem jedan od njih bude rangiran što bolje. Ova mjera je jedna od najčešće korištenih mjera u području pretraživanja informacija.

Preciznost na rangu K (engl. *precision at k*) – P@k. Mjera se računa prema sljedećem izrazu

$$P@K = \frac{1}{|Q|} \sum_{q \in Q} \frac{n_{kq}}{k}$$

gdje je sa n_{kq} označen broj relevantnih dokumenata koji su dohvaćeni u najbolje rangiranih k rezultata za upit q. Ova se mjera koristi kada je važno da što više rezultata pri vrhu dohvaćenog popisa bude relevantno.

Odziv od prvih K (engl. *recall out of k*) – ROOk. Ova se mjera računa kao

$$ROOk = \frac{1}{|Q|} \sum_{q \in Q} \min(n_{kq}, 1)$$

gdje n_{kq} ima isto značenje kao i za prethodnu mjeru. Mjera ROOk može se interpretirati kao udio upita za koje se relevantan dokument može pronaći u prvih k rezultata. Korištenje ove mjere prikladno je u slučajevima kada nije bitno dohvatiti sve relevantne dokumente, već barem jedan bilo gdje u prvih k rezultata.

Srednja prosječna preciznost (engl. *mean average precision*) – MAP. Ova mjera za evaluaciju rangiranja uzima u obzir cijeli popis dohvaćenih dokumenata, a računa se prema sljedećem izrazu:

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q)$$

pri čemu $AP(q)$ predstavlja srednju preciznost za upit q definiranu kao

$$AP(q) = \frac{1}{|R|} \sum_{k \in R} \frac{n_{kq}}{k}$$

gdje n_{kq} ima isto značenje kao i u prethodnim izrazima, a R je skup koji sadrži rangove na kojima su dohvaćeni dokumenti relevantni za upit q.

R preciznost (engl. *R precision*) – RP. Ova mjera za svaki upit razmatra broj relevantnih dokumenata dohvaćenih u prvih r_q rezultata, pri čemu je r_q broj relevantnih dokumenata za upit q. Izraz za računanje ove mjere je

$$RP = \frac{1}{|Q|} \sum_{q \in Q} \frac{n_{r_q q}}{r_q}$$

gdje $n_{r_q q}$ označava broj dohvaćenih relevantnih dokumenata u prvih r_q rezultata. Vrijedi napomenuti da broj r_q varira od upita do upita. Pokazuje se da je ova mjera visoko korelirana s mjerom MAP.

Normaliziran smanjen kumulativni dobitak (engl. *normalized discounted cumulative gain)* – NDCG. Ova mjera može, za razliku od do sada opisanih mjera, raditi i sa stupnjevanim ocjenama relevantnosti (engl. *graded relevance judgements*), pa je prikladna kada su takve ocjene dostupne. Pretpostavka je da je za svaki upit q_i i svaki dokument d_j označena veličina $rel_j^{(i)}$ koja predstavlja razinu relevantnosti dokumenta j za upit i (veća vrijednost znači veću relevantnost). Kako bi se došlo do mjere NDCG, prvo je dobro razmotriti mjeru kumulativnog dobitka (engl. *cumulative gain* – CG) na rangu k koja je definirana kako slijedi.

$$CG_k^{(i)} = \sum_{j=1}^{k} rel_j^{(i)}$$

Intuicija ovog izraza jest da će mjera biti to veća što je više dokumenata visokog iznosa $rel_j^{(i)}$ u prvih k rezulta. No, problem je što se ne uzima u obzir poredak dokumenata. Idealno, mjera bi dokumentima koji su bliže vrhu popisa trebala davati veći utjecaj na konačni rezultat. Ovo je riješeno u mjeri smanjenog kumulativnog gubitka (engl. *discounted cumulative gain*) tako što je dokumentima blizu vrha popisa dana veća težina, što se vidi iz sljedećeg izraza:

$$DCG_k^{(i)} = \sum_{j=1}^{k} \frac{2^{rel_j^{(i)}} - 1}{\log_2(j+1)}$$

Konačno, zadnji problem jest što broj relevantnih dokumenata varira od upita do upita. Također, variraju i iznosi ocjena relevantnosti $rel_j^{(i)}$. Zbog ovoga iznosi mjere dobiveni za različite upite nisu usporedivi, pa zato nema smisla računati njihov prosjek po više upita. Ovaj se problem može riješiti normalizacijom, što nas vodi do mjere NDCG, koja se računa prema sljedećem izrazu:

$$NDCG_k^{(i)} = \frac{DCG_k^{(i)}}{N^{(i)}}$$

gdje je sa $N^{(i)}$ označen teoretski maksimalan $DCG_k^{(i)}$, tj. onaj iznos koji bi se dobio da je popis dohvaćenih dokumenata bio silazno sortiran po $rel_j^{(j)}$. Sada se konačan iznos mjere može dobiti kao prosjek $NDCG_k^{(i)}$ po svim upitima q_i.

Poglavlje 3.

Referentni skupovi podataka

Skupovi podataka važni su kako bi se moglo ispravno vrednovati modele pretraživanja. Također, bez njih se ne bi moglo izgraditi modele pretraživanja temeljene na nadziranom strojnom učenju. U okviru rada, izgrađena su tri skupa podataka na engleskom jeziku: VerizonFAQ, FAQIR i StackFAQ. Dodatno, u istraživanju se koristi i otprije izgrađen skup podataka VipFAQ koji je na hrvatskom jeziku.

U ovom poglavlju prvo je dan opširan pregled prednosti i nedostataka skupova podataka dostupnih u literaturi. Potom, slijedi detaljan opis svih skupova podataka koji su korišteni za izgradnju i vrednovanje postupaka predloženih u ovom radu.

3.1. Skup podataka u kontekstu pretraživanja FAQ-zbirki

Kako bismo mogli vrednovati modele i postupke predložene u ovom radu potrebni su nam gotovi FAQ skupovi podataka. Ovakvi skupovi podataka su specijalni slučajevi općenitih skupova podataka za pretraživanje informacija (engl. *IR test collections*), kakvi su predloženi u (Voorhees i Harman, 2005). U kontekstu FAQ pretraživanja ovakav skup podataka sastoji se od (1) FAQ-zbirke, (2) skupa upita usmjerenih na FAQ-parove iz zbirke i (3) skupa oznaka relevantnosti koji govori koji su FAQ-parovi relevantni za koje upite. S obzirom na zadatke kojima se bavi ovaj rad, FAQ skup podataka bi dodatno trebao imati i sljedeća svojstva:

1. **Dokumenti kao FAQ-parovi.** Svaki dokument u zbirci sastoji se od jednog pitanja i točnog odgovora na to pitanje. Ovo svojstvo, koje je tipično za FAQ-zbirke, ne zadovoljavaju mnoge postojeće zbirke, najčešće iz područja odgovaranja na pitanja na temelju zajednice (engl. *Community Question Answering* – CQA). Kod takvih je zbirki dokument koji sustav za pretraživanje traži zapravo samo pitanje koje je netko drugi postavio, koje može biti neodgovoreno, ili istovremeno imati više odgovora, koji nisu nužno svi točni;
2. **Usredotočenost na usku domenu.** FAQ-zbirke najčešće se koriste upravo za rješavanje čestih informacijskih potreba iz uske domene. Stoga ovo svojstvo doprinosi realnosti

rezultata vrednovanja modela i postupaka na takvom skupu;

3. **Dovoljna veličina za nadzirano strojno učenje.** Većina postupaka predloženih u ovom radu temeljena je na postupcima nadziranog strojnog učenja. Zato je vrlo važno da veličina skupova podataka na kojima se provode pokusi bude dovoljna za primjenu ovih postupaka;
4. **Zalihost na razini upita.** Ovo se svodi na prisutnost dovoljnog broja parafraza svake informacijske potrebe prisutne u skupu podataka. U kontekstu ovog rada, postoje tri razloga zašto je ovo svojstvo važno. Prvi razlog je za vrednovanje prve komponente strojno-potpomognutog postupka za izgradnju FAQ zbirke, opisane u odjeljku 7.3. Drugi je razlog izgradnja modela za detekciju nedostajućih informacijskih potreba, opisanog u odjeljku 8.3. Treći je razlog izgradnja nadziranih modela strojnog učenja za pretraživanje FAQ-zbirki, što je detaljno objašnjeno u odjeljku 9.1.

3.2. Pregled FAQ skupova podataka iz literature

U literaturi postoje FAQ skupovi podataka koji imaju neka, no ne i sva važna svojstva navedena u prethodnom odjeljku. Skupovi podataka[1] iz (Agarwal i dr., 2012; Feng i dr., 2015; Moschitti i Quarteroni, 2011; Surdeanu i dr., 2008; Kothari i dr., 2009; Bickel i Scheffer, 2004) sastoje se od dokumenata koji sadrže samo odgovore, ali ne i potpune FAQ-parove. S druge strane, skup podataka iz (dos Santos i dr., 2015) ima dokumente koji pak sadrže samo pitanja. Jijkoun i de Rijke (2005) izgradili su velik skup podataka koji sadrži dokumente u obliku FAQ-parova, dok su Jeon i dr. (2006) izgradili sličan, ali manji skup podataka za korejski jezik. No, oba ova skupa podataka nisu domenski specifična.

Važnije od tipa dokumenta i domenske specifičnosti jest to što niti jedan od do sada spomenutih skupova podataka nema zalihost na razini upita. Taj kriterij ispunjava skup podataka opisan u (Wang i dr., 2009). No, osim što nije javno dostupan, taj skup podataka sadržava parafraze upita dobivene grupiranjem, što znači da bi one mogle manje kvalitetne nego parafraze dobivene kroz ljudsko označavanje. Nedavno izgrađen skup podataka CQADupStack (Hoogeveen i dr., 2015) ima sva gore navedena svojstva. Posebice, oznake o podvostručenim pitanjima uvode, do neke mjere, zalihost na razini upita.[2] Na žalost, skupovi podvostručenih pitanja u ovom skupu podataka premaleni su za pokuse u ovom radu. U velikoj većini slučajeva broj podvostručenih pitanja za neku informacijsku potrebu je samo dva ili tri, što nije dovoljno za primjenu u pokusima ovog istraživanja. Ova pojava je očekivana, budući da je skup podataka CQADupStack nastao iz podataka na CQA stranici, koja aktivno potiče korisnike da *izbjegavaju* postavljanje podvostručenih pitanja. Konačno, skup podataka opisan u (Figueroa i Neumann,

[1] Svi skupovi spomenuti u ovom odjeljku su, ako nije drugačije napomenuto, na engleskom jeziku.
[2] Pitanja koja su označena kao duplikati su zapravo parafraze iste informacijske potrebe.

2013) izgrađen je na temelju podataka o klikovima na stranici Yahoo Answers.[3] Ovaj skup ima visoku razinu zalihosti na razini upita, no nažalost nije javno dostupan.

Zbog toga što niti jedan javno slobodno dostupan skup podataka nije imao sva željena svojstva, za potrebe ovog istraživanja izgrađen je niz novih FAQ skupova podataka. Ostatak ovog poglavlja detaljno opisuje izgradnju ovih skupova i predistraživanja na njima.

3.3. Skup podataka VerizonFAQ

Za potrebe predistraživanja prvo je izgrađen malen, domenski specifičan FAQ-skup. Skup je izgrađen tako što su preuzeti javno dostupni FAQ-parovi sa FAQ-stranice velike telekomunikacijske tvrtke.[4] Iz ovih podataka uzorkovano je 500 FAQ-parova koji su bili uvršteni u novonastalu FAQ-zbirku. Ovi FAQ-parovi predstavljaju skup dokumenata koje će korisnici FAQ-zbirke pretraživati.

U svrhu izgradnje skupa upita, tri ljudska označivača oblikovala su parafraze za pitanja u svakom od 500 FAQ-parova. Ove parafraze se smatraju upitima. Za svaki od upita, onaj FAQ-par čijim parafraziranjem je upit nastao, smatra se relevantnim dokumentom za dotični upit. Označivači su dobili upute da osmisle realistične parafraze, slične upitima koje bi stvarni korisnici mogli uputiti sustavu za pretraživanje FAQ-zbirke. Svaki FAQ-par označio je jedan od označivača, osmislivši pri tome tri parafraze. Nakon uklanjanja nekih nejasnih slučajeva, skup podataka ima 486 FAQ-parova i 1.450 upita. Primjeri iz ovog skupa podataka mogu se naći u tablici 3.1.

U predistraživanjima na ovom skupu, opisanim u poglavlju 4., pokazalo se da on nije vrlo izazovan za modele pretraživanja informacija. Točnije, već vrlo jednostavne temeljne metode pretraživanja postižu izvrsne rezultate. Ovaj je problem posljedica toga što označivači nisu kod parafraziranja u dovoljnoj mjeri izmijenili upite. Točnije, usprkos izmjeni samo nekih od riječi u upitu ili izmjeni sintaktičke strukture upita, leksičko preklapanje parafraziranog upita s izvornim upitom najčešće je i dalje vrlo visoko. To pogoduje modelima za pretraživanje koji se temelje na jednostavnoj analizi leksičkog preklapanja između upita i FAQ-para. Kako je takva zbirka nerealistična, izgrađena je nova zbirka koja je opisana u sljedećem odjeljku.

[3] `https://answers.yahoo.com`
[4] `http://www.verizon.com`

Tablica 3.1: Primjeri FAQ-pitanja u skupu VerizonFAQ i njihovih parafraza koje se koriste kao upiti.

Pitanje	Upit
What other refill options are available through my prepaid device?	What are the possible ways of refilling with a prepaid device?
How soon will I be able to use an added feature?	The timespan from adding a feature to using it.
What is local number portability?	Can I keep my phone number when changing service providers?

3.4. Skup podataka FAQIR

3.4.1. FAQ-parovi

FAQ-parovi za ovaj skup izdvojeni su iz javno dostupne[5] zbirke FAQ-parova, koju su izgradili Surdeanu i dr. (2011). Izvorna zbirka izgrađena je korištenjem stranice *Yahoo Answers*[6] te pokriva vrlo širok spektar različitih domena. Budući da je ovo istraživanje usmjereno na domenski specifične FAQ-zbirke, u izgradnji ovog skupa podataka razmatran je samo onaj dio FAQ-parova koji pripadaju jednoj specifičnoj kategoriji na stranici. Konkretno, korištena je kategorija "maintenance & repairs" (održavanje i popravci), koja sadrži pretežito pitanja o održavanju kuće/stana. Ovo ostavlja ukupno 4.313 FAQ-parova koji sačinjavaju FAQ-zbirku ovog FAQ-skupa.

3.4.2. Upiti

Sljedeći korak izgradnje FAQ skupa jest osmišljavanje upita koji će ciljati na neki od dobivenih FAQ-parova. Kako bi se smanjio potreban ljudski rad, označavanje je provedeno u tri koraka te su u njemu sudjelovala tri ljudska označivača. Označavanje je provedeno na sljedeći način:

- Označivač A1 izgradio je 50 *predložaka upita*. Predložak upita (PU) definiran je kao opis informacijske potrebe korisnika, koji olakšava osmišljavanje upita u sljedećem koraku. Primjer PU-a jest *"Useful information on how to keep insects out of a house/apartment/home."* (*"Korisne informacije o tome kako spriječiti ulazak insekata u kuću/stan/dom."*);
- Isti označivač A1 tada je osmislio osam različitih upita za svaki PU. Kod ovog zadatka bilo je dopušteno u upitima navoditi dodatne kontekstne informacije koje ne mijenjaju osnovnu informacijsku potrebu izraženu kroz PU. Na primjer, za PU *"How to get stains out of carpet."* (*"Kako maknuti mrlje s tepiha."*), jedan osmišljen upit bi mogao biti *"My son spilled cocoa on my brand new carpet what can I do to fix it?"* (*"Moj sin je prolio*

[5] `https://webscope.sandbox.yahoo.com/`
[6] `https://answers.yahoo.com`

kakao po mom sasvim novom tepihu, kako mogu to popraviti?"). Ovo značajno olakšava osmišljavanje raznolikih upita za isti PU. Dodatno, dodavanje kontekstnih informacija povećava leksički jaz između upita i potencijalno relevantnih dokumenata, pa time čini ovaj FAQ skup podataka izazovnijim i realističnijim (pitanja s dodatnim kontekstom su česta na stranici Yahoo Answers);

- Konačno, označivači A2 i A3 obavili su isti zadatak – osmislili su po osam upita za svaki PU. Njihovo raznoliko shvaćanje PU-ova omogućilo im je da osmisle upite kojih se A1 nije sjetio, što pomaže da se još više poveća raznolikost upita za svaki PU. Ipak, kako se upiti ovih označivača ne bi previše semantički udaljili od informacijske potrebe definirane preko PU-a, svaki od označivača dobio je uz PU i dva pripadna upita iz prethodnog koraka kao referentne primjere. Označavanjem je prikupljeno 24 upita za svaki PU. Označivač A1 je naknadno pregledao sve upite i provjerio da su u potpunosti usklađeni s pripadnim PU-om.

3.4.3. Oznake relevantnosti

Zadnji korak koji je potrebno provesti jest označavanje relevantnosti. Relevantnost se označava na razini svakog PU-a. Posljedica ovoga jest da upiti koji su nastali na temelju istog PU-a imaju iste relevantne FAQ-parove. Ovaj način označavanja osjetno smanjuje ukupnu količinu posla za označivače. Kako bismo za svaki PU dobili skup dobrih kandidata za relevantne FAQ-parove, koristimo tehniku agregiranja rezultata više modela pretraživanja (engl. *pooling technique*), koja se često primijenjuje pri vrednovanju sustava za pretraživanje informacija (Manning i dr., 2008). Za svaki PU koristimo postupke BM25, VS i SG, opisane u odjeljku 2.2. Za svaki zadani PU sva tri postupka pretraživanja primjenjuju se na sve upite koji pripadaju tom PU. Ovo generira prosječno 72 rezultata[7] za svaki PU. Skup dobrih kandidata za neki PU dobiva se kao podskup onih FAQ-parova koji se pojavljuju u prvih deset FAQ-parova u *barem jednom* od rezultata za taj PU.

Kako bi se bolje opisale različite dimenzije relevantnosti koje se mogu pojaviti u kontekstu pretraživanja FAQ-zbirki, ne koriste se standardne binarne oznake relevantnosti. Umjesto toga, označivači su relevantnost označili pomoću višerazinskog sustava oznaka. Korištene oznake za svaki par upita i FAQ-para iz skupa dobrih kandidata su kako slijedi:

- *Relevantan* (engl. *relevant*) (R) – FAQ-par je u potpunosti relevantan za informacijsku potrebu iskazanu kroz PU;
- *Koristan* (engl. *useful*) (U) – FAQ-par nije sasvim usklađen sa informacijskom potrebom iskazanom u PU, ali ipak daje informacije koje bi korisniku ipak mogle biti korisne u kontekstu te informacijske potrebe. Na primjer, PU *"Information about removing bad odors*

[7] Za svaki PU upotrijebljena su 3 modela pretraživanja puta 24 upita. Jedan rezultat zapravo je rangirana lista FAQ-parova.

Tablica 3.2: Primjeri oznaka koje se koriste u skupu podataka FAQIR.

Oznaka	Opis	Primjer
Relevantan (R)	Savršeno poklapanje	Q: *How to repair a lawn mowing machine* ... ? (*Kako popraviti kosilicu* ... ?) A: *First, check the fuel line by disconnecting it and* ... (*Prvo, provjerite crijevo s benzinom tako da ga izvadite i* ...)
Koristan (U)	Djelomično poklapanje i korisno	Q: *Where to find gardening equipment handyman?* (*Gdje mogu naći majstora za dvorišni alat?*) A: *You can usually find one through newspaper adds, or contact an agency that will* ... (*Obično ga možete naći preko oglasa u novinama, ili kontaktirajte agenciju koja će* ...)
Beskoristan (X)	Djelomično poklapanje i nije korisno	Q: *How can I fix my lawnmower?* (*Kako mogu popraviti svoju kosilicu?*) A: *Throw it away and buy a new one.* (*Baci ju u smeće i kupi novu.*)
Nerelevantan (N)	Nema poklapanja	Q: *Getting stains out of a carpet?* (*Uklanjanje mrlja s tepiha?*) A: *It's best to use lots of water and* ... (*Najbolje je koristiti puno vode i* ...)

from car" (*"Informacije o uklanjanju neugodnih mirisa iz auta"*) i FAQ-par *"Removing gasoline smell from garage"* (*"Uklanjanje mirisa benzina iz garaže"*);

- *Beskoristan* (engl. *Useless*) (X) – FAQ-par je tematski usklađen s informacijskom potrebom iz PU-a, ali nije koristan. Na primjer, PU – *"Information on getting mold out of a fridge."* (*"Kako ukloniti plijesan iz frižidera"*) i FAQ-par u kojem se pitanje dobro poklapa sa PU ali s *"With a chainsaw"* (*"Motornom pilom"*) kao odgovorom;
- *Nerelevantan* (engl. *non-relevant*) (N) – FAQ-par je u potpunosti nepovezan s informacijskom potrebom iz PU.

Dodatni primjeri ovih oznaka dani su u tablici 3.2. Vrijedi napomenuti da ove oznake zapravo predstavljaju razinu relevantnosti FAQ-para u odnosu na informacijsku potrebu iskazanu kroz PU. Slična shema oznaka predložena je za označavanje relevantnosti u (Bunescu i Huang, 2010).

Jedan označivač (A1) sam je označio cijeli skup podataka. Posao je uključivao dodjelu jedne od gore navedenih oznaka svakom paru PU-a i nekog FAQ-para koji je dobar kandidat za taj PU. U prosjeku je za svaki PU bilo 202 FAQ-parova koji su označeni s jednom od gore navedenih oznaka. Svim FAQ-parovima koji nisu bili na popisu dobrih kandidata za neki PU automatski je pretpostavljena oznaka *nerelevantan* za taj PU. Označavanje je ukupno trajalo oko 40 sati. Dodatno, radi procjene slaganja označivača (engl. *inter annotator aggreement*), uzorak od pet

Tablica 3.3: Statistike skupa podataka FAQIR.

	Minimum	Maksimum	Prosjek	Medijan
Duljina upita (broj riječi)	1	26	7,3	7
Duljina FAQ pitanja (broj riječi)	1	94	12,3	8
Duljina FAQ odgovora (broj riječi)	1	376	33	22
Broj upita po PU (A1)	6	9	7,8	8
Broj upita po PU (A2)	6	8	7,5	8
Broj upita po PU (A3)	8	11	8,3	8
Broj dobrih kandidata po PU	72	358	202,8	200
Broj relevantnih kandidata po PU	1	56	9,8	5
Broj korisnih kandidata po PU	0	29	6,7	4

slučajno odabranih PU-ova označio je i drugi označivač A4, što je trajalo tri sata. Detaljnija analiza slaganja označivača dana je u sljedećem odjeljku.

Konačan skup podataka FAQIR sadrži 4.313 FAQ-parova i 1.233 upita (koji su stvoreni na temelju 50 PU-ova). Skup podataka ukupno ima 201.349 pojavnica i 11.752 različnica. Dodatne statistike skupa navedene su u tablici 3.3, a primjeri se mogu pronaći u tablici 3.4. Skup podataka javno je dostupan za istraživačke svrhe.[8]

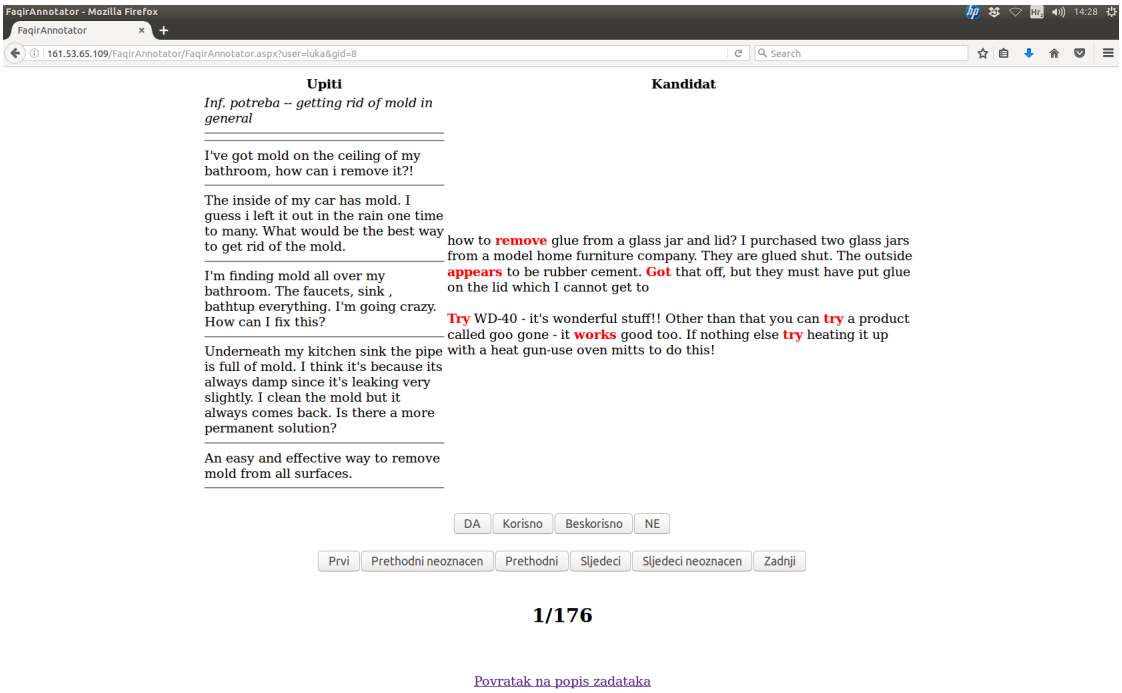
3.4.4. Implementacija sustava za označavanje

Za potrebe označavanja relevantnosti na ovom skupu, razvijen je sustav za označavanje koji je implementiran kao ASP.NET web aplikacija u programskom jeziku C#. Manji dio funkcionalnosti korisničkog sučelja sustava implementiran je u programskom jeziku JavaScript. Sustav podržava istovremen rad više označivača. Glavno sučelje sustava prikazano je slici 3.1. S lijeve strane označivaču je prikazana eksplicitno opisana informacijska potreba i pet parafraziranih upita koji iskazuju tu informacijsku potrebu. S desne strane prikazan je dokument koji je kandidat za relevantnost. Riječi koje su zajedničke dokumentu i nekoj od parafraza upita, označene su crvenom bojom. Ovo olakšava pronalazak riječi na koje je potrebno obratiti pažnju te tako štedi vrijeme označivačima.

[8]Dostupno pod CC BY-SA-NC licencijom na `http://takelab.fer.hr/faqir`

Tablica 3.4: Primjeri iz skupa podataka FAQIR. Relevantnost se definira prema shemi RU-XN.

Upit	FAQ-par	Relevantno?
How does one separate a carpet that's been superglued to the floor?	Q: How to Remove Glue From Carpet ? A: 1.What kind Hot glue you would need to warm it with an Iron with a towel in between and wipe up quickly2.White glue you would really have to wet the area wipe and use a product called awesome from the 1.00 store.3.Nail glue you would have to cut it out Little by little.	Da
	Q: How do I remove glued down carpet? A: Spend 20 bucks and get a heat gun. It will soften the glue and you can peel the carpet right up.	Da
	Q: How do you unscrew a superglue lid that has become superglued to the rest of the tube? The superglue is 110mL, so quite a bit is still left. I want to use it, but i can't get off the dried up lid. A: Acetone.(Or as the next poster says, nail polish remover, but it must be the acetone kind.)	Ne
Is it okay just to drive short screws through the floor in order to make it less squeaky? I am no carpenter and I wouldn't like to damage floor completely.	Q: How do you keep hardwood floors from creaking? A: Anywhere there is a creaking sound, you can try to sprinkle a little talcum powder in the cracks. If the cracks are too tight, try to get at the spot(s) from underneath and get the powder between the joist and the floor board. The sound is caused by the boards rubbing together, the powder acts as a lubricant to help eliminate the sound.	Da
	Q: How can I remove wax from a wooden floor? I dropped a pot of wax on my hardwood floor. How can I remove it without damaging the floor? A: lay a bag of frozen vegetables on the wax wait till frozen and then scrap off with a soft edge tool....repeat till gone....or lay newspaper over the wax and use an iron to heat up the spot the wax transfers to the paper repeat till gone... good luck	Ne
	Q: How do you fix a squeaky hardwood floor? A: the best way to do this is if you have a basement and you can access the floor joists. if you can have someone walk around up stairs so you can find the squeak down stairs. once you find the squeak drive a shim between the sub-floor and joist until the squeak stops.(DON'T DRIVE IT IN TO FAR OR YOU WILL PUT A HUMP IN THE FLOOR) not to hard just time consuming if ya have a lot of squeaks.	Da



Slika 3.1: Sučelje sustava za označavanje relevantnosti.

Relevantnost dokumenta označava se klikom na jednu od ponuđenih oznaka: "DA", "Koristan", "Beskoristan" i "NE". Ove oznake izravno odgovaraju oznakama R, U, X i N, opisanima u prethodnom odjeljku. Nakon što se dokumentu dodijeli oznaka, sustav automatski prikazuje sljedeći neoznačeni dokument. Ako označivač želi revidirati svoju oznaku, može u svakom trenutku dobiti prikaz bilo kojeg dokumenta koristeći navigacijsko sučelje u donjem dijelu ekrana. Oznake se pamte u obliku XML datoteke u poslužiteljskom djelu sustava, pa se označavanje može u bilo kojem trenutku prekinuti te bez gubitka oznaka kasnije nastaviti.

3.4.5. Analiza slaganja označivača

Analizom međusobnog slaganja označivača relevantnosti utvrđeni su neki općeniti uzroci neslaganja koji se često javljaju na ovom skupu podataka. Oni uključuju:

- Nedostatak pažnje – pogreške ove vrste najčešće se javljaju kada *koristan* FAQ-par ima koristan dio informacija iskazan vrlo kratko i neizravno u dugačkom tekstu FAQ odgovora, što označivaču lako promakne;
- Različito razumijevanje PU-ova – u nekim slučajevima postojale su manje razlike među označivačima u interpretaciji informacijske potrebe iskazane PU-om. Ovo je uglavnom uzrokovalo razlike u odluci je li ispravna oznaka *koristan* ili *relevantan*;

Tablica 3.5: Međusobno slaganje označivača na skupu FAQIR za različite sheme interpretacije oznaka.

	R-U-X-N	R-UXN	RU-XN
Početno	0,623	0,545	0,844
Nakon revizije	0,843	0,855	0,908

- Drugačija procjena relevantnosti – subjektivne razlike u procjeni relevantnosti. Na primjer, za PU *"Information on removing bad odor from car"* (*"Uklanjanje neugodnog mirisa iz auta."*) i FAQ-par koji sadrži *"Making your kitchen smell good."* (*"Uvođenje ugodnog mirisa u kuhinju."*), oznake *koristan* i *beskoristan* bi se obje mogle smatrati opravdanima.

Na temelju analize međusobnog slaganja označivača oba označivača su trebala ugoditi svoje razumijevanje PU-ova te izmijeniti svoje odluke na svim parovima upita i FAQ-para na kojima je postojalo neslaganje. Izmjenjivanje nije nužno tražilo izmjenu oznaka, već samo njihovu dodatnu provjeru. Od 101 oznake na kojima se označivači A1 i A2 nisu slagali, označivač A1 je izmijenio 18 oznaka, a označivač A4 je izmijenio 48 oznaka.

Međusobno slaganje označivača izračunato na sve četiri oznake, no u pokusima razmotrena su i dodatna dva načina interpretacije oznaka kao binarnih, tj. kao dva razreda – *relevantno* i *nerelevantno*, kao što je opisano u nastavku.

- *Shema RU-XN* – spaja *R* i U u *relevantno*, dok se ostale oznake smatraju kao *nerelevantno*;
- *Shema R-UXN* – smatra samo *R* oznaku kao razred *relevantno*, a sve ostale oznake kao razred *nerelevantno*.

Rezultati mjerenja međusobnog slaganja označivača dani su u tablici 3.5. Slaganje je izračunato koristeći mjeru Fleissove κ mjere (Fleiss, 1971). Prije izmjena, slaganje u shemi *RU-XN* bilo je mnogo veće nego u shemi *R-UXN*. Ovo opažanje upućuje na to da je mnogo neslaganja postoji upravo između *R* i *U* oznaka. Sva izračunata slaganja postaju razmjerno visoka nakon izmjene, koja je uklonila greške prva dva tipa. Neslaganje koje je ostalo nakon izmjene posljedica je subjektivnosti samih označivača. Kako je ostatak skupa podataka označio samo jedan označivač, izmjena oznaka na cijelom skupu nije bila moguća. Ipak, važno je naglasiti da je slaganje u shemi *RU-XN* visoko i bez provođenja izmjena. Ovo upućuje na to da su, razmatrane kroz ovu shemu, oznake dosljedne na cijelom skupu podataka.

3.5. Skup podataka StackFAQ

U idealnom slučaju predloženi bi postupci trebali biti vrednovani na više različitih skupova podataka kako bi se dodatno potvrdila njihova učinkovitost. Da bi daljnja istraživanja mogla biti provedena na dva različita domenski specifična skupa podataka, poluautomatskim postupkom izgrađen je još jedan skup podataka – StackFAQ. Skup je izgrađen na temelju podataka preuzetih s web-stranice StackExchange.[9] Radi se o stranici s pitanjima i odgovorima temeljenoj na radu zajednice (engl. *community based*). Specifična domena koja je razmatrana za izgradnju ovog FAQ skupa bila je kategorija "web apps" (*"web aplikacije"*). U toj kategoriji korisnici stranice StackExchange postavljaju pitanja pretežito povezana s korištenjem neke od popularnih web-aplikacija, kao što su GMail, Trello ili Facebook.

Prvi korak u izgradnji ovog skupa podataka bio je dohvat dretvi sa stranice StackExchange. Svaka dretva sastoji se od korisničkog pitanja (KP) i niza korisničkih odgovora (KO), koje su samostalno dali drugi korisnici stranice StackExchange. Svaki KO ima pripadni broj glasova (engl. *up-votes*), koji neizravno upućuju na to kolika je kvaliteta odgovora. Velik broj glasova koje je primio neki KO upućuje da on ima veliku korisnost s obzirom na informacijsku potrebu iskazanu kroz KP. Prilikom odabira dretvi koje će biti korištene pri izgradnji u FAQ skupa, odabrane su one koje su imale najveću frekvenciju. U terminologiji stranice StackExchange, frekvencija je definirana kao broj linkova koji upućuju na neku dretvu. Motivacija za ovakav odabir jest ta što je poželjno da FAQ-parovi u izgrađenom FAQ skupu podataka predstavljaju informacijske potrebe koje se zaista često pojavljuju kod stvarnih korisnika u stvarnim slučajevima korištenja. Dodatno, frekventnije dretve tipično sadržavaju veći broj dobrih KO za pripadni KP. Zato je za njih veći broj relevantnih FAQ-parova za informacijsku potrebu izraženu kroz KP, što će biti detaljno opisano u nastavku. Prema opisanom kriteriju, sakupljeno je ukupno 125 dretvi od kojih svaka odgovara nekoj jedinstvenoj informacijskoj potrebi.

U sljedećem koraku, za svaku od 125 dretvi generiraju se FAQ-parovi (Q_i, A_{ij}), gdje je FAQ pitanje Q_i je zapravo KP iz i-te dretve, a FAQ odgovor A_{ij} je zapravo j-ti KO iz i-te dretve. Ovaj postupak generira jedan ili više FAQ parova iz svake dretve, te je ukupno dao 719 FAQ-parova. Valja napomenuti da se svaki KP sastoji od kratkog naslova pitanja i od (tipično dužeg) tijela pitanja, koje je oboje napisao isti korisnik StackExchange platforme. Za FAQ pitanja u ovom skupu podataka koristi se samo naslov KP-a, što je motivirano činjenicom da su FAQ pitanja tipično vrlo kratka. Ipak, korištenje samo naslova KP-a moglo bi uzrokovati da generirano FAQ pitanje bude nepotpuno. Ovakav slučaj bi se dogodio kada bi ključan dio informacijske potrebe bio izražen samo u tijelu KP-a. Kako bi se spriječio ovakav scenarij, ljudski označivač je prije gornjeg postupka pregledao svih 125 dretvi te je, gdje je to bilo potrebno, izmijenio naslov KP-a dodavanjem ključnih informacija iz tijela.

[9] `http://stackexchange.com`

Nakon generiranja FAQ-parova, idući korak jest stvaranje upita koji bi ciljali na informacijske potrebe koje opisuju FAQ-parovi. Ovaj zadatak provodila su dva označivača. Oni su prvo prošli kroz svih 125 dretvi, te za svaku od njih raspravili i usuglasili se oko toga što je točno bila korisnikova informacijska potreba. Cilj ovoga ovog postupka jest osigurati da se između dva označivača neće pojaviti velike razlike u shvaćanju informacijske potrebe izrečene u svakoj pojedinoj dretvi. Označivači su tada, neovisno jedan o drugom, za svaku dretvu tj. informacijsku potrebu, napisali po pet upita. Rezultat ovog koraka je 10 različitih upita za svaku od 125 dretvi, što ukupno čini 1.250 upita. Vrijedi napomenuti da su upiti koji su označivači napisali za istu dretvu zapravo samo parafraze iste informacijske potrebe, što uvodi u skup visoku razinu zalihosti na razini upita, koja je razmatrana kao vrlo poželjno svojstvo u odjeljku 3.1.

Zadnje što je potrebno je povezati upite s relevantnim FAQ-parovima putem oznaka relevantnosti. Neka je q upit stvoren iz dretve i. Skup relevantnih FAQ-parova za upit q definiran je kao skup svih FAQ-parova (Q_i, A_{ij}), takvih da FAQ odgovori A_{ij} (KO-ovi) u njima imaju barem αm glasova. Pri tome je m maksimalan broj glasova koji je dobio bilo koji KO u dretvi i. Svi ostali FAQ-parovi smatraju se nerelevantnima za upit q. Ovaj uvjet zapravo uklanja iz skupa relevantnih FAQ-parova one koji nisu dovoljno kvalitetni, što se za njih odražava kroz premalen broj glasova. Vrijednost praga $\alpha = 0,2$ utvrđena je empirijski kroz ručni pregled dobivenih podataka za različite vrijednosti parametra α. Važno je napomenuti da se svi FAQ-parovi koji su iz dretve različite od one iz koje je stvoren q također smatraju nerelevantnima. Ovo bi moglo rezultirati u krivom označavanju primjera kao nerelevantnih, u slučajevima kada bi dvije različite dretve odnosile na blisko povezane informacijske potrebe. No, činjenica da je svaka od 125 dretvi odabrana tako da se odnosi na jedinstvenu informacijsku potrebu čini ovakvu mogućnost vrlo malo vjerojatnom. Konačno, valja napomenuti da cijeli gore opisani postupak osigurava da za svaki FAQ-par postoji barem jedan korisnički upit za koji je taj upit relevantan.

Konačan izgrađen skup podataka StackFAQ sastoji se od 719 FAQ-parova, 1.250 upita koji su izvedeni iz 125 različitih informacijskih potreba (10 parafraza upita po informacijskoj potrebi) te binarnih oznaka relevantnosti za sve parove upita i FAQ-para. Statistike ovog skupa podataka dane su u tablici 3.6, dok se primjeri iz skupa podataka mogu pronaći u tablici 3.7 Kako bi se potaknula daljnja istraživanja u području pretraživanja FAQ-zbirki, ovaj FAQ skup podataka, kao i kod za njegovo generiranje, javno su dostupni.[10]

3.6. Skup podataka VipFAQ

U predistraživanju pretraživanja FAQ-zbirki za hrvatski jezik korištena je gotova FAQ-zbirka koju je izgradio Žmak (2009). Zbog potpunosti, u ovom radu je detaljno opisan postupak iz-

[10] Skup podataka i programski kod su dostupni na adresi `http://takelab.fer.hr/data/StackFAQ/`. Valja napomenuti da je kod primjenjiv i na StackExchange arhive podataka (engl. *data dump*).

Tablica 3.6: Statistike za skup podataka StackFAQ.

	Minimum	Maksimum	Prosjek	Medijan
Duljina upita (broj riječi)	1	35	13,84	13
Duljina FAQ pitanja (broj riječi)	4	23	10,39	10
Duljina FAQ odgovora (broj riječi)	3	746	76,54	50
Broj relevantnih FAQ-parova po dretvi	1	24	5,75	5

gradnje ovog skupa podataka.

Skup VipFAQ izgrađen je tako što su prvo dohvaćeni FAQ-parovi sa web-stranice velikog hrvatskog mobilnog telekomunikacijskog operatera Vip[11]. Za svaki FAQ-par dohvaćeno je FAQ pitanje i FAQ odgovor. Kako su u Vip FAQ-zbirci pitanja kategorizirana u više širih kategorija (npr., po tipu usluge), za svaki FAQ-par dohvaćeno je i ime pripadne kategorije. Ovim postupkom sakupljeno je ukupno 1.344 FAQ-parova. Nakon uklanjanja duplikata preostala su 1.222 jedinstvena FAQ-para.

U sljedećem koraku deset označivača izmislilo je svaki po dvanaest upita. Označivačima su dane upute da izmisle upite za koje smatraju da bi ih postavili stvarni korisnici Vip usluga. Kako bi se osiguralo da su upiti što originalniji, označivači nisu imali uvid u prikupljene FAQ-parove prilikom provedbe označavanja. Slično kao u (Lytinen i Tomuro, 2002), označivači su parafrazirali svaki osmišljeni upit. Svaki označivač napravio je između tri i pet parafraza za svaki svoj upit. Strategija parafraziranja je pritom obuhvaćala (1) izmjenu upita u višerečenični upit, (2) promjenu sintaktičke strukture upita, (3) zamjenu nekih riječi upita njihovim sinonimima, (4) pretvaranje upita u izjavnu rečenicu i (5) bilo koju kombinaciju prethodnih koraka. Označivačima je posebno naglašeno da je vrlo važno da parafraziranjem *ne mijenjaju* informacijsku potrebu iskazanu u upitu.

Zadnji korak uključivao je označavanje binarnih oznaka relevantnosti za svaku kombinaciju upita i FAQ-para. Nije vremenski izvedivo označiti sve moguće kombinacije, jer ih ima previše. Zbog toga je, slično kao i za skup FAQIR, korištena metoda agregiranja rezultata više postupaka pretraživanja (engl. *pooling technique*) (Manning i dr., 2008). Ova metoda radi tako što se za dani upit ne označavaju svi FAQ-parovi nego samo oni koji su dobri kandidati da budu relevantni. Dobri kandidati definirani su kao oni FAQ-parovi koji se pojavljuju u uniji prvih k rezultata svih postupaka pretraživanja koji ulaze u agregaciju. Ovaj postupak značajno smanjuje broj FAQ-parova koje treba ručno označiti, a za dovoljno velik k postupak će imati zadovoljavajući odziv. Točnije, broj FAQ-parova koji su zaista relevantni, a nisu poslani na

[11] `http://www.vipnet.hr/pitanja-i-odgovori/` (datum pristupa: rujan, 2009.)

Tablica 3.7: Primjeri iz skupa podataka StackFAQ.

Upit	FAQ-par	Relevantno?
I'd like to completely remove my fb account, how do I do this?	Q: How do I delete my Facebook account? A: Go to "Your Account". Go to "Deactivate Account". Facebook is a megalomaniac. It will never ever ever EVER delete your profile. If you decide ten years later to sign in, then it will still be there. I'd erase my pictures from the site before deleting my account. Actually, I just did last week.	Da
	Q: Determine who unfriended me on Facebook A: Intentionally, Facebook prevents sending out this info for privacy purposes. You are free to keep track of your friends though in some other form, and compare it to the current list to find out who is missing each week. who.removed.me will do this for you automatically	Ne
	Q: How to remove Dropbox access from a computer you no longer have? A: You can manage and revoke access to your computers from your control panel at the following link. https://www.dropbox.com/account/settings#security	Ne
Can I be part of a google mailing list without a gmail address?	Q: How can I subscribe to a Google mailing list with a non-Google e-mail address? A: Procedure to join Google Groups without creating a Google account: Look for group email address in the heading of "Group email", it will look like: Group email (Group Name)@googlegroups.com or visit the group web page on Google Groups, and click on "About this group". Send an email to (Group Name)+subscribe@googlegroups.com . +subscribe is the key, which is appended to the end of group's email address. (Group Name) is the name of the group. Space is substituted by "-" (hyphen). Source . (gone) Note: You should send the subscribing email from the email address which you want to receive the messages mailing. You may need to confirm the subscription request for some groups.	Da
	Q: Can I see only mail I have archived in Gmail? A: If you search for the following, only the archived emails should show up in the list: -in:Sent -in:Chat -in:Draft -in:Inbox As dzilbers pointed out, if you have "group messages by conversation" enabled in your settings, some emails with those labels may still show up in the results if one of the messages in the conversation is archived. You will still have the "Move to Inbox" option for those results, which will un-archive any archived emails in the conversation.	Ne

Tablica 3.8: Primjeri upita iz skupa podataka VipFAQ i pripadnih relevantnih FAQ-parova.

Upit	FAQ pitanje	FAQ odgovor
Kako se spaja na internet?	Što mi je potrebno da bih spojio računalo i koristio se internetom?	Morate spojiti računalo sa Homebox uređajem LAN kabelom...
Putujem izvan Hrvatske i želim koristiti svoj Vip mobilni uređaj. Koliko će me to koštati?	Koja je mreža najpovoljnija za razgovore, a koja za slanje SMS i MMS poruka u roamingu?	Cijene za odlazne pozive u inozemstvu su najpovoljnije u mrežama Vodafone partnera...
Kako pogledati e-mail preko mobitela?	Koja je cijena korištenja BlackBerry Office usluge?	... business e-mail usluga uračunata je u cijenu...

Tablica 3.9: Statistike skupa podataka VipFAQ.

	Broj riječi			Oblik	
	Minimum	Maksimum	Prosjek	Upitno	Deklarativno
Upiti	1	25	8	372	47
FAQ upiti	4	63	7	287	4
FAQ odgovori	1	218	30	–	–

označavanje kao dobri kandidati, bit će malen. Ovo se događa zato što je mala vjerojatnost da *relevantan* FAQ-par ne završi u prvih k rezultata *niti jednog* postupka pretraživanja, te tako ne bude među dobrim kandidatima. Kao postupci pretraživanja u agregaciji korišteni su: (1) pretraživanje po ključnim riječima, (2) pretraživanje po frazama, (3) postupak tf-idf vektorskog prostora i (4) jezični model. Opisi svih ovih postupaka mogu se pronaći u odjeljku 2.2. Broj FAQ-parova koji su bili dobri kandidati i bili označeni[12] je između 50 i 150 po upitu. Kako bi se smanjila pristranost koju bi označivači mogli imati prema onim FAQ-parovima koji su bili rangirani bolje u rezultatima postupaka pretraživanja, FAQ-parovi su označivačima bili prikazivani slučajnim redoslijedom. Za svaki par upita i FAQ-para koji je dobar kandidat, označivači su odabrali jednu od oznaka *relevantan* ili *nerelevantan*. FAQ-parovi koji za neki upit nisu u skupu dobrih kandidata automatski su bili označeni oznakom *nerelevantan*. Iako je prikladnost binarnih oznaka relevantnosti u literaturi dovedena u pitanje, npr., u Kekäläinen (2005), binarne se oznake i dalje vrlo često koriste za FAQ i slične skupove (Wu i dr., 2006; Voorhees i Tice, 2000). Primjeri upita i relevantnih FAQ-parova iz zbirke mogu se pronaći u tablici 3.8.

Opisani postupak stvara skup parova (Q_r, F_{rel}), gdje je Q_r skup parafraza nekog izmišlje-

[12] Broj je to veći što se popisi prvih k dohvaćenih FAQ-parova više razlikuju za različite postupke pretraživanja.

nog upita tj. neke informacijske potrebe, a F_{rel} skup relevantnih FAQ-parova tu informacijsku potrebu. Ukupan broj takvih parova je 117. Iz ovog skupa generiran je skup parova (q, F_{rel}), gdje je $q \in Q_r$ pojedini upit. Ukupan broj takvih parova je 419, od kojih 327 imaju barem jedan relevantan odgovor ($F_{rel} \neq \emptyset$), dok 92 nisu odgovoreni ($F_{rel} = \emptyset$). Prosječan broj relevantnih FAQ-parova po pojedinom upitu je 1,26, dok je u prosjeku svaki pojedini FAQ-par relevantan za 1,44 upita. Dodatne statistike ovog skupa podataka prikazane su u tablici 3.9. Skup podataka javno je dostupan za istraživačke svrhe.[13]

Opisom ovog skupa podataka završen je opis svih skupova podataka korištenih u ovom radu. U nastavku rada će biti opisana predistraživanja na nekim od ovih skupova u poglavljima 4., 5. i 6. Potom slijedi opis glavnog dijela istraživanja.

[13] Dostupno pod CC BY-SA-NC licencijom na `http://takelab.fer.hr/faqirHR`

Poglavlje 4.

Pravila za proširenje upita

4.1. Motivacija i opis problema

FAQ-zbirke pružaju mnoge praktične prednosti. Zato je važan zadatak izgraditi kvalitetne modele pretraživanja informacija nad njima. Zadatak pretraživanja FAQ-zbirki (engl. *FAQ retrieval*) svodi se na dohvat najrelevantnijeg FAQ-para za zadani upit korisnika. Glavni izazov ovog zadatka jest to što su tekstovi kratki i domenski specifični, što povećava vjerojatnost leksičkog jaza (Berger i dr., 2000; Lee i dr., 2008). Na primjer, upit *"Can't connect to the net"* (*"Ne mogu se spojiti na internet."*) bi se trebao preslikavati na pitanje *"Why is my internet down?"* (*"Zašto mi internet ne radi?"*), iako je broj zajedničkih riječi između ta dva teksta vrlo malen.

U literaturi su predloženi brojni sustavi za pretraživanje FAQ-zbirki. Primjeri se mogu naći u (Burke i dr., 1997; Surdeanu i dr., 2008; Sneiders, 2009). Jedan od naprednijih je model predložen u (Surdeanu i dr., 2008), koji kombinira značajke temeljene na semantičkoj sličnosti, strojnom prevođenju, frekvenciji riječi i korelaciji s rezultatima web-tražilica s modelom nadziranog strojnog učenja, te na taj način premošćuje leksički jaz. Za uspješno učenje ovog modela potreban je velik skup podataka.

Alternativan, manje zahtjevan način za rješavanje problema leksičkog jaza jest proširenje upita (engl. *query expansion* – QE). Glavna ideja tog pristupa jest proširenje upita dodatnim riječima koje bi trebale povećati izglednost ispravnog dohvata relevantnih dokumenata. Jedna varijanta QE jest QE temeljen na pravilima za proširenje upita (QE-pravila), u kojem je svaka riječ povezana s popisom povezanih riječi (najčešće sinonimi ili riječi sličnog značenja). Kada se neka riječ pojavi u upitu, primjenjuje se pripadno QE-pravilo, te se povezane riječi s pripadnog popisa automatski nadodaju u korisnički upit. U okviru ovog predistraživanja naglasak je na razmatranju QE temeljenog na pravilima.

Carpineto i Romano (2012) daju pregled postupaka za pronalaženje dobrih QE-pravila za velike FAQ-zbirke. Dodatna istraživanja u vezi automatiziranog učenja QE-pravila mogu se naći u (Wei i dr., 2000; Latiri i dr., 2003). Svi predloženi postupci primjenjivi su na veliku

FAQ-zbirku usredotočenu na općenitu domenu, što im ograničava primjenjivost u slučaju malih domenski specifičnih FAQ-zbirki. Sneiders (2009) pristupa problemu kroz definiciju "upitnih predložaka" koji pokrivaju ontološku, leksičku, morfološku i sintaktičku varijaciju moguću u upitima korisnika koji ciljaju na pojedinu informacijsku potrebu. Ovakvi upitni predlošci mogli bi se smatrati naprednijom varijantom QE-pravila. Predlošci su definirani ručno, nakon analize korisničkih upita i klikova bilježenih tijekom više mjeseci.

Budući da su srodna istraživanja (Carpineto i Romano, 2012; Sneiders, 2009) pokazala da je QE općenito koristan za pretraživanje informacija, u ovom istraživanju razmatra se pretpostavka da je to također slučaj i za zadatak domenski specifičnog pretraživanja FAQ-zbirki. Konkretno, naglasak je na slučajevima gdje nema unaprijed poznatih oznaka relevantnosti, te je veličina FAQ-zbirki malena. Ovo onemogućava primjenu naprednijih postupaka za izgradnju QE-pravila, koji bi koristili nadzirano strojno učenje. S druge strane, kako FAQ-zbirka nije prevelika, ručna izgradnja QE-pravila je praktično izvediva. Pokusi su provedeni na FAQ skupu podataka izgrađenom za potrebe ovog istraživanja, koji je opisan u odjeljku 3.3. Istraživačka pitanja kojima se bavi ovo poglavlje su: (1) do koje mjere može optimalan skup QE-pravila povećati učinkovitost sustava za pretraživanje FAQ-zbirki i (2) mogu li ljudski označivači napraviti skup domenski specifičnih QE-pravila sumjerljiv po kvaliteti s optimalnim pravilima. U nastavku ovog poglavlja slijedi detaljan opis postupaka za pronalaženje QE-pravila koji su razmatrani u ovom predistraživanju, te rezultati vrednovanja na domenski specifičnom Verizon-FAQ skupu podataka na engleskom jeziku.

4.2. Opis istraženih postupaka

4.2.1. Postupci za pretraživanje

Kao postupke pretraživanja koji se koriste za dohvat prije ili nakon primjene proširenja upita koristimo dva temeljna postupka za pretraživanje informacija.

1. **BM25 model** (Robertson i dr., 1995), temeljen na modelu izglednosti upita (Manning i dr., 2008), koji je detaljno opisan u odjeljku 2.2.2. Iako jednostavan za implementaciju, ovaj model se pokazao kao vrlo učinkovit.
2. **Model skip-gram (SG)**, koji se temelji na prikazima teksta kao zbroja semantičkih vektora sadržajnih riječi u tekstu. U ovom dijelu istraživanja korišteni su vektori izgrađeni u (Mikolov i dr., 2013).[1] Postupak za pretraživanje informacija temelji se na usporedbi korisničkog upita i FAQ-para preko kosinusne sličnosti. Detaljan opis ovog modela može se pronaći u odjeljku 2.2.

Prije primjena ovih postupaka radi se jednostavna predobrada tekstova koja uključuje uklanjanje

[1] `http://code.google.com/p/word2vec/`

zaustavnih riječi i korjenovanje riječi korištenjem Porterova algoritma za korjenovanje (Porter, 2001).

4.2.2. Pravila za proširenje upita

U ovom dijelu istraživanja naglasak je na QE-pravilima koja su ručno izrađena prilikom izgradnje same FAQ-zbirke. To znači da konkretan korisnički upit nije poznat u trenutku izrade QE-pravila, pa se stoga ne može uzimati u obzir za prilagodbu pravila. Postoji cijela porodica postupaka za generiranje QE-pravila koji ne uzimaju u obzir korisnički upit, već generiraju generička pravila koja općenito dobro rade za sve upite. Ova porodica postupaka često se u literaturi javlja pod nazivom "globalni" postupci za proširenje upita (Xu i Croft, 1996).[2] Postupci za pronalaženje QE-pravila predloženi u ovom istraživanju također spadaju u ovu porodicu. Konkretno, QE se provodi tako da se gradi globalni skup QE-pravila. Pojedino QE-pravilo definirano je kao skup riječi $R = \{r_1, \ldots, r_n\}$ takav da, ako se bilo koja riječ iz R pojavi u korisničkom upitu, sve ostale riječi iz R će automatski biti dodane u upit. Razmatramo način ručne izrade domenski specifičnih QE-pravila, gdje za zadanu riječ ljudski označivač navodi riječi s kojima bi se ona mogla proširiti. Ovaj način izgradnje QE pravila uspoređujemo sa zlatnim pravilima, koja predstavljaju skup pravila koji maksimizira učinkovitost modela pretraživanja na ispitnom skupu. Cilj je odrediti učinkovitost QE-pravila općenito za FAQ pretraživanje, te koliko se ručno napravljenim domenski specifičnim pravilima možemo približiti empirijskom maksimumu performansi koji dostižu zlatna pravila. Sva potrebna označavanja i vrednovanja opisana u nastavku ovog poglavlja provedena su na skupu podataka VerizonFAQ, koji je opisan u odjeljku 3.3.

Zlatna QE-pravila. Kako bi se vrednovala učinkovitost optimalnog skupa QE-pravila, napravljen je skup zlatnih pravila. Izrada pravila provedena je tako što su ljudski označivači analizirali pogreške na ispitnom skupu te su ručno osmislili pravila koja premošćuju problematičan leksički jaz, na mjestima gdje je to bilo moguće. Ovakav skup zlatnih pravila je optimalan u smislu da je napravljen tako da maksimizira performanse modela pretraživanja na ispitnom skupu. Razumno je pretpostaviti da zlatna QE-pravila u ovom slučaju ne ovise o specifičnom modelu pretraživanja koji se koristi. Zbog toga je ovaj korak proveden samo za BM25 model, te su tako dobivena pravila u daljnjim pokusima korištena bez promjena za sve modele. Iako rezultati pokusa, opisani u odjeljku 4.3., govore u prilog ovoj pretpostavci, potrebno je provesti detaljnije pokuse kako bi ona bila u potpunosti potvrđena.

[2] Alternativa su "lokalni" postupci koji dodatno prilagođavaju QE-pravila svakom pojedinom korisničkom upitu.

Domenski specifična QE-pravila. U sklopu istraživanja izgrađen je i skup QE-pravila koji su također izradili ljudski označivači, ali bez razmatranja pogrešaka koje je sustav radio, već korištenjem svog domenskog znanja. Dakle, trebalo je osmisliti pravila koja su općenito korisna (za sve upite) u specifičnoj domeni ovog skupa podataka. Označivači su kao zadatak dobili riječ, te su morali donijeti dvije odluke. Prvo, je li dana riječ prikladna da se bude dio QE-pravila. Drugo, ako je prethodna odluka bila pozitivna, navesti skup prikladnih (po svom subjektivnom mišljenju) *riječi kandidata*, s kojima bi se dana riječ trebala proširiti. Na primjer, za zadanu riječ *pay*, označivač bi mogao navesti riječi kao što su *charge*, *rate*, *cost*, itd. Ove sve riječi bi kao skup tvorile jedno označeno QE-pravilo.

Jedan zanimljiv problem kod ovog postupka označavanja jest odabir riječi koje ćemo zadati označivačima. Točnije, kako iz skupa svih mogućih riječi odabrati baš one koje će kroz označavanje stvoriti korisna QE-pravila. Nakon razmatranja više različitih strategija, pokazalo se učinkovitim jednostavno rangirati riječi po frekvenciji u skupu podataka. Korišteno je ukupno 300 (broj koji je bio razuman za označivače) riječi s najvećom frekvencijom. Ovaj skup riječi pokriva 95% riječi koje se nalaze u zlatnim pravilima.

Tri označivača neovisno su izgradili skupove riječi kandidata. Razmotreno je više načina za kombiniranje ovih skupova u jedinstven skup. Kao najbolji odabir pokazala se operacija presjeka. Na ovaj način dobivena su 24 QE-pravila. Primjeri pravila bili bi skupovi kao što su:

- $\{broadband, internet, connection\}$;
- $\{customer, buyer, client\}$;
- $\{number, phone, telephone, mobile, digit\}$.

Domenski specifična QE-pravila moguće je izravno usporediti sa zlatnim pravilima uspoređujući konkretne riječi u pravilima. Usporedbu je također moguće napraviti neizravno, kroz razmatranje utjecaja pojedinih vrsta pravila na učinkovitost sustava za pretraživanje FAQ-zbirki. U ovom istraživanju, provedene su obje vrste vrednovanja, kako je opisano u nastavku.

4.2.3. Implementacija

Svi postupci i mjere vrednovanja u potpunosti su implementirani u programskom jeziku C# koristeći radni okvir Mono,[3] koji je implementacija Microsoft .NET radnog okvira temeljena na otvorenom kodu i prikladna za korištenje na operacijskom sustavu Unix. Za manje dijelove ovog istraživanja, kao npr. preuzimanje podataka s Verizon web-stranice, korišten je i programski jezik Python. Nisu korištene nikakve dodatne vanjske biblioteke.

[3] `https://github.com/mono`

4.3. Vrednovanje

4.3.1. Vrednovanje ispravnosti pravila

Tijekom označavanja, označivači su mogli napraviti dvije vrste pogreške. Prvo, mogli su krivo procijeniti je li neka riječ prikladna za proširenje. Drugo, za one riječi koje jesu bile prikladne, mogli su ne navesti one riječi kandidate koje bi bile najkorisnije. U ovom dijelu vrednovanja, za svakog označivača razmatramo oba aspekta: sposobnost označivača da odredi prikladnost riječi za proširenje i kvalitetu navedenih riječi kandidata.

Za prvi aspekt, razmatramo koje riječi u skupu zadanih riječi je označivač odlučio proširiti. One koje se pojavljuju u nekom od zlatnih QE-pravila, smatraju se stvarno pozitivno označenim primjerima. Sve riječi koje se ne pojavljuju u zlatnim QE-pravilima, a označivač ih je proširio, smatraju se lažno pozitivnim primjerima. Konačno, riječi koje označivač nije proširio, a nalaze se u zlatnim QE-pravilima smatraju se lažno negativnim primjerima. Na temelju ovakve interpretacije računa se mjera F_1, koja je opisana u odjeljku 2.4.

Za vrednovanje drugog aspekta, svako QE-pravilo (skup riječi kandidata za neku riječ) smatra se grupom riječi. Tako nam skup QE-pravila zapravo prikazuje grupiranje (engl. *clustering*) riječi. Na isti način možemo i zlatna pravila shvatiti kao grupiranje riječi. Ova dva grupiranja možemo usporediti nekom od mjera za evaluaciju grupiranja. U ovom slučaju koristi se mjera F_1 na razini uparenih binarnih odluka. Ova mjera uzima u obzir samo riječi koje se pojavljuju u zlatnim pravilima, jer je naglasak na kvaliteti dobivenih proširenja za upravo te riječi.[4]

Rezultati ovog vrednovanja dani su u tablici 4.1. Za zadatak određivanja prikladnosti riječi za proširenje preciznost je vrlo niska. Ovo je očekivano, jer su zlatna pravila izgrađena tako da rješavaju vrlo specifične problematične slučajeve na ispitnom skupu. Posljedica toga je da velik broj pravila, koja bi u općenitom slučaju bila korisna, nisu uključena u zlatna pravila. S druge strane, za dvoje od troje označivača preciznost je iznimno visoka. Ovaj rezultat upućuje na to da su označivači uspjeli navesti QE-pravila za većinu riječi za koja postoje QE-pravila u zlatnom standardu. No, pritom su također naveli i velik broj QE-pravila za riječi za koje u ovom konkretnom slučaju to nije bilo potrebno,

Razmatranjem grupa riječi koje predstavljaju QE-pravila pokazuje se drugačija slika. Preciznost je vrlo visoka, što upućuje na to da su označivači navodili ispravna proširenja. No, odziv je prilično nizak, što znači da navedena proširenja nisu bila potpuna. Iz ovog rezultata može se vidjeti da označivači često propuštaju navesti manje intuitivne, važne riječi, čija prisutnost u pravilima bi pokrila neki vrlo netipičan slučaj problema leksičkog jaza.

[4]Označivači su mogli dati dobre skupove riječi kandidata za zadane riječi koje nisu dio zlatnih pravila, jer na ispitnom skupu to nije bilo potrebno. Za takve skupove ne postoje skupovi u zlatnim pravilima s kojima bi ih se moglo usporediti. Zato bi oni, iako potencijalno točni, uvijek "umjetno" snižavali mjeru F_1 po parovima.

Tablica 4.1: Ispravnost QE-pravila u smislu odabira prikladnih riječi za proširenje (lijevo) i kvalitete navedenih riječi kandidata za proširenje (desno).

	Prikladnost riječi			Kvaliteta proširenja		
	P	R	F_1	P	R	F_1
Označivač 1	8,2	95,0	**15,1**	75,0	9,4	16,6
Označivač 2	6,2	65,0	11,3	85,0	9,4	16,9
Označivač 3	6,9	95,0	12,9	62,5	15,6	**25,0**

4.3.2. Vrednovanje pretraživanja

Kako bi se odredio potencijal dobivenih QE-pravila u smislu poboljšanja učinkovitosti pretraživanja FAQ-zbirki, provedeno je vrednovanje modela pretraživanja sa QE-pravilima i bez njih. Vrednovanje je provedeno na skupu podataka VerizonFAQ, opisanom u odjeljku 3.3. Kao mjere vrednovanja sustava koristi se srednji recipročni rang (engl. *mean reciprocal rank* – MRR) i odziv na rangu 1 (R@1), koji su opisani u odjeljku 2.4. Vrednovanje je provedeno za oba modela – BM25 i SG – te obje strategije izgradnje QE pravila – zlatna pravila i domenska pravila. Kao tekst u FAQ-paru isprobane su varijante gdje se razmatra samo FAQ pitanje ("postav Q") i varijanta gdje se razmatra spojen tekst FAQ pitanja i FAQ odgovora ("postav QA").

Učinkovitost modela na cijelom FAQ skupu podataka bila ja iznimno visoka, što je možebitno maskiralo pozitivne efekte QE-pravila. Zbog toga je stvorena "teška" varijanta FAQ skupa podataka tako što su ostavljeni samo oni upiti za koje relevantan FAQ-par nije bio rangiran na prvom mjestu (primjerice, treći upit iz tablice 3.1). Ovo je napravljeno zasebno za sve kombinacije modela (BM25 / SG) i postava (Q / QA). Rezultati su prikazani u tablici 4.2

Bez QE-pravila, najučinkovitiji je model BM25. Pritom je model SG usporediv samo u postavu Q, ali radi dosta slabije u postavu QA. Razlog za ovo bi moglo biti to što u potonjem slučaju velik broj riječi ulazi u izračun vektorskog prikaza FAQ-para, što predstavlja šum.

QE-pravila općenito smanjuju učinkovitost modela pretraživanja ako se razmatra cijeli skup upita. Zlatna pravila bila su napravljena za BM25 model i postav Q te, očekivano, tamo donose poboljšanja. Zanimljivo je primijetiti da zlatna QE-pravila pomažu i na nekim drugim teškim upitima za koje nisu posebno izgrađena.[5] Domenski specifična pravila rade nešto slabije od zlatnih pravila, što je u skladu s očekivanjima. Na teškim upitima, na kojima zlatna pravila nisu eksplicitno građena, razlika između zlatnih i domenski specifičnih pravila manje je izražena. Općenito, sva pravila lagano kvare mjeru MRR, ali popravljaju mjeru R@1.

[5]Treba napomenuti da rezultati na teškim skupovima nisu sasvim usporedivi, jer za različite kombinacije model/postav skupovi težih upita ne moraju biti identični.

Tablica 4.2: Rezultati pretraživanja informacija za sve modele i postave (MRR/R@1).

	postav Q		postav QA	
	Svi upiti	Teški upiti	Svi upiti	Teški upiti
BM25	92,9/88,9	38,7/0,0	90,4/84,6	38,0/0,0
BM25 + zlatna pravila	93,2/89,5	43,1/10,6	88,4/81,5	39,9/3,6
BM25 + domenski specifična pravila	92,6/88,4	36,8/2,5	90,0/84,2	39,5/4,5
SG	89,5/86,1	25,0/0,0	79,5/71,1	29,0/0,0
SG + zlatna pravila	88,0/83,6	27,8/3,5	77,8/69,1	31,8/5,2
SG + domenski specifična pravila	86,6/82,4	23,7/1,0	74,8/66,0	28,0/4,0

4.4. Rasprava

U pokusima na domenski specifičnom FAQ skupu podataka, domenski specifična pravila za proširenje upita (engl. *query expansion* – QE) su, suprotno intuiciji, malo narušila performanse. U slučaju kada su razmatrani samo teži upiti, rezultati pokazuju da QE-pravila ipak pomažu, posebno u kontekstu mjere R@1. Glavni uzroci ovakvih rezultata mogu se pronaći u nedovoljno jako parafraziranim upitima prilikom izgradnje skupa podataka, što je detaljnije diskutirano u odjeljku 3.3. Ovakav skup podataka pogoduje jednostavnim modelima poput BM25, što se i potvrdilo kroz iznimno visoke rezultate tog modela. Kroz analizu pogrešaka, pokazuje se da one pogreške koje modeli pretraživanja ipak naprave uopće nisu posljedica leksičkog jaza, već složenijih problema (npr. za ispravan dohvat je potrebno provesti logičko zaključivanje ili imati vrlo specifično domensko znanje). Za bolju potvrdu dobivenih rezultata potrebni su daljnji pokusi na drugim, izazovnijim FAQ skupovima podataka.

Poglavlje 5.

Osnovni postupci pretraživanja zbirki pitanja i odgovora za engleski jezik

5.1. Motivacija i opis problema

Skup podataka VerizonFAQ, opisan u poglavlju 3.3., pokazao se kao nedovoljno izazovan za istraživanje postupaka pretraživanja FAQ-zbirki temeljenih na nadziranome strojnom učenju. Zbog toga je izgrađen FAQIR skup podataka, opisan u poglavlju 3.4. U ovom poglavlju opisano je predistraživanje postupaka za pretraživanje FAQ-zbirki na ovom skupu, provedeno kako bi se osiguralo da je on prikladan za daljnje istraživanje ovakvih postupaka. Za razliku od prethodnog poglavlja, nisu razmatrani postupci za proširenje upita, već samo osnovni postupci za pretraživanje. Postupci za pretraživanje FAQ-zbirki istraženi u ovom poglavlju ujedno čine i temeljne postupke s kojima će se uspoređivati napredniji postupci za pretraživanje FAQ-zbirki razmatrani u nastavku ovog istraživanja, opisanom u poglavlju 9.

5.2. Opis istraženih postupaka

Upiti i FAQ-parovi su predobrađeni tako što su velika slova pretvorena u mala te su uklonjene sve riječi kraće od tri znaka i riječi u kojima barem jedan znak nije bio slovo ili broj. Kao temeljni postupci pretraživanja informacija, pomoću kojih su dobiveni preliminarni rezultati na ovom skupu podataka, korišteni su:

- BM25 (Robertson i dr., 1995) – detaljno opisan u odjeljku 2.2.2. Jednostavan model koji se temelji na leksičkom preklapanju između upita i FAQ-para, ali pokazuje vrlo dobre rezultate;
- VS – klasičan model pretraživanja informacija temeljen na vektorskom prostoru dokumenata koji koristi shemu težina tf-idf. Ovaj model je detaljno opisan u odjeljku 2.2.1.;
- SG (skip-gram) – model koji se temelji na semantičkoj kompozicionalnosti riječi u tekstu

	postav Q			postav A			postav QA		
	MAP	MRR	P@5	MAP	MRR	P@5	MAP	MRR	P@5
BM25	0,204	0,518	0,253	0,121	0,362	0,173	**0,203**	**0,527**	0,268
VS	0,197	0,523	0,247	0,092	0,306	0,156	0,168	0,477	0,244
W2V	0,183	0,504	0,249	0,115	0,370	0,175	0,161	0,474	0,247
Kombinacija	**0,208**	**0,548**	**0,285**	**0,119**	**0,387**	**0,194**	0,184	0,518	**0,282**

Tablica 5.1: Rezultati pretraživanja za *R-UXN* shemu.

	postav Q			postav A			postav QA		
	MAP	MRR	P@5	MAP	MRR	P@5	MAP	MRR	P@5
BM25	**0,163**	0,613	0,355	**0,103**	0,481	0,251	**0,166**	0,638	0,375
VS	0,159	0,615	0,352	0,077	0,412	0,222	0,140	0,579	0,357
W2V	0,139	0,593	0,341	0,097	0,486	0,263	0,130	0,588	0,352
Kombinacija	**0,163**	**0,645**	**0,396**	0,102	**0,515**	**0,284**	0,150	**0,642**	**0,410**

Tablica 5.2: Rezultati pretraživanja za *RU-XN* shemu.

te predstavlja tekstove upita i FAQ-para kao zbroj semantičkih vektora sadržajnih riječi. U ovom istraživanju korišteni su slobodno dostupni[1] semantički vektori riječi koje je izgradio Mikolov i dr. (2013). Identično kao u (Šarić i dr., 2012), zbroj vektora je utežan težinama definiranim preko informacijskog sadržaja (engl. *information content*) svake riječi, zato da bi se naglasio utjecaj onih riječi koje su informativnije;

- Kombinacija – združeni postupak u kojem se rangirana lista FAQ-parova oblikuje tako da se za svaki FAQ-par izračuna zbroj rangova koje on ima u rezultatu prethodna tri modela. Potom se FAQ-parovi poredaju uzlazno po izračunatom zbroju.

Slično kao i dio istraživanja opisan u prethodnom poglavlju, ovi postupci pretraživanja i mjere vrednovanja u potpunosti su implementirani u programskom jeziku C# koristeći radni okvir Mono, bez vanjskih biblioteka.

5.3. Vrednovanje

Provedeno je vrednovanje temeljnih postupaka za pretraživanje informacija na skupu podataka FAQIR, kako bi se odredili preliminarni rezultati na ovom skupu, te potvrdilo da je on prikladan za daljnja istraživanja pretraživanja FAQ-zbirki. Temeljni postupci su vrednovani u shemama oznaka *R-UXN* i *RU-XN*, koje su detaljnije opisane u poglavlju 3.4. Dobiveni rezultati prikazani su u tablicama 5.1 i 5.2. Dodatno, razmotrena su tri načina interpretacije FAQ-para: (1) samo pitanje (postav Q), (2) samo odgovor (postav A) te (3) združen tekst pitanja i odgovora (postav QA). Kao mjere vrednovanja koriste se klasične mjere za pretraživanje informacija: srednja prosječna preciznost (engl. *mean average precision* – MAP) i srednji recipročni rang (engl. *mean reciprocal rank* – MRR), opisan u odjeljku 2.4. Nadalje, koristi se i mjera preciznosti na rangu pet (engl. *precision at rank five* – P@5). Ta mjera je posebno prikladna za ovaj zadatak, jer bi u realističnom slučaju korištenja korisnik očekivao da relevantan odgovor na zadani upit bude u prvih pet dohvaćenih FAQ-parova. Određena ponašanja mogu se dosljedno opaziti u svim pokusima.

Najbolji model Kombinirani postupak općenito ili ima najbolji rezultat ili je vrlo blizu najboljem rezultatu. Ovo upućuje na to da pojedini postupci pretraživanja ne rade iste pogreške, pa je stoga njihova kombinacija korisna. Složeniji način kombiniranja postupaka, kao što je nadzirano strojno učenje rangiranja (engl. *learning to rank*) (Liu i dr., 2009), možda bi mogao dati još bolje rezultate.

FAQ pitanje i FAQ odgovor Učinkovitost u postavu A dosljedno je osjetno slabija nego u postavu Q i postavu QA. Ovo upućuje na to da je usporedba korisničkog upita s pitanjem u FAQ-paru korisnija nego usporedba s odgovorom u FAQ-paru. Između postava Q i postava QA postoje samo blage razlike u učinkovitosti modela pretraživanja. Ipak, moglo bi biti korisno da se, kod usporedbe s korisničkim upitom, pitanje i odgovor u FAQ-paru razmatraju sasvim odvojeno. Potom bi se obje dobivene sličnosti mogle združeno koristiti za pretraživanje FAQ-zbirki.

Usporedba shema Ako razmatramo MRR i P@5, rezultati za shemu *RU-XN* dosljedno su bolji nego za shemu *R-UXN*. Ovo nije iznenađujuće, jer je shema *RU-XN* općenito manje stroga oko toga što se smatra relevantnim. Zbog toga, visoko rangirani dokument koji ima oznaku *U* umjesto *R* će u shemi *RU-XN* biti smatran ispravnim, dok u *R-UXN* neće. Dodatno, stvarnog korisnika bi zanimali FAQ-parovi označeni i sa oznakom *R* i sa oznakom *U*, pa se može smatrati da oznake prema shemi *RU-XN* daju realističnije vrednovanje. Za mjeru MAP situacija je obrnuta, te su rezultati dosljedno bolji za shemu *R-UXN*. Uzrok ove pojave je to što mjera MAP,

[1] `https://code.google.com/p/word2vec/`

umjesto da samo razmatra vrh popisa dohvaćenih FAQ-parova, razmatra cijeli popis. Posljedica toga je da u oznakama prema shemi *RU-XN* i *R* i *U* označeni FAQ-parovi moraju *svi* biti visoko rangirani. Ovo zadatak rangiranja čini težim nego što je slučaj kod oznaka prema shemi *R-UXN*, gdje je dovoljno da visoko rangirani budu samo oni FAQ parovi koji su označeni oznakom *R*.

5.4. Rasprava

Provedeno je predistraživanje kako bi se utvrdila učinkovitost više temeljnih postupaka za pretraživanje informacija te prikladnost ovog skupa podataka za daljnje korištenje u istraživanju. Razmotreno je kako pojedini dijelovi FAQ-para (pitanje i odgovor) utječu na rezultate. Dodatno, razmotreno je kakav je utjecaj različitih shema označavanja na međusobno slaganje označivača, kao i na rezultate pretraživanja. Dobiveni rezultati su očekivani te pokazuju da je skup izazovan te čini dobru osnovu za daljnja istraživanja postupaka za pretraživanje FAQ-zbirki.

Skup FAQIR izvrsna je početna točka za istraživanje nadziranih modela strojnog učenja za pretraživanje FAQ-zbirki. No, ostaje otvoreno pitanje da li rezultati dobiveni na skupu FAQIR vrijede i u drugim domenama, ili za druge jezike. Kako bi se barem djelomično odgovorilo na ovo pitanje, usporedno sa skupom FAQIR u većini pokusa opisanih u nastavku rada koristi se dodan skup podataka, stackFAQ, opisan u odjeljku 3.5.,

Poglavlje 6.

Osnovni postupci pretraživanja zbirki pitanja i odgovora za hrvatski jezik

6.1. Motivacija i opis problema

U ovome poglavlju nastavljamo opis predistraživanja pretraživanja FAQ-zbirki. Bit će opisan postupak za pretraživanje FAQ-zbirki razvijen za hrvatski jezik na skupu podataka VipFAQ. Cilj ovog istraživanja bio je određivanje potencijala nadziranog strojnog učenja za pretraživanje FAQ-zbirki na hrvatskome jeziku. Jedini prijašnji rad koji se bavi ovim zadatkom je (Žmak, 2009), gdje je za postupak pretraživanja korištena linearna kombinacija četiri mjera usporedbe teksta: (1) preklapanja riječi, (2) sličnosti teksta na temelju tf-idf vektora, (3) usporedbe riječi u tekstu na temelju semantičkog grafa te (4) poklapanja tipa pitanja za korisnički upit i FAQ pitanje. Za razliku od (Žmak, 2009), u ovom se radu istražuje širi skup značajki inspiriranih rezultatima (Agirre i dr., 2012; Šarić i dr., 2012). One se koriste u modelu nadziranog strojnog učenja iz radnog okvira učenja rangiranja (engl. *learning to rank*) (Liu i dr., 2009)).

6.2. Opis istraženog postupka za pretraživanje

Zadaća postupka za pretraživanje jest rangiranje FAQ-parova po relevantnosti za korisnički upit. Postupak pretraživanja koji se koristi u ovom predistraživanju temelji se na pouzdanosti klasifikacije binarnog klasifikacijskog modela. Takav model uči na binarnim oznakama relevantnosti, kakve su dostupne u skupu podataka VipFAQ. Ulaz u klasifikacijski model su parovi korisničkog upita i FAQ-para predstavljeni kao vektor značajki koje modeliraju različite vrste semantičke sličnosti između njih. Klasifikacijski model za svaki ulaz predviđa klasu *relevantan*, ako je ulazni FAQ-par relevantan za ulazni upit, ili *nerelevantan* inače. Dodatno, klasifikacijski model za svaku svoju odluku daje mjeru pouzdanosti da je ispravna klasa *relevantan*, koja se može interpretirati kao stupanj relevantnosti ulaznog FAQ para za ulazni upit. Rangirana lista

za zadani upit generira se tako da se za svaki FAQ-par u kombinaciji sa zadanim upitom primijeni klasifikacijski model. FAQ-parovi se potom redaju silazno po pouzdanosti modela da je ispravna klasa *relevantan*.

Skup za učenje sastoji se od parova (q, f) iz skupa podataka VipFAQ. Pri tome je $q \in Q_r$ korisnički upit iz skupa parafraziranih upita, a $f \in F_{rel}$ je FAQ-par iz skupa relevantnih FAQ-parova za taj upit. Svaki takav (q, f) par predstavlja pozitivan primjer za učenje. Kako bi se stvorili negativni primjeri, slučajno su odabrani (q, f) parovi iz skupa pozitivnih parova, te je f dio zamijenjen sa slučajno odabranim FAQ-parom f' koji nije relevantan za q. Generiranje svih mogućih negativnih primjera (q, f') uzrokovalo bi da skup za učenje modela postane vrlo neuravnotežen. Zbog toga se, uz dostupnih N pozitivnih primjera, generira samo $2N$ od svih mogućih negativnih primjera. Kako $|F_{rel}|$ varira ovisno o korisničkom upitu q, broj primjera za učenje N po upitu također varira. U prosjeku, N je 329.

Kao klasifikacijski model koristi se stroj potpornih vektora, koji je opisan u poglavlju 2.1., s radijalnom baznom jezgrenom funkcijom (engl. *radial basis kernel function* – RBF). Koristi se LIBSVM implementacija iz (Chang i Lin, 2011).

Za učenje klasifikacijskog modela potrebno je izračunati vektor značajki za svaki primjer za učenje (q, f). Značajke mjere semantičku sličnost između q i f. Preciznije, one mjere (1) semantičku sličnost q i pitanja iz f i (2) semantičku sličnost q i odgovora iz f. Ovakvo razmatranje oba dijela FAQ-para pokazalo se korisnim u srodnim istraživanjima (Tomuro i Lytinen, 2004). Dodatno, značajke koje se temelje na preklapanju n-grama također se izračunavaju i na temelju upita i imena kategorije u koju je FAQ-par svrstan (više detalja o kategorijama opisano je u poglavlju 3.6.). Prije izračuna značajki provedena je morfološka normalizacija koristeći morfološki leksikon izgrađen u (Šnajder i dr., 2008). Također, uklonjene su zaustavne riječi koristeći popis od 179 hrvatskih zaustavnih riječi. Pri tome one zaustavne riječi koje su dio naziva usluge (npr., zamjenica *"me"* u usluzi *"Nazovi me"*) nisu uklonjene. Na tako predobrađenom tekstu izračunat je niz značajki opisanih u nastavku.

6.2.1. Značajke temeljene na preklapanju riječi

Razumno je pretpostaviti da će relevantnost FAQ-para za neki korisnički upit biti pozitivno korelirana s leksičkim preklapanjem između teksta FAQ-para i upita. Klasifikacijski model koristi više značajki temeljenih na leksičkom preklapanju, slične onima predloženim u (Michel i dr., 2011) za zadatak klasifikacije parafraza i u (Šarić i dr., 2012) za određivanje semantičke sličnosti tekstova.

Preklapanje n-grama (NGO). Neka su T_1 i T_2 skupovi neprekinutih n-grama (npr., bigrama) u neka dva teksta. Značajka je definirana kao:

$$ngo(T_1, T_2) = 2 \times \left(\frac{|T_1|}{|T_1 \cap T_2|} + \frac{|T_2|}{|T_1 \cap T_2|} \right)^{-1} \tag{6.2.1}$$

Značajka izračunava u kojoj mjeri prvi tekst pokriva drugi te u kojoj mjeri drugi tekst pokriva prvi. Tako dobivene dvije veličine kombiniraju se pomoću harmonijske sredine. U ovom istraživanju računate su dvije inačice ove mjere – na unigramima i bigramima (pojedinim riječima i frazama duljine dvije riječi).

Preklapanje n-grama utežano informacijskim sadržajem (ICNGO). Značajka NGO daje jednaku važnost svim riječima. U stvarnim primjenama očekivano je da neke riječi nose više informacije od drugih. Količina informacije neke riječi može se mjeriti kroz mjeru informacijskog sadržaja (engl. *information content* – IC) (Resnik, 1995), koja je definirana kao:

$$ic(w) = \ln \frac{\sum_{w' \in C} freq(w')}{freq(w)} \tag{6.2.2}$$

gdje je C skup riječi iz velikog korpusa a $freq(w)$ frekvencija[1] riječi w u tom korpusu. Kod izračuna ove značajke korišten je korpus HRWAC koji su izgradili Ljubešić i Erjavec (2011). Neka su S_1 i S_2 skupovi riječi koje se pojavljuju u prvom i drugom tekstu. Mjera u kojoj prvi tekst pokriva drugi definirana je kao:

$$wwc(S_1, S_2) = \frac{\sum_{w \in S_1 \cap S_2} ic(w)}{\sum_{w' \in S_2} ic(w')} \tag{6.2.3}$$

Značajka ICNGO računa se kao harmonijska sredina $wwc(S_1, S_2)$ i $wwc(S_2, S_1)$.

6.2.2. Značajke temeljene na vektorskom prostoru

Sličnost tf-idf vektora (TFIDF). Ova značajka predstavlja tekstove kao tf-idf vektore, kao što je detaljno opisano u poglavlju 2.2.1. Iznos značajke je kosinusna sličnost tako dobivenih vektora teksta. Težine idf izračunate su na skupu svih FAQ-parova u skupu podataka VipFAQ. Pri tome je svaki FAQ-par (bez razlikovanja pitanja, odgovora i imena kategorije) tretiran kao jedinstven dokument.

LSA semantička sličnost (SS). Latetna semantička analiza (engl. *latent semantic analysis* – LSA), prvi puta opisana u (Deerwester i dr., 1990), vrlo je učinkovit način računanja sličnosti

[1] U području računalne obrade teksta pojam frekvencije riječi često znači broj pojavljivanja riječi u tekstu.

riječi i dokumenata. Više detalja o ovom postupku može se pronaći u poglavlju 2.2.1. Za izgradnju modela LSA primijenjen je postupak sličan onom opisanom u (Karan i dr., 2012). Model se gradi na temelju velikog korpusa hrvatskog weba iz (Ljubešić i Erjavec, 2011). Prije izgradnje modela riječi se lematiziraju pomoću morfološkog leksikona za hrvatski (Šnajder i dr., 2008). Prije dekompozicije SVD, svaki element matrice supojavnosti riječi transformira se u svoju tf-idf utežanu varijantu. Preliminarni pokusi pokazali su da su rezultati ovog modela zadovoljavajući i kod smanjenja dimenzionalnosti na samo 25 dimenzija, dok je veće smanjenje ipak uzrokovalo pad kvalitete rezultata. Zbog toga, u konačnoj inačici ove značajke korišteno je smanjenje dimenzionalnosti na 25 dimenzjia.

Model LSA predstavlja značenje riječi w semantičkim vektorom $v(w)$. Sukladno načelu semantičke kompozicionalnosti (Mitchell i Lapata, 2008), semantički vektor teksta T izračunat je kao semantička kompozicija (definirana kao zbrajanje vektora) sadržajnih riječi iz T:

$$v(T) = \sum_{w \in T} v(w) \tag{6.2.4}$$

Sličnost između tekstova T_1 i T_2, koja predstavlja vrijednost značajke SS, računa se kao kosinusna sličnost između $v(T_1)$ i $v(T_2)$.

Sličnost LSA utežana informacijskim sadržajem (engl. *Information Content SS* – ICSS). U značajci SS sve riječi koje se pojavljuju u tekstu smatraju se jednako važnima za računanje komponiranog vektora. Ovakav pristup zanemaruje činjenicu da neke riječi nose više informacije od drugih. Ovo je ispravljeno u varijanti Značajke SS koja, umjesto običnog zbroja, računa težinski zbroj semantičkih vektora riječi. Težine za pojedine riječi predstavlja njihov informacijski sadržaj, definiran izrazom (6.2.2). Komponirani semantički vektor teksta T računa se tada na sljedeći način:

$$c(T) = \sum_{w_i \in T} ic(w_i) v(w_i) \tag{6.2.5}$$

Vrijednost značajke ICSS za par tekstova T_1 i T_2 računa se, slično kao kod značajke SS, kao kosinusna sličnost između $c(T_1)$ i $c(T_2)$.

Pohlepno uparivanje riječi (engl. *aligned lemma overlap* – ALO). Ova značajka mjeri sličnost dvaju tekstova tako što pohlepno semantički uparuje riječi. Za usporedbu tekstova $T1$ i $T2$ prvo se računaju semantičke sličnosti svih parova riječi od kojih je jedna iz $T1$, a druga iz $T2$. Potom se najsličniji par stavlja na popis pronađenih parova te se riječi iz para uklanjaju iz tekstova. Postupak se ponavlja dok jedan od tekstova ne postane prazan. Pronađeni parovi su utežani informacijskim sadržajem. Informacijski sadržaj para riječi definiran je kao maksimum

informacijskih sadržaja riječi u paru. Sličnost dvije riječi tada je dana sa

$$sim(w_1, w_2) = \max(ic(w_1), ic(w_2)) \times ssim(w_1, w_2)$$

gdje je $ssim(w_1, w_2)$ semantička sličnost riječi w_1 i riječi w_2, izračunata kao kosinusna sličnost njihovih LSA vektora, dok je ic informacijski sadržaj definiran izrazom (6.2.2). Ukupna sličnost dvaju tekstova definirana je kao težinski zbroj sličnosti parova pronađenih u tekstovima normaliziran duljinom duljeg teksta:

$$alo(T_1, T_2) = \frac{\sum_{(w_1,w_2)\in P} sim(w_1, w_2)}{\max(length(T_1), length(T_2))} \tag{6.2.6}$$

Ovdje P označava gore opisani skup svih pronađenih parova riječi. Sličnu mjera predložili su Lavie i Denkowski (2009) za evaluaciju strojnog prevođenja. Također, ista mjera sličnosti pokazala se vrlo učinkovitom značajkom za određivanje semantičke sličnosti kratkih tekstova u (Šarić i dr., 2012).

6.2.3. Poklapanje klasa tipa pitanja (QC)

Srodna istraživanja na temu odgovaranja na pitanja pokazuju da se točnost sustava za odgovaranja na pitanja može povećati korištenjem klasifikacije vrste pitanja (Ferrucci i dr., 2010). Intuicija iza ovog je da različite vrste pitanja traže različite vrste odgovora. Primjerice, ako je korisnički upit *"Koliko košta SMS poruka?"*, to nosi informaciju da se u odgovoru mora pojaviti neki numerički iznos. Zato bi poznavanje vrste pitanja u koju spada korisnički upit moglo biti korisno kao značajka u klasifikacijskom modelu.

Kako bi se istražila ova mogućnost, naučen je jednostavan klasifikator vrste pitanja. Za učenje je korišten skup podataka sa označenim vrstama pitanja razvijen u (Lombarović i dr., 2011). Skup se sastoji od 1.300 pitanja na hrvatskome jeziku, koja su svrstana u neki od šest mogućih razreda: *numerički iznos*, *entitet*, *čovjek*, *opis*, *lokacija* i *kratica*.

Slično kao u (Lombarović i dr., 2011), po frekvenciji riječi u skupu podataka odabrano je najčešćih 300 riječi i 600 bigrama kao značajke za klasifikator. Klasifikator SVM naučen na ovom skupu podataka i vrednovan unakrsnom provjerom s pet preklopa dostiže točnost od 80,16%. Ovo je neznatno slabiji rezultat nego najbolji rezultat postignut u (Lombarović i dr., 2011), no to se može pripisati tome što je u ovom istraživanju korišten manji skup značajki. Izgrađeni klasifikator koristi se za izračun dvije značajke za glavni SVM model: (1) tip pitanja za korisnički upit i (2) tip pitanja za pitanje u FAQ-paru. Svaka od ove dvije značajke prikazana je modelu SVM-a kao šest binarnih značajki od kojih za pojedini primjer je samo jedna aktivna.[2]

[2]Ovakva shema se u literaturi na engleskom jeziku naziva *dummy features* ili *one-hot-encoding*.

Tablica 6.1: Primjeri iz rječnika za proširenje upita.

Riječ iz upita	Riječi za proširenje upita	Napomena
face	facebook	Leksičko nepoklapanje koje se često događa.
ograničiti	ograničenje	Vrlo slične riječi koje nisu ista vrsta riječi.
cijena	trošak, koštati	Sinonimi koji se vrlo često koriste u domeni.
inozemstvo	roaming	Uključuje znanje o svijetu.
ADSL	internet	Povezane riječi koje se često koriste u domeni.

6.2.4. Rječnik za proširenje upita (engl. *query expansion dictionary* – QED)

Analiza pogrešaka (v. odjeljak 6.3.3.). otkrila je da bi se značajan broj pogrešaka kod kojih klasifikacijski model krivo svrstava *relevantan* primjer u razred *nerelevantan* mogla popraviti kroz proširenje upita sličnim ili povezanim riječima. Za ovu svrhu napravljen je malen, domenski-specifičan rječnik za proširenje upita. Ciljevi rječnika bili su: (1) rješavanje problema kod manjih varijacija u pisanju nekih riječi, (2) uvođenje u model informacije o visokoj sličnosti nekih domenski specifičnih pojmova i (3) grubo uključivanje "znanja o svijetu" korisnog za danu domenu u model. Stvoren rječnik sadrži 53 unosa, čiji primjeri su dani u tablici 6.1

6.3. Vrednovanje

6.3.1. Postav vrednovanja

Kako je SVM nadzirani klasifikacijski model, prikladno je vrednovanje pomoću unakrsne provjere s pet preklopa na skupu podataka VipFAQ. U svakom preklopu klasifikacijski se model uči na podacima za učenje generiranima koristeći korisničke upite iz skupa za učenje (u svakoj iteraciji četiri od pet preklopa), na način kako je opisano ranije u ovom odjeljku. Potom se naučeni model primjenjuje na korisničke upite iz skupa za ispitivanje. Hiperparametri modela SVM se u svakom preklopu optimiraju na malom izdvojenom skupu za provjeru. Hiperparametri su u ovom slučaju C i γ, kao što je opisano u odjeljku 2.1.

Kako bi se temeljitije istražio utjecaj pojedinih značajki na kvalitetu postupka pretraživanja, isprobano je više inačica postupka. Inačice se međusobno razlikuju po skupovima značajki koje koriste. Pregled inačica koje su razmotrene dan je u tablici 6.2.

Kao temeljni postupak pretraživanja koristi se standardni postupak pretraživanja temeljen na vektorskom prostoru tf-idf, kakav je opisan u odjeljku 2.2.1. Ovaj postupak predstavlja korisničke upite i FAQ-parove kao vektore tf-idf, te provodi rangiranje računajući kosinusnu sličnost između njih. U okviru ovog istraživanja pitanje, odgovor i kategorija koji su dostupni u

Tablica 6.2: Značajke za pojedine inačice postupka.

Inačica	RM1	RM2	RM3	RM4	RM5
NGO	+	+	+	+	+
ICNGO	+	+	+	+	+
TFIDF	–	+	+	+	+
SS	–	–	+	+	+
ICSS	–	–	+	+	+
ALO	–	–	+	+	+
QED	–	–	–	+	+
QC	–	–	–	–	+

svakom FAQ-paru su združeni u jedinstveni komad teksta prije računanja vektora tf-idf za svaki FAQ-par.

6.3.2. Rezultati

Vrednovanje klasifikacijskog modela Kao postupak za pretraživanje koristi se nadzirani klasifikacijski model SVM. Učinkovitost ovog klasifikacijskog postupka izravno utječe na konačnu učinkovitost pretraživanja. Zbog toga je zanimljivo klasifikacijski model vrednovati na zadatku klasificiranja parova korisničkih upita i FAQ-parova u razrede *nerelevantan* ili *relevantan*. U ovu svrhu, u svakoj iteraciji unakrsne provjere stvoren je klasifikacijski zadatak koristeći samo korisničke upite iz ispitnog skupa. Ovo je napravljeno na identičan način na koji je generiran i skup za učenje klasifikacijskog modela.[3]

Preciznost, odziv i mjera F_1 za svaku inačicu postupka prikazane su u tablici 6.3. Inačica RM4 radi bolje od svih ostalih inačica. Inačica RM5, koja dodatno koristi značajke dobivene klasifikacijom pitanja, daje slabije rezultate od inačice RM4. Ovo upućuje na to da je kvaliteta klasifikacijskog modela za pitanja nedovoljna. Detaljnija analiza ove pojave pokazala je da se uzrok može pronaći u nepoklapanju domene u kojoj je klasifikator pitanja učen (općenita činjenična pitanja) i domene u kojoj je korišten (FAQ mobilnog operatera). Dodatno, neki od korisničkih upita u skupu VipFAQ uopće nisu pitanja (vidjeti Tablicu 3.9); npr., *"Popravak mobitela."*. Zato nije iznenađujuće da značajke temeljene na klasifikaciji pitanja ne pomažu.

Vrednovanje pretraživanja Rezultati klasifikacijskog modela SVM, kada se taj koristi za rangiranje FAQ-parova za upite u ispitnom skupu, dani su u tablici 6.4. Dane su standardne

[3] Uzorkovanje N pozitivnih i $2N$ negativnih parova, gdje je N ovdje broj primjera u ispitnom skupu.

Tablica 6.3: Rezultati klasifikacije na skupu VipFAQ.

Inačica postupka	P	R	F_1
RM1	14,1	68,5	23,1
RM2	25,8	75,1	37,8
RM3	24,4	75,4	36,3
RM4	**25,7**	**77,7**	**38,2**
RM5	25,3	76,8	37,2

Tablica 6.4: Rezultati pretraživanja na skupu VipFAQ.

Inačica postupka	MRR	MAP	RP
tf-idf	0,341	21,77	15,28
RM1	0,326	20,21	17,6
RM2	0,423	28,78	24,37
RM3	0,432	29,09	24,90
RM4	**0,479**	**33,42**	**28,74**
RM5	0,475	32,37	27,30

mjere za vrednovanje pretraživanja informacija: srednji recipročni rang (engl. *mean reciprocal rank* – MRR), srednja preciznost (engl. *average precision* – AP) te R-preciznost (engl. *R precision* – RP), koje su opisane u odjeljku 2.4. Najbolja inačica postupka pretraživanja je je RM4, koja koristi sve značajke osim onih temeljenih na klasifikaciji pitanja. Najbolji rezultat mjere MRR od 0,479 (sa standardnom devijacijom $\pm$0,04 na pet preklopa) ukazuje na to da, u prosjeku, inačica RM4 rangira relevantan dokument u prva dva rezultata.

Učinkovitost inačica postupaka se, očekivano, povećava s dodavanjem sve složenijih vrsta značajki. No, inačica RM5 opet je iznimka, te ima slabiji rezultat od preostalih inačica, iz istih razloga kao i u prethodnom pokusu.

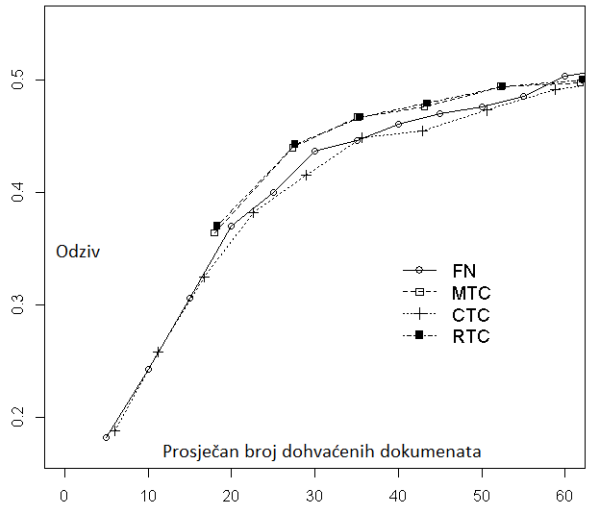
Također očekivano, učinkovitost postupka na klasifikacijskom zadatku pozitivno je korelirana s učinkovitosti postupka na zadatku pretraživanja informacija (usporediti tablicu 6.3 i tablicu 6.4). Zanimljiv slučaj je inačica RM4, koja primjerice u odnosu na inačicu RM2 popravlja mjeru F_1 za samo relativnih 1%. No, relativno poboljšanje u smislu npr. MRR mjere je čak oko 10%. Ovo ukazuje na to da, osim što popravlja odluke klasifikatora, rječnik za proširenje upita popravlja i pouzdanost klasifikatora u binarne odluke koje su i prije bile točne, a time i konačno dobiveno rangiranje.

Pri gornjoj analizi vrlo je važno napomenuti da je rječnik za proširenje upita sastavljen razmatrajući rezultate unakrsne provjere. Iako je broj pogrešaka koje rječnik ispravlja vrlo malen, ovo ipak čini inačice postupka RM4 i RM5 pristranima ovom skupu podataka. Objektivna procjena maksimalne učinkovitosti pretraživanja na neviđenim podacima bi stoga bila negdje između učinkovitosti inačica RM3 i RM4.

6.3.3. Analiza pogrešaka

Ručnim pregledom lažno pozitivno i lažno negativno klasificiranih primjera pronađeno je nekoliko vrsta karakterističnih pogrešaka koje imaju značajan udio u skupu svih pogrešaka. U nastavku je dan kratak pregled dobivenih saznanja.

1. **Leksička interferencija** – Iako korisnički upit ima značajnu leksičku sličnost s relevantnim FAQ-parovima, on također ima (najčešće slučajnu) visoku leksičku sličnost sa nerelevantnim FAQ-parovima. Budući da klasifikacijski model velik značaj pridaje značajkama temeljenim na preklapanju n-grama, takvi nerelevantni FAQ-parovi kvare rezultat otimajući od relevantnih FAQ-parova neke od pozicija pri vrhu rangirane liste;
2. **Leksički jaz** – Neki korisnički upiti izražavaju pitanje koje je vrlo slično nekom pitanju koje postoji kao dio nekog od FAQ-parova. No, korisnički upit je parafraziran na takav način da ostaje vrlo malo leksičkog preklapanja. Iako je ovaj problem, do neke mjere, ublažen značajkama temeljenim na modelu LSA, u nekim slučajevima relevantni FAQ-parovi ipak će biti rangirani vrlo nisko;
3. **Semantički jaz** – U ekstremnim slučajevima, parafraza korisničkog upita može biti toliko daleko od FAQ pitanja da, osim što nastaje leksički jaz, nastaje i semantički jaz. Semantički jaz je zapravo posebno težak slučaj leksičkog jaza, za čije razrješavanje je potrebno znanje o svijetu i domeni te dodatno logičko zaključivanje na temelju tog znanja. Primjer takvog upita je *"Postoji li mogućnost korištenja Vip kartice u Australiji?"*, gdje je relevantno FAQ pitanje *"Kako mogu saznati postoji li GPRS/EDGE ili UMTS/HSDPA roaming u zemlji u koju putujem?"*. Potrebno je znati da je Australija inozemstvo i da se roaming koristi u inozemstvu. Dodatno, potrebno je na temelju toga zaključiti da će informacija o roamingu vjerojatno nositi informaciju o korištenju kartice u Australiji;
4. **Problemi kod preklapanja** – U nekim slučajevima riječi koje se odnose na identične koncepte imaju blago različite oblike koji se ne preklapaju. Ovo se najčešće događa za raznolike morfološke oblike riječi koje nisu prisutne u korištenom rječniku za morfološku normalizaciju. Vrlo čest primjer je riječ *"Facebook"*, te njene kolokvijalne varijante kao što su *"fejs"* i *"face"*, te brojni morfološki oblici tih riječi. Rješavanje ovih slučajeva je vrlo važno jer su oni vrlo česti u skupu podataka VipFAQ. Jedno jednostavno rješenje bi bilo da se, u slučajevima kada riječ nije prisutna u rječniku za morfološku normalizaciju,



Slika 6.1: Odziv u odnosu na prosječan broj dohvaćenih dokumenata (za različite strategije odsijecanja).

normalizacija provede korjenovanjem.[4]

6.3.4. Strategije odsijecanja

Postupak pretraživanja na svom izlazu daje popis svih FAQ-parova rangiranih po relevantnosti za korisnički upit. Budući da većina nisko-rangiranih FAQ-parova nije relevantna, prikazivanje cijelog popisa korisniku nepotrebno ga dodatno opterećuje. Kako bi se ublažio ovaj problem, razmotreno je nekoliko strategija za ograničavanje duljine popisa rezultata:

- **Prvih N (FN).** Vraća najbolje rangiranih N dokumenata sa popisa;
- **Prag na relevantnost (MTC).** Definiran je prag na iznos mjere relevantnosti. Strategija vraća samo one FAQ-parove za koje je relevantnost iznad zadanog praga;
- **Kumulativni prag na relevantnost (CTC).** Definiran je prag na kumulativnu mjeru relevantnosti. Strategija vraća one FAQ-parove f za koje je zbroj relevantnosti FAQ-para f i svih drugih dokumenata rangiranih bolje od f iznad zadanog praga;
- **Relativni udio (RTC).** Vraća one FAQ-parove čiji iznos relevantnosti je unutar određenog zadanog postotka u odnosu na iznos relevantnosti najbolje rangiranog FAQ-para.

Dobra strategija odsijecanja bi, u prosjeku, trebala vratiti što manji broj FAQ-parova, ali istovremeno održavati visok odziv. Kako bi se istražilo ovo svojstvo strategije odsijecanja, razmatran je odziv u odnosu na prosječan broj dohvaćenih FAQ-parova. Rezultat razmatranja je prikazan na slici 6.1. Krivulja za idealnu strategiju odsijecanja prolazila bi kroz gornji lijevi kut slike. Nije se pokazala značajna razlika između četiri strategije, MTC i RTC su slične učinkovitosti te malo bolje od FN i CTC. Kada broj dokumenata koji se dohvaća raste, razlike između strategija iščezavaju.

[4]Jednostavan nenadziran postupak morfološke normalizacije koji se najčešće temelji na pravilima.

6.3.5. Brzina i skalabilnost

Ovaj postupak za pretraživanje FAQ-zbirki implementiran je u programskom jeziku Java. Jedina vanjska biblioteka koja se koristi je Java verzija biblioteke LIBSVM (Chang i Lin, 2011). U smislu brzine izvođenja sustava glavno usko grlo predstavlja izračun značajki za klasifikacijski model. One ovise o korisničkom upitu, pa ne mogu biti unaprijed izračunate. Značajka koja ima najveću računsku složenost je ALO jer uključuje računanje vrlo velikog broja kosinusnih sličnosti nad semantičkim vektorima riječi.

Općenito, vrijeme odziva izgrađenog sustava za pretraživanje FAQ-zbirki je prihvatljivo – na 1.222 FAQ-para, koliko sadrži skup VipFAQ, rezultati za korisnički upit se dohvaćaju unutar jedne sekunde. No, pri izgradnji rezultata za dani korisnički upit sustav mora generirati značajke i primijeniti klasifikacijski model na svaki FAQ-par u FAQ-zbirci. Zbog toga vrijeme odziva linearno ovisi o broju FAQ-parova. Za veće FAQ-zbirke bio bi potreban korak predobrade u kojem bi se prvo primijenio jednostavan i brz postupak pretraživanja. Potom bi se samo manji skup boljih kandidata slao u klasifikacijski model radi izgradnje konačnog rangiranja.

6.4. Rasprava

U ovom poglavlju opisano je istraživanje pretraživanja FAQ-zbirki provedeno za hrvatski jezik. Postupak pretraživanja temeljen je na nadziranome strojnom učenju. Model za pretraživanje naučen je na skupu podataka VipFAQ koji sadrži binarne oznake relevantnosti. Kako bi se premostio poznat problem leksičkog jaza korištene su brojne značajke temeljene na različitim načinima modeliranja semantičke sličnosti. Predloženi model pretraživanja vrednovan je na skupu podataka VipFAQ, na kojem postiže zadovoljavajuće performanse od 0,47 bodova mjere MRR.

Analiza pogrešaka pokazala je da izgrađen model pridaje velik značaj značajkama temeljenim na preklapanju n-grama. Zbog toga je većina grešaka uzrokovana varljivo velikim ili malim preklapanjem n-grama. Jedan način za rješavanje prvog problema je korištenje značajki temeljenih na sintaktičkoj analizi. Jedan jednostavan način za ovo je uključenje uzoraka po vrstama riječi (engl. *part-of-speech patterns*), kako bi se pronašle slične sintaksne strukture u tekstovima. Složeniji način korištenja sintakse bi bilo korištenje ovisnosnih relacija koje se mogu dobiti na izlazu sustava za ovisnosno parsiranje. Takav sustav za hrvatski jezik razvijen je u (Agić i Merkler, 2013).

Vrednovanje modela je pokazalo da čak i malen, domenski-specifičan rječnik za proširenje upita može dovesti do poboljšanja rezultata. Jedna mogućnost daljnjeg unaprjeđenja sustava je da se razmotri automatiziranje postupaka za stvaranje takvog rječnika. Jedna mogućnost za to bi bila korištenje zapisa s klikovima sakupljenih tijekom dužeg vremenskog perioda, što je uspješno primijenjeno na zadatak pretraživanja teksta (Cui i dr., 2002; Kim i Seo, 2006).

Rezultati opisani u ovom poglavlju pokazuju da je moguće napraviti učinkovit sustav za pretraživanje FAQ-zbirki na hrvatskom jeziku. Iako je ovo predistraživanje bilo uspješno, u nastavku se rad bavi isključivo zbirkama na engleskom jeziku. Ipak, za sva dalje opisana istraživanja ne postoji prepreka tome da se ona provedu za hrvatskom jeziku. No, zbog vremenskih ograničenja, to je ostavljeno za buduća istraživanja.

Ovim poglavljem završen je opis predistraživanja, u sljedeća tri poglavlja bit će opisani postupci za (1) izgradnju FAQ-zbirke, (2) održavanje FAQ-zbirke kroz otkrivanje nepokrivenih pitanja i (3) pretraživanje FAQ-zbirke. Ova tri postupka ujedno tvore i glavne doprinose istraživanja opisanog u ovom radu.

Poglavlje 7.

Izgradnja zbirke pitanja i odgovora

Ovim poglavljem kreće opis glavnog dijela istraživanja, svako od sljedeća tri poglavlja bavi se opisom jednog od izvornih znanstvenih doprinosa. Prvi doprinos, koji će biti opisan u ovom poglavlju, jest postupak za izgradnju zbirke pitanja i odgovora.

7.1. Motivacija i uvod

U prethodnim poglavljima nabrojane su brojne prednosti korištenja zbirki često postavljanih pitanja. One daju značajnu motivaciju za primjenu FAQ-zbirki kako bi se korisnicima sustava odgovaranja na pitanja olakšao i ubrzao pristup traženim informacijama. No, kako bismo mogli iskoristiti dobrobiti FAQ-zbirki, potrebno ih je prvo izgraditi. Budući da su predmet ovog rada domenski specifične FAQ zbirke, usredotočit ćemo se na postupke za strojno-potpomognutu izgradnju upravo takve vrste zbirki.

Najjednostavniji pristup ovom problemu jest da stručnjak za pojedinu domenu ručno izgradi cijelu zbirku. Pri tome stručnjak mora samostalno predvidjeti informacijske potrebe koje će budući korisnici zbirke imati te oblikovati odgovarajuće FAQ-parove koji pokrivaju upravo te informacijske potrebe. Očekivana kvaliteta sadržaja u FAQ-zbirci nastaloj ovim postupkom vrlo je visoka, jer očekujemo da je domenski stručnjak vrlo dobar u oblikovanju sadržaja koji unosi u zbirku. Također, domenski stručnjak može u zbirku dodati metapodatke koji olakšavaju pretraživanje, npr. ključne riječi. Primjeri takvih zbirki mogu se naći u (Sneiders, 2002, 2009). Ipak, kao što je već komentirano u uvodnom poglavlju, postoje tri nedostatka kod ovakvog postupka izgradnje zbirke. Prvo, iako je domenski stručnjak dobar u predviđanju budućih upita, nije realno očekivati da će pokriti sve informacijske potrebe koje bi korisnici mogli imati. Posljedica toga jest da će izgrađena zbirka biti nepotpuna. Drugo, domenski stručnjak može u zbirku nehotice ubaciti FAQ-parove koji zapravo nisu zanimljivi korisnicima. Tako izgrađena zbirka sadržava nepotrebne FAQ-parove, koji zbunjuju modele za pretraživanje zbirke te tako smanjuju ukupnu kvalitetu pretraživanja. Treće i najvažnije, trud koji domenski stručnjak mora

uložiti u postupak ručne izgradnje FAQ-zbirke može biti vrlo velik. Zbog ovog nedostatka, ručna izgradnja FAQ-zbirke koja bi bila praktično primjenjiva u stvarnim uvjetima često je vrlo dugotrajna ili čak nemoguća zadaća.

Alternativa ručnoj izgradnji FAQ-zbirke je strojno-potpomognuta izgradnja. U ovom postupku domenski stručnjak[1] gradi zbirku uz pomoć računalnog postupka. Uloga računalnog postupka jest da što je moguće više olakša i ubrza izgradnju zbirke tako što će pomoći domenskom stručnjaku da (1) odredi koji korisnički upiti su najčešći te (2) učinkovito pronađe odgovore na te upite u dostupnoj literaturi. U okviru ovog rada pretpostavljamo da računalni postupak obavlja ovu zadaću na temelju analize:

1. Q – velikog broja neodgovorenih korisničkih upita;
2. L – dostupne baze znanja u tekstnom obliku.

Opravdanje za ovakvu pretpostavku jest to što su stvarnim primjenama često oba ova izvora dostupna. Na primjer, pretpostavimo da radimo FAQ-zbirku koja se bavi uslugama neke telekomunikacijske tvrtke. U ovom slučaju Q bi mogao biti skup upita poslanih preko web-forme ili e-maila prema korisničkoj službi tvrtke (upiti ne moraju nužno biti odgovoreni). Nadalje, L bi bio skup dokumentacije koju tvrtka održava i u kojoj se mogu naći informacije potrebne za odgovaranje na korisnička pitanja. Ovaj dio rada predlaže postupak za strojno-potpomognutu izgradnju FAQ-zbirke koji se bavi ovom zadaćom, a sastoji se od dvije glavne komponente:

1. Postupak za grupiranje upita aktivnim učenjem – analizira skup korisničkih upita Q i pomaže stručnjaku odrediti koje informacijske potrebe su najzastupljenije (dosljedno se ponavljaju) u njemu. Upravo te informacijske potrebe najbolji su kandidati za pitanja u FAQ-parovima izgrađene FAQ-zbirke. Stručnjak vodi postupak grupiranja tako što na temelju aktivnog učenja odgovara na pitanja sustava te tako uvodi ograničenja u grupiranje;
2. Postupak za dohvat relevantnih odlomaka iz dokumentacije – postupak na temelju podskupa upita iz Q, koji odgovaraju istoj informacijskoj potrebi, pretražuje skup dokumenata L i izlučuje odlomke koji bi mogli biti relevantni za tu informacijsku potrebu. Upravo ti odlomci trebali bi omogućiti stručnjaku da brzo i učinkovito oblikuje odgovor u FAQ-paru.

Stručnjak koristi sustav tijekom izrade FAQ-zbirke tako da pomoću prve komponente odredi skup informacijskih potreba koje će oblikovati u pitanja kod stvaranja novih FAQ-parova. Potom, za tako odabrane informacijske potrebe, preko druge komponente pronalazi i oblikuje odgovore na njih, odnosno dodaje odgovore u pripadne FAQ-parove. Skup FAQ-parova koji nastaje ovim postupkom predstavlja novu domenski specifičnu FAQ-zbirku.

Ovakav pristup ublažava probleme nedostajućih i suvišnih FAQ-parova na način da olakšava stručnjaku da razluči koje informacijske potrebe zaista trebaju biti uvrštene u FAQ-zbirku.

[1] U ovom poglavlju pod pojmom *stručnjak* označavamo osobu koja koristi sustav za izgradnju zbirke, dok pojmom *korisnik* označavamo krajnjeg korisnika koji koristi gotovu zbirku za dohvat informacija.

Nadalje, pristup automatizira značajan dio posla izgradnje zbirke, pa tako ublažava problem velikog truda potrebnog za njenu ručnu izgradnju. Točnije, stručnjak ne mora više sam provoditi analizu mogućih informacijskih potreba te može brže dohvatiti relevantne odlomke dokumentacije koji su potrebni za formulaciju odgovora.

U nastavku ovog poglavlja dan je pregled relevantne literature iz ovog područja. Potom su detaljno opisani način rada i vrednovanje obje komponente predloženog postupka za strojno-potpomognutu izgradnju FAQ-zbirke temeljenog na gornjim pretpostavkama.

7.2. Pregled literature

Prva komponenta predloženog rješenja svodi se na grupiranje s ograničenjima (engl. *constrained clustering* u kombinaciji s aktivnim učenjem (engl. *active learning*) ograničenja. Grupiranje s ograničenjima dobro je istražen zadatak. U tom zadatku, algoritmu grupiranja je uz skup primjera dostupan i skup ograničenja. Ograničenje je definirano nad parom primjera i daje informaciju o tome trebaju li ta dva primjera biti svrstana u istu grupu (*pozitivno* ograničenje) ili ne smiju biti svrstani u istu grupu (*negativno* ograničenje). Budući da algoritmi grupiranja s ograničenjima kroz ograničenja imaju pristup dodatnim informacijama o ispravnom izlazu, oni u pravilu rade bolje od algoritama koji takva ograničenja nemaju. Međutim, cijena za bolje performanse jest što je potrebno definirati ograničenja. To skoro uvijek znači dodatno ručno označavanje.

Postoje dva temeljna pristupa zadatku ograničenog grupiranja. Prvi pristup ima ograničenja ugrađena u sam postupak grupiranja. Jedan takav pristup opisan je u (Wagstaff i dr., 2001), gdje je klasičan postupak K srednjih vrijednosti nadograđen s dodatnim pravilima koja onemogućavaju da pojedini primjer bude dodijeljen nekom centroidu, ako bi to prekršilo neko od zadanih ograničenja. U (Rangapuram i Hein, 2012) opisano je poopćenje postupka spektralnog grupiranja koje može raditi s neizrazitim ograničenjima. Postupak može optimirati kompromis između funkcije cijene grupiranja i broja ograničenja koja su pri tome prekršena. Postupak učenja temelji se na varijanti gradijentnog spusta. Pristup istih svojstava, također za postupak spektralnog grupiranja, izložen je u (Wang i Davidson, 2010) te dodatno razrađen u (Wang i dr., 2014). U ovom slučaju učenje se temelji na rastavu matrice na vlastite vektore i vlastite vrijednosti. Postupak koji uključuje obje vrste ograničenja za postupak grupiranja Gaussovim mješavinama, temeljen na postupku maksimizacije očekivanja, opisan je u (Shental i dr., 2004). Postupak za grupiranje DBSCAN također ima varijante koje uključuju ograničenja i one su istražene u (Lelis i Sander, 2009; Campello i dr., 2013).

Drugi moguć pristup jest da se umjesto promjene algoritma uvede promjena mjere udaljenosti između primjera koje treba grupirati. Ovakav pristup poznat je u literaturi kao učenje mjere udaljenosti (engl. *metric learning*). Primjer ovog pristupa može se naći u (Xing i dr.,

2003), gdje je funkcija sličnosti između primjera $\vec{x}$ i $\vec{y}$ definirana kao $\sqrt{(\vec{x}-\vec{y})A(\vec{x}-\vec{y})}$, gdje je matrica A parametar. Vrijednosti u A određuju se kroz optimizacijski postupak tako da primjeri za koje vrijedi pozitivno ograničenje postanu bliži, a primjeri za koje vrijedi negativno ograničenje postanu udaljeniji. Kriterij optimizacije temelji se na euklidskoj udaljenosti primjera za koje imamo definirana ograničenja. Uz izmijenjene udaljenosti među primjerima, bilo koji postupak grupiranja doći će do rezultata koji vrlo vjerojatno zadovoljava većinu danih ograničenja. Sličan algoritam koji formalizira problem učenja mjere udaljenosti kao traženje faktora skaliranja za Mahalanobisovu (Mahalanobis, 1936) udaljenost opisan je u (Davis i dr., 2007), uz optimizacijski kriterij temeljen na mjeri diferencijalne relativne entropije iz područja teorije informacija.

Osim pristupa koji pripadaju u jednu od dvije gore navedene skupine postoje i kombinirani pristupi koji ugrađuju ograničenja u postupak grupiranja te istovremeno uče mjeru udaljenosti. Primjer takvog pristupa opisan je u (Bilenko i dr., 2004), gdje je definirana posebna funkcija cijene za K-Means u koju su istovremeno ugrađena i ograničenja i mjera udaljenosti. Povezan pristup opisan je u (Basu i dr., 2004), gdje se kao model za grupiranje koristi skriveno Markovljevo slučajno polje (engl. *Hidden Markov Random Field*). Dodatno, uvedene su parametrizirane varijante kosinusne sličnosti i Kullback-Leibler divergencije (Kullback i Leibler, 1951), čiji parametri su također uključeni u optimizacijski postupak koji provodi grupiranje.

Gore opisani postupci grupiranja mogu iskoristiti pozitivna i negativna ograničenja na parove primjera. Takva ograničenja potrebno je ručno označiti, što predstavlja značajan posao za ljudske označivače. Kako bi se, koliko je to moguće, olakšao posao označivača, istraženi su postupci označavanja ograničenja temeljeni na aktivnom učenju. Glavna ideja ovih pristupa jest da se prije označavanja na podacima provede grupiranje bez ograničenja. Potom se razmatraju granični primjeri ili grupe primjera za koje je postupak grupiranja bio najmanje siguran. Za te slučajeve sustav postavlja označivaču pitanja oblika: "Trebaju li ova dva primjera biti svrstana u istu grupu?". Odgovorima na pitanja korisnik uvodi pozitivno ili negativno ograničenje nad odabranim primjerima. Tako označena ograničenja ubacuju se u postupak grupiranja kako bi se dobilo kvalitetnije grupiranje. Cijeli postupak može se ponoviti više puta. Kako su ograničenja uvedena nad upravo onim primjerima nad kojima je postupak grupiranja bio najmanje siguran, očekivano je da ona nose više korisne informacije za grupiranje nego što bi bio slučaj da su označavani slučajno odabrani parovi primjera. Ovo je glavna motivacija aktivnog učenja, jer se na ovaj način povećava učinkovitost označavanja.

Središnji problem kojim se bave postupci grupiranja temeljeni na aktivnom učenju je kako odrediti pitanja koja su najinformativnija. Pristupe možemo podijeliti u dvije glavne skupine. Prvi su pristupi koji pitanja određuju na razini pojedinog para primjera. Jedan takav postupak opisan je u (Mallapragada i dr., 2008), gdje se za pitanje označivaču bira primjer x koji je najdalje od sebi najbližeg centroida grupe. Potom se postavljaju pitanja za svaki par (x, c_i) gdje

je c_i centroid i-te grupe. U istu grupu spada i pristup opisan u (Xu i dr., 2005) koji pronalazi primjere za koje je grupiranje nesigurno analizom vlastitih vektora matrice podataka. Druga porodica pristupa umjesto parove pojedinih primjera razmatra cijele grupe primjera. Postupak opisan u (Nogueira i dr., 2012) provodi hijerarhijsko aglomerativno grupiranje (engl. *Hierarchical Agglomerative Clustering*) s ograničenjima tako da bira par grupa za koje je model najviše nesiguran po kriteriju temeljenom na međusobnoj udaljenosti grupa. Potom se postavlja pitanje za najbliža dva primjera iz odabranih grupa. Na hijerarhijskom grupiranju i međusobnoj udaljenosti grupa temelji se i postupak COBRA izložen u (Van Craenendonck i dr., 2017), koji kreće od većeg broja malih grupa, te ih iterativno spaja. Prije svakog spajanja dvije grupe označivač mora potvrditi spajanje tako što odgovara na upit za dva najbliža primjera iz tih grupa. Postupak COBRA specifičan je po tome što u njemu aktivno učenje ograničenja nije samo neobavezna nadogradnja, već je ugrađeno u duboko u postupak te on ne može raditi bez ljudskog označavanja. Vrijedi napomenuti da, iako ovi pristupi biraju primjere za upite označivaču analizirajući cijele grupe primjera, svako pojedino pitanje je i dalje na razini para primjera. To je slučaj jer je za označivača mnogo lakše odgovarati na pitanje definirano nad parom primjera, nego na pitanje definirano nad parom grupa.

Predložen novi postupak grupiranja korisničkih upita pripada skupini postupaka gdje su ograničenja ugrađena u sam postupak. Za razliku od većine opisanih pristupa, uz iznimku postupka COBRA, aktivno učenje ograničenja integrirano je u postupak grupiranja. Slično kao (Nogueira i dr., 2012; Van Craenendonck i dr., 2017) predložen postupak generira upite za označivača kroz analizu cijelih grupa primjera. Važna razlika jest da se u ovom radu osim spajanja grupa razmatra i operacija podjele grupa, te se obje vrste operacija provode u objedinjenom radnom okviru. Konačno, za razliku od ostalih postupaka, predloženi postupak posebno je prilagođen specifičnim svojstvima problema grupiranja upita za izgradnju FAQ-zbirke. Zato on na izlazu daje praktično korisniji skup grupa, nego što je to slučaj za općenite postupke grupiranja.

Cjelovito predloženo rješenje za strojno potpomognutu izgradnju FAQ-zbirke sastoji se od postupka za grupiranje upita aktivnim učenjem i postupka za dohvat relevantnih odlomaka u korisničkoj dokumentaciji. Obje komponente rješenja, opisane su u nastavku uz prikladno vrednovanje i završnu diskusiju.

7.3. Postupak za grupiranje upita aktivnim učenjem

7.3.1. Temeljne postavke

Kod strojno-potpomognute izgradnje FAQ-zbirke pod pretpostavkama opisanim u odjeljku 7.1., prvi problem koji je potrebno riješiti jest određivanje informacijskih potreba koje se često ponavljaju u korisničkim upitima. Ovaj problem može se svesti na zadatak grupiranja korisnič-

kih upita. U idealnom slučaju, postupak grupiranja trebao bi svrstati one upite koji pokrivaju istu informacijsku potrebu u istu grupu. Formalno, na raspolaganju je skup korisničkih upita $Q = \{q_1, ..., q_n\}$ za koji je potrebno naći skup grupa $C = \{c_1, ..., c_m\}$. U kontekstu zadatka pronalaženja čestih informacijskih potreba zanimaju nas zapravo samo one grupe upita koje su najizraženije, tj. sadrže najviše sličnih upita. Zbog toga je dozvoljeno da manje česti upiti ne budu pokriveni niti jednom grupom iz C. Ovo je razlika u odnosu na klasičan pristup grupiranju podataka, gdje nije običaj da postoje primjeri koji nisu dio neke od pronađenih grupa. Predložen postupak za grupiranje upita aktivnim učenjem obavlja ovako definiran zadatak grupiranja upita, a sastoji se od dva dijela. Prvi dio obuhvaća algoritme 1, 2 i 4, dok drugi dio obuhvaća algoritme 3 i 5. Ova dva dijela zajedno tvore prvu komponentu predloženog rješenja.

Oba dijela postupka kreću od gotovog grupiranja. Potom se iterativno postavljaju pitanja ljudskom označivaču (u daljnjem tekstu: stručnjak) te se na temelju odgovora prilagođava grupiranje. Kako bi opis bio općenitiji uvodimo sljedeće oznake. Prvo, pretpostavlja se da su nam na raspolaganju sljedeće generičke mjere:

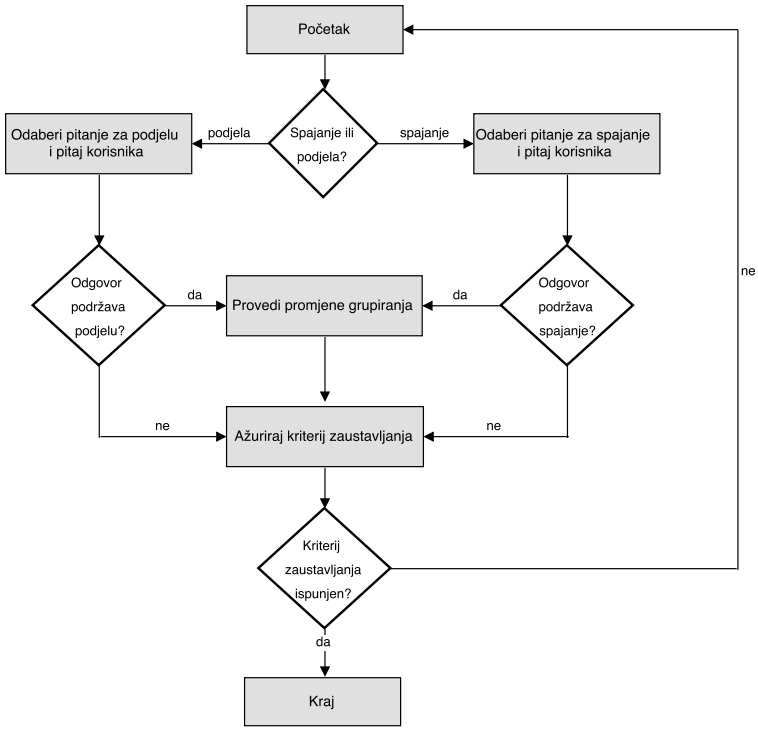
- $dist(q_i, q_j)$ – udaljenost upita q_i i q_j. Ova mjera opisuje razliku u značenju između dva upita. Veća semantička razlika upita povlači veću udaljenost;
- $S(c_i, c_j)$ – sličnost grupa. Mjera opisuje sličnost grupa c_i i c_j. Veća semantička sličnost upita u grupama povlači veću vrijednost ove mjere;
- $R(c)$ – raspršenost grupe. Mjeri koliko je neka grupa upita raspršena. Veće vrijednosti znače da je grupa više raspršena, tj., da su primjeri iz te grupe manje zbijeni skupa.

Drugo, pretpostavlja se da su za prilagodbu postojećeg grupiranja na raspolaganju sljedeće generičke operacije:

- $spoji(c_i, c_j)$ – spaja grupe c_i i c_j u jednu veliku grupu koja sadrži sve primjere koji su prije bili u grupama c_i i c_j;
- $podijeli(c_i)$ – provodi podjelu grupe c_i na dvije manje grupe koje zajedno pokrivaju sve primjere koji su prije bili u c_i;
- $obrisi(c_i)$ – briše grupu c_i. Svi upiti koji su bili u toj grupi nakon ove operacije nisu više dio niti jedne grupe.

Treće, pretpostavlja se da nam je dostupna funkcija $pitaj(q_i, q_j)$, koja nam za upite q_i i q_j vraća *pozitivno* ograničenje, ako ti upiti trebaju biti u istoj grupi, ili *negativno* ograničenje, ako upiti ne smiju biti u istoj grupi. Prilikom poziva funkcije, ažurira se skup P u kojem se pamte svi parovi (q_i, q_j) za koje je već postavljeno pitanje, kako bi se izbjeglo da se stručnjaku više puta postavlja isto pitanje. Također, s obzirom na način odabira konkretnih upita q_i i q_j za koje će pitanje biti postavljeno, definirane su sljedeće vrste pitanja:

- *Pitanje spajanja* – pitanje se postavlja ako postupak utvrdi da su dvije grupe vrlo slične te postoji indikacija da ih treba spojiti u jednu veću grupu. Detaljan pregled načina na koji se bira konkretan par upita iz tih grupa dan je u algoritmu 2;



Slika 7.1: Dijagram tijeka prvog dijela postupka.

- *Pitanje podjele* – pitanje se postavlja za grupu koja ima visoku mjeru raspršenosti. Za takvu grupu postoji indikacija da ju treba podijeliti na dvije manje grupe. Način na koji se bira konkretan par upita iz te grupe dan je u algoritmu 1;
- *Pitanje brisanja* – pitanje se postavlja za grupu c koja ima visoku mjeru raspršenosti. Za takvu grupu postoji indikacija da ju treba obrisati. Način na koji se bira konkretan par upita iz te grupe dan je u algoritmu 3. Iako je ovo pitanje vrlo slično pitanju podjele, razlika je što se kod pitanja podjele pitanje stručnjaku postavlja nad najdalja dva upita iz c dok se kod pitanja brisanja ono postavlja nad slučajno odabrana dva upita iz c. Motivacija za ovu razliku su specifična svojstva grupiranja nakon prvog dijela postupka, koja će biti detaljno opisana u opisu drugog dijela postupka.

7.3.2. Prvi dio postupka

Prvi dio postupka sastoji se od primjene operacija *podijeli* i *spoji*, ovisno o odgovorima stručnjaka na pitanja. Dijagram tijeka ovog djela dan je na slici 7.1. U ovom dijelu razmatraju se samo *pitanja podjele* i *pitanja spajanja*. Između ove dvije varijante pitanja u svakoj iteraciji postupak odlučuje dinamički. Cjelovit pregled postupka dan je u algoritmu 4. Važno je spomenuti nekoliko pojedinosti koji zaslužuju posebnu pažnju:

Prilagodljiv odabir vrste pitanja. U svakoj iteraciji algoritma vrsta pitanja odabire se na temelju vjerojatnosti p. Ova vjerojatnost dinamički se prilagođava tijekom izvođenja algoritma. Veličina p predstavlja vjerojatnost da će biti odabrano pitanje spajanja. U svakoj iteraciji p se

modificira sukladno jednom od sljedećih slučajeva:

1. Postavljeno je pitanje spajanja i odgovor stručnjaka je podržao spajanje – ovakav slučaj indicira da je grupiranje takvo da je spajanje povoljna operacija, zato se p povećava, kako bi se potaknulo postupak da više radi spajanje;
2. Postavljeno je pitanje spajanja i odgovor stručnjaka nije podržao spajanje – ovakav slučaj je indikacija da spajanje nije povoljna operacija u trenutnom grupiranju. Stoga se p smanjuje kako bi se potaknulo postupak da manje radi spajanje;
3. Postavljeno je pitanje podjele i odgovor stručnjaka je podržao podjelu – ovaj slučaj je analogan slučaju 1. Potrebno je potaknuti model da više radi podjelu. To se postiže smanjenjem p;
4. Postavljeno je pitanje podjele i odgovor stručnjaka nije podržao podjelu – ovakav slučaj je analogan slučaju 2. Model treba potaknuti da manje radi podjelu, što se ostvaruje povećanjem p.

Popisi zabrana i strpljenje. Tijekom izvođenja algoritma može se dogoditi da se postavi velik broj pitanja s ciljem provođenja neke operacije na istoj grupi ili paru grupa. Na primjer, pretpostavimo da je u nekoj iteraciji postupak odredio da će postaviti pitanje spajanja za par grupa (c_i, c_j), koji je najbolji kandidat za spajanje. Postupak će odabrati dva upita q_i iz c_i i q_j iz c_j kao što je opisano u algoritmu 2, te će postaviti stručnjaku pitanje nad ta dva upita. Pretpostavimo, nadalje, da grupe c_i i c_j nije potrebno spajati. Odgovor stručnjaka na pitanje spajanja generirati će negativno ograničenje nad q_i i q_j, pa spajanje neće biti provedeno. Par (q_i, q_j) bit će dodan u skup postavljenih pitanja P. Kada u nekoj od idućih iteracija postupak opet postavi pitanje spajanja, najbolji par grupa opet će biti upravo (c_i, c_j) te će postupak opet postaviti pitanje nad neka druga dva upita iz tih grupa. Pri tome će odgovor stručnjaka opet biti negativan. Usprkos tome, postupak će u svim daljnjim iteracijama gdje postavlja pitanje spajanja, postavljati pitanja za taj isti par grupa. Ovo će trajati sve dok ne se ne iscrpe sva moguća pitanja za par grupa (c_i, c_j). Broj mogućih pitanja je u ovom slučaju broj mogućih parova upita od kojih je jedan iz c_i a drugi iz c_j. Ovakvo ponašanje je vrlo nepoželjno, jer sustav postavlja vrlo velik broj nepotrebnih pitanja. Za ublažavanje ovog problema, u postupak se uvodi koncept strpljenja. *Strpljenje* je veličina koja broji pogreške koje je postupak napravio tijekom postavljanja pitanja. Pogreška je definirana kao slučaj gdje je sustav (1) postavio pitanje spajanja koje stručnjak nije podržao (pogreška spajanja) ili (2) postavio pitanje podjele koju stručnjak nije podržao (pogreška podjele). Za slučaj podjele, strpljenje se prati za svaku grupu, dok se za slučaj spajanja strpljenje prati za svaki par grupa. Ako strpljenje za pojedinu grupu ili par grupa pređe određen prag – SP_{max} za podjele i SS_{max} za spajanja – ažurira se pripadni popis zabrana. Ako se neka grupa ili par grupa nađe na pripadnom popisu zabrana, postupak ih zanemaruje u daljnjem postavljanju pitanja. Konkretno, u gornjem primjeru bi se nakon SS_{max} neuspješnih upita spajanja, par grupa (c_i, c_j) našao na popisu zabrane za spajanje. Zato bi, u

daljnjim iteracijama u kojima se postavlja pitanje spajanja, postupak postavljao pitanja za neki drugi par grupa. Uvođenjem koncepta strpljenja postigli smo da postupak nakon određenog broja pogreški odustane od grupe ili para grupa umjesto da postavlja velik broj nepotrebnih pitanja stručnjaku, što u značajnoj mjeri ublažava opisani problem.

Uvjeti zaustavljanja. Definiramo dva kriterija za zaustavljanje algoritma. Prvo, postupak završava kada nema više dozvoljenih upita. Ovo će se dogoditi u trenutku kada sve grupe budu na popisu zabrana za podjelu, a svi parovi parovi grupa budu na popisu zabrane za spajanje. Drugo, pokazuje se da se broj pitanja za stručnjaka može dodatno smanjiti tako da se uvede koncept *globalnog strpljenja*. Motivacija za ovo proizlazi iz zapažanja da bi veći broj uzastopnih pogreški određene vrste trebao biti indikacija da treba prestati postavljati pitanja te vrste. Ova intuicija izravno je ugrađena u postupak. Ako broj uzastopnih pogrešaka podjele premaši GSP_{max}, prestaju se postavljati pitanja podjele. Analogno, ako broj uzastopnih pogrešaka spajanja premaši GSS_{max}, prestaju se postavljati pitanja spajanja. U preliminarnim pokusima pokazalo se da uvođenje drugog kriterija zaustavljanja minimalno narušava kvalitetu dobivenog grupiranja, dok istovremeno značajno smanjuje ukupan broj pitanja koja postupak mora postaviti stručnjaku.

Algoritam 1: postavi_pitanje_podjela(Q,C)

1: **Ulaz**: Skup upita $Q = \{q_1, \ldots, q_n\}$ grupiranih u grupe $C = \{c_1, \ldots, c_k\}$
2: **Inicijalizacija**:
3: zabrana_podjela ← set()
4: strpljenje_podjela ← dict(), ključevi su identifikatori grupa a vrijednosti 0
5: **postavi_pitanje_podjela**:
6: kandidati_podjela ← skup svih grupa c iz C sortiran silazno po $R(c)$
7: **za** c **u** kandidati_podjela **učini**
8: **ako** c **nije u** zabrana_podjela **učini**:
9: $(q_1, q_2) \leftarrow q_1$ i q_2 iz c takvi da je dist(q_1, q_2) maksimalan i (q_1, q_2) nije u P
10: odgovor ← $pitaj(q_1, q_2)$
11: **ako** odgovor = spojeno **učini**:
12: strpljenje_podjela[c] ← strpljenje_podjela[c] + 1
13: **ako** strpljenje_podjela = SP_{max} **učini**:
14: zabrana_podjela.dodaj(c)
15: **vrati** (q_1,q_2,odgovor)
16: **ako** gornja petlja nije uspjela postaviti niti jedno pitanje **učini**:
17: **vrati** (-, -, nema mogućih pitanja)
18: **Izlaz**: q_1,q_2, odgovor

Algoritam 2: postavi_pitanje_spajanja(Q,C)

1: **Ulaz**: Skup upita $Q = \{q_1, \ldots, q_n\}$ grupiranih u grupe $C = \{c_1, \ldots, c_k\}$
2: **Inicijalizacija**:
3: zabrana_spajanje ← set()
4: strpljenje_spajanje ← dict(), ključevi su identifikatori parova grupa a vrijednosti 0
5: **postavi_pitanje_spajanja**:
6: kandidati_spajanje ← popis svih parova grupa (c_1, c_2) sortiran uzlazno po $S(c_1, c_2)$
7: **za** c_1, c_2 **u** kandidati_spajanje **učini**
8: **ako** c **nije u** zabrana_spajanje **učini**:
9: (q_1, q_2) ← q_1 iz c_1 i q_2 iz c_2 takvi da je dist(q_1, q_2) minimalan i (q_1, q_2) nije u P
10: odgovor ← $pitaj(q_1, q_2)$
11: **ako** odgovor = razdvojeno **učini**:
12: strpljenje_spajanje[c_1, c_2] ← strpljenje_spajanje[c_1, c_2] + 1
13: **ako** strpljenje_spajanje = SS_{max} **učini**:
14: zabrana_spajanje.dodaj(c_1, c_2)
15: **vrati** (q_1,q_2,odgovor)
16: **ako** gornja petlja nije uspjela postaviti niti jedno pitanje **učini**:
17: **vrati** (-, -, nema mogućih pitanja)
18: **Izlaz**: q_1,q_2, odgovor

Algoritam 3: postavi_pitanje_brisanja(Q,C')

1: **Ulaz**: Skup upita $Q = \{q_1, \ldots, q_n\}$ grupiranih u grupe $C' = \{c'_1, \ldots, c'_k\}$
2: **Inicijalizacija**:
3: zabrana_brisanje ← set()
4: strpljenje_brisanje ← dict(), ključevi su identifikatori grupa a vrijednosti 0
5: kandidati_brisanje ← popis svih grupa c' iz C' sortiran silazno po $R(c')$
6: **postavi_pitanje_brisanja**:
7: **za** c **u** kandidati_za_brisanje **učini**
8: **ako** c **nije u** zabrana_brisanje i c **nije u** zabrana_podjela **učini**:
9: (q_1, q_2) ← dva slučajna upita iz grupe c
10: odgovor ← $pitaj(q_1, q_2)$
11: **ako** odgovor = razdvojeno **učini**:
12: strpljenje_brisanje[c] ← strpljenje_brisanje[c] + 1
13: **ako** strpljenje_brisanje = SB_{max} **učini**:
14: zabrana_brisanje.dodaj(c)
15: **vrati** (q_1,q_2,odgovor)
16: **ako** gornja petlja nije uspjela postaviti niti jedno pitanje **učini**:
17: **vrati** (-, -, nema mogućih pitanja)
18: **Izlaz**: q_1,q_2, odgovor

Algoritam 4: PrviDio(Q, C)

1: **Ulaz**: Skup upita $Q = \{q_1, \ldots, q_n\}$ grupiranih u grupe $C = \{c_1, \ldots, c_k\}$
2: **Inicijalizacija**:
3: podjele_krivo ← 0
4: spajanja_ krivo ← 0
5: podjele_gotovo ← False
6: spajanja_ gotovo ← False
7: vjerojatnost_spajanja ← $0,5$
8: **PrviDio**:
9: **dok ne** podjele_gotovo **ili ne** spajanja_gotovo **učini**
10: **ako** slučajan broj $R \in [0,1]$ < vjerojatnost_spajanja **učini**
11: **ako** spajanje_gotovo: preskoči ostatak iteracije
12: (q_1, q_2, odgovor) ← postavi_pitanje_spajanja(Q,C)
13: **ako** odgovor = nema vise mogućih pitanja: spajanje_gotovo ← True
14: **inače ako** odgovor = spojeni **učini**:
15: vjerojatnost_spajanja ← vjerojatnost_spajanja + α
16: spajanja_krivo ← 0
17: spoji grupu od q_1 sa grupom od q_2
18: **inače ako** odgovor = razdvojeni **učini**:
19: vjerojatnost_spajanja ← vjerojatnost_spajanja - α
20: spajanja_krivo = spajanja_krivo + 1
21: spajanja_gotovo ← (spajanja_krivo = KS_{max})
22: **inače učini**
23: **ako** podjele_gotovo: preskoči ostatak iteracije
24: (q1, q2, odgovor) = postavi_pitanje_podjela(Q,C)
25: **ako** odgovor = nema vise mogućih pitanja: podjela_gotovo ← True
26: **inače ako** odgovor = razdvojeni **učini**:
27: vjerojatnost_spajanja ← vjerojatnost_spajanja - α
28: podjela_krivo ← 0
29: podijeli grupu u kojoj su q_1 i q_2
30: **inače ako** odgovor = spojeni **učini**:
31: vjerojatnost_spajanja ← vjerojatnost_spajanja + α
32: podjela_krivo ← podjela_krivo + 1
33: podjele_gotovo ← (podjela_krivo = KP_{max})
34: vjerojatnost_spajanja ← min(max(vjerojatnost_spajanja,0),1)
35: **Izlaz**: Ažurirane grupe $C' = \{c'_1, ..., c'_r\}$

7.3.3. Drugi dio postupka

Pažljivim pregledom grupiranja koje je rezultat prvog djela postupka utvrđeno je da se često događaju dvije karakteristične vrste problema:

1. *Nečiste grupe* – ovo su grupe koje imaju malu raspršenost, ali istovremeno sadrže upite za dvije informacijske potrebe ili više njih. Ovo se može dogoditi u slučaju kada upiti za različite informacijske potrebe sadrže slične riječi. Na primjer, *"Kako popraviti dječji*

bicikl?" i *"Kako popraviti svjetlo u dječjoj sobi?"*. Kako su upiti u skupovima podataka koje koristimo iz uske domene, ovakav slučaj se lako može dogoditi. Idealno, ovakve grupe trebale bi biti podijeljene tijekom prvog dijela postupka, no njihova niska raspršenost čini ih lošim kandidatima za podjelu, pa se to ne događa. Sa stanovišta stručnjaka koji gradi FAQ-zbirku, ovakve grupe su beskorisne jer ne opisuju jednu specifičnu informacijsku potrebu. Za potrebe daljnjeg razmatranja definiramo *koherentnost grupe* – $K(c)$ kao maksimalan udio koji neka informacijska potreba ima u grupi.[2] Uz ove oznake, nečiste grupe su one grupe c za koje je $K(c)$ malen.

2. *Suvišne grupe* – Ovo su različite grupe male raspršenosti i veće međusobne udaljenosti koje sadrže upite pretežito iste informacijske potrebe. Ovakav slučaj događa se kad postoji više različitih podjednako jasnih načina za izražavanje neke informacijske potrebe. Na primjer, *"Kako oprati proliven sok s kauča?"* i *"Kako ukloniti mrlje od tekućine trosjeda?"* Idealno, ove grupe bi trebale biti spojene tijekom prvog dijela postupka, no zbog njihove velike međusobne udaljenosti, to se ne događa. Ova zalihost nije dobra, jer bi stručnjaku bilo dovoljno da vidi samo jednu od tih grupa kako bi postao svjestan pripadne informacijske potrebe.

Način na koji se mogu ublažiti gornji problemi jest zapravo isti – brisanje nekih grupa. Brisanje pomaže u kontekstu problema (1) tako što briše grupe koje stručnjaku nisu korisne. Dodatno, brisanje pomaže i kod problema (2) tako što od više grupa koje odgovaraju istoj informacijskoj potrebi briše neke, te na taj način smanjuje broj suvišnih grupa za svaku informacijsku potrebu.

Ova intuicija motivira drugi dio postupka, koji se temelji na istim načelima kao i prvi dio. Razlika je što je jedina vrsta pitanja koja se koristi *pitanje brisanja*, a jedina operacija koja se provodi je operacija *obrisi*. Dijagram tijeka ovog dijela prikazan je na slici 7.2, dok je detaljan pseudokod naveden u algoritmu 5. Postupak radi tako da za svaku grupu postavlja određen broj pitanja brisanja te, ako stručnjak odgovori s *negativnim* ograničenjem na bilo koje od njih, grupa se briše. U odnosu na prvi dio postupka potrebno je podrobnije opisati sljedeće važne implementacijske pojedinosti.

Popisi zabrana i strpljenje. Analogno prvom djelu algoritma, za pojedinu grupu se neće postavljati sva moguća pitanja već najviše SB_{max} njih. Ako nakon tog broja pitanja odgovori stručnjaka ne podrže brisanje grupe c, ona se dodaje na popis zabrana brisanja te se više ne razmatra.

Uvjeti zaustavljanja Postupak prestaje kada ponestane mogućih pitanja. Preliminarni pokusi pokazali su da uvođenje dodatnog kriterija zaustavljanja pomoću koncepta globalnog strpljenja, kakav je bio uveden za prvi dio, u ovom slučaju nije korisno. Iako takav pristup smanjuje broj postavljenih pitanja, on previše kvari kvalitetu konačnog grupiranja te zato ovdje nije kori-

[2] Na primjer, ako neka grupa c sadrži pet upita q_i od kojih svaki adresira jednu od tri različite informacijske potrebe I_j kako slijedi: $q1 \rightarrow I_2$, $q2 \rightarrow I_1$, $q3 \rightarrow I_3$, $q4 \rightarrow I_1$, $q5 \rightarrow I_1$. Najveći udio ima informacijska potreba I_1 koju pokrivaju 3 upita od ukupno 5, pa je stoga $K(c) = \frac{3}{5}$.

šten.

Odabir upita. Kod pitanja brisanja potrebno je za neku grupu c postaviti pitanje nad dva upita q_i i q_j iz dotične grupe. Intuitivno bi bilo odabrati ove upite tako da je $dist(q_i, q_j)$ maksimalan, jer za upravo takva dva upita očekujemo da je najizgledniji odgovor stručnjaka negativan. No, u slučaju problema (1) opisanog gore, upiti unutar c koji odgovaraju *različitim* informacijskim potrebama nalaze se neočekivano blizu, često bliže nego upiti iz c koji odgovaraju *istim* informacijskim potrebama. Zbog toga gornja intuicija ne vrijedi. To se potvrdilo i u preliminarnim pokusima u kojima je strategija koja bira (q_i, q_j) slučajnim odabirom iz c pokazala najbolje rezultate. Važno je napomenuti da u slučaju pogreške (2) gornja intuicija ne vrijedi, stoga je strategija slučajnog odabira bolje prilagođena rješavanju problema (1) nego problema (2). No, strategija je ipak korisna, jer se pokazuje da nakon prvog dijela postupka postoji značajan broj nečistih grupa, tj. da je problem (1) izraženiji od problema (2). Dodatno, kako zbog ove strategije postupak radi brisanje nečistih grupa, neke od suvišnih grupa (koje su manje čiste) će također biti obrisane, što neizravno ublažava problem (2).

Pogrešno brisanje. Iako brisanje smanjuje broj suvišnih i nečistih grupa prisutnih u konačnom izlazu postupka, kod brisanja se može dogoditi pogreška. Na primjer, pretpostavimo da imamo grupu sa deset upita od kojih devet pokriva istu informacijsku potrebu. Čistoća takve grupe jest $0,9$ te nije izgledno da ju treba obrisati. Ipak, može se dogoditi da prilikom postavljanja pitanja stručnjaku slučajno bude odabran baš onaj par upita za koji će odgovor biti negativno ograničenje te će grupa biti obrisana. Zato je važno napomenuti da drugi dio postupka istovremeno i popravlja i kvari grupiranje. No, pokazuje se da je pozitivan utjecaj znatno veći od negativnog, pa se isplati provoditi ovaj dio postupka.

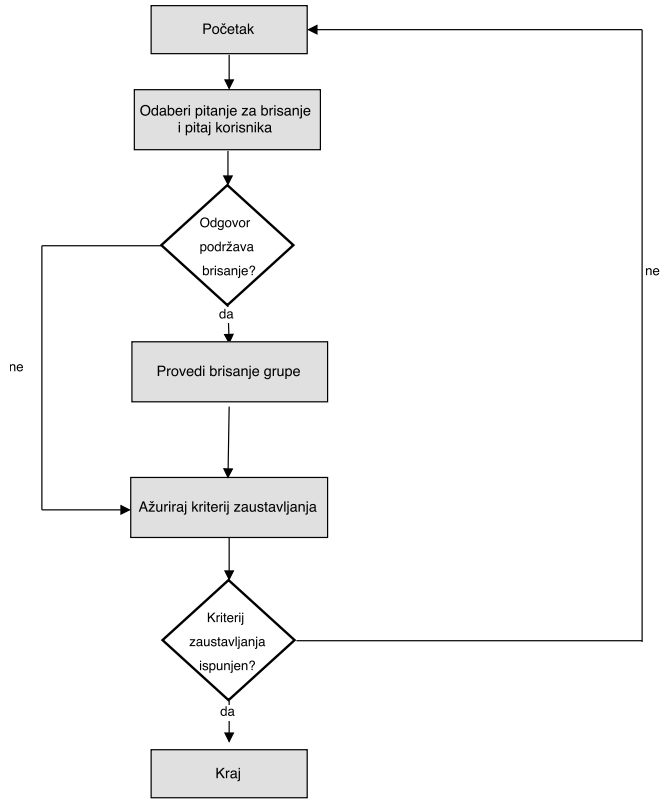
Algoritam 5: DrugiDio(Q, C')

1: **Ulaz**: Skup upita $Q = \{q_1, \ldots, q_n\}$ grupiranih u grupe $C' = \{c'_1, \ldots, c'_k\}$
2: **DrugiDio**:
3: **dok** True **učini**
4: (q1, q2, odgovor) ← postavi_pitanje_brisanja()
5: **ako** odgovor = nema vise mogućih pitanja: kraj
6: **inače ako** odgovor = razdvojeni **učini**:
7: obriši grupu koja sadrži q_1 i q_2
8: **Izlaz**: Ažurirane grupe $C'' = \{c''_1, ..., c''_r\}$

7.3.4. Vrednovanje

Postupak grupiranja upita vrednovan je na skupu FAQIR (opisan u odjeljku 3.4.), koji ima 1.233 upita grupiranih u 50 grupa[3] i skupu StackFAQ (opisan u odjeljku 3.5.), koji ima 1.250 upita

[3]Na oba skupa podataka, svaka grupa odgovara jednoj informacijskoj potrebi



Slika 7.2: Dijagram tijeka drugog dijela algoritma.

grupiranih u 125 grupa. Postupak zahtijeva početni broj grupa, koji nam u stvarnoj primjeni ne bi bio unaprijed poznat. Stoga, ako točan broj grupa označimo s K^*, evaluacija je provedena u sva tri moguća scenarija: (1) početni broj grupa jednak stvarnom ($K = K^*$), (2) početni broj grupa manji od stvarnog ($K < K^*$) i (3) početni broj grupa veći od stvarnog ($K > K^*$). Predloženi postupak uspoređen je s tri temeljna postupka: hijerarhijskim aglomerativnim grupiranjem (HAC), postupkom K-srednjih vrijednosti (KMeans), te postupkom spektralnog grupiranja (SC). Ovi temeljni postupci, opisani u odjeljku 2.1., ne postavljaju pitanja stručnjaku, pa su nenadzirani. Zbog potpunosti, vrednovano je izlazno grupiranje predloženog postupka nakon prvog dijela i nakon drugog dijela postupka.

Konačan cilj grupiranja upita jest identifikacija informacijskih potreba koje je potrebno uključiti u FAQ-zbirku. Način na koji stručnjak ostvaruje taj cilj jest da razmatra sve dobivene grupe te za one iz kojih se može jasno vidjeti neka informacijska potreba, uključuje tu informacijsku potrebu kao pitanje novog FAQ-para u izgrađenoj FAQ-zbirci. Smatra se da se informacijska potreba I jasno vidi iz grupe c ako je I većinska informacijska potreba za upite iz c i vrijedi $K(c) > 0,7$.[4] U tom slučaju, kaže se da grupa c *pokriva* informacijsku potrebu I. Potrebno je osmisliti mjeru vrednovanja dobivenog grupiranja koja će uzimati u obzir način na koji se ono koristi. Kroz razmatranje dolazimo do sljedećih svojstava očekivanih za kvali-

[4] Vrijednost 0,7 je odabrana unaprijed kao objektivno razumna čistoća grupe iz perspektive potencijalnog stručnjaka koji bi koristio sustav. Važno je naglasiti da ovo nije hiperparametar modela grupiranja.

tetno grupiranje upita. Prvo, što veći broj informacijskih potreba prisutnih u podacima mora biti pokriven s barem jednom grupom. Nedostatak ovog svojstva uzrokovao bi da stručnjak nepokrivene informacijske potrebe ne uključi u nastavak postupka izgradnje FAQ-zbirke, koja bi stoga ispala nepotpuna. Drugo, što manji broj informacijskih potreba trebao bi biti pokriven od strane dvije ili više grupa. Nedostatak ovog svojstva uzrokuje dodatan posao stručnjaku koji mora više puta razmatrati istu informacijsku potrebu te može dovesti do redundancije u izgrađenoj FAQ zbirci. Predložena mjera koja uvažava oba navedena svojstva grupiranja je F_1^{FAQ}, koja je inačica mjere F_1 opisane u odjeljku 2.4. Izračun ove mjere svodi se na izračun mjere F_1 na razini informacijskih potreba. Pri tome, za svaku informacijsku potrebu postoje dvije mogućnosti:

1. Nije pokrivena niti jednom grupom – u ovom slučaju se to broji kao lažno negativan (engl. *false positive*) primjer. Motivacija iza ovog pristupa jest da se kazni model ako propusti pokriti neku informacijsku potrebu, što je izravno povezano sa prvim željenim svojstvom;
2. Pokrivena je od strane N grupa – ovo se broji kao jedan stvarno pozitivan (engl. *true positive*) primjer i $N-1$ lažno pozitivnih (engl. *false positive*). Ovdje se model nagrađuje ako uspije pronaći grupu, što je u skladu s prvim željenim svojstvom. Dodatno, model se kažnjava ako tu grupu nađe više od jednom jer to uzrokuje duplikate i narušava drugo svojstvo.

Detaljan postupak računanja formaliziran je algoritmom 6. Predložena mjera F_1^{FAQ} dobro je usklađena s kvalitetom grupiranja u kontekstu opisane primjene na zadatak identifikacije informacijskih potreba kroz grupiranje upita.

Kod opisa predloženog postupka više potrebnih funkcija je, zbog općenitosti, definirano generički. Također, definirano je više hiperparametara koji utječu na rad algoritma. Prikladne implementacije za ove funkcije i vrijednosti hiperparametara određene su kroz preliminarne pokuse koji su provedeni na polovici skupa FAQIR. Kao vektorski prikaz upita koristi se prosjek PARAGRAM vektorskih prikaza (Wieting i dr., 2015) svih sadržajnih riječi u upitu. Prosjek je odabran jer su ovi vektorski prikazi optimirani da dobro rade s operacijom prosjeka, kao što je opisano u odjeljku 2.2.4. Razmatrane i konačno odabrane implementacije funkcija napravljene su u scenariju $K > K^*$ te su prikazane u tablici 7.1. U slučaju hiperparametara postupka, preliminarni pokus proveden je za sva tri scenarija. Tako su dobivena tri skupa vrijednosti za hiperparametre, koji su prikazani u tablici 7.2. Ovi skupovi razlikuju se samo u vrijednostima za GSP_{max} i GSS_{max}. Cilj ovakvog pristupa pronalasku prikladnih implementacija funkcija i vrijednosti hiperparametara nije bio iscrpna optimizacija postupka, već samo određivanje onih postavki koje daju donekle dobre rezultate na različitim skupovima podataka. Ovaj cilj je, do neke mjere, postignut, jer pronađene postavke daju prihvatljive rezultate u kasnijim pokusima na cijelom skupu FAQIR, kao i na skupu StackFAQ. To je indikacija da one općenito prihvatljivo

Algoritam 6: $F_1^{FAQ}(I,C)$

1: **Ulaz**: Skup informacijskih potreba $I = \{i_1, \ldots, i_m\}$ i skup grupa upita $C = \{c_1, \ldots, c_k\}$
2: **Inicijalizacija**:
3: $TP \leftarrow 0$
4: $FP \leftarrow 0$
5: $FN \leftarrow 0$
6: $\mathbf{F_1^{FAQ}(I,C)}$:
7: **za svaki** $i \in I$ **učini**
8: $N_i \leftarrow$ broj grupa $c \in C$ koji pokrivaju i
9: **ako** $N_i = 0$ **učini**:
10: $FN \leftarrow FN + 1$
11: **inače učini**:
12: $TP \leftarrow TP + 1$
13: $FP \leftarrow FP + N - 1$
14: $P \leftarrow \frac{TP}{TP+FP}$
15: $R \leftarrow \frac{TP}{TP+FN}$
16: $F_1^{FAQ} \leftarrow \frac{2PR}{P+R}$
17: **Izlaz**: Mjera kvalitete grupiranja F_1^{FAQ}

rade na skupovima podataka slične vrste i veličine kao što su skupovi korišteni u ovom radu. Dodatno, vrlo važno je napomenuti da su zbog korištenja dijela skupa FAQIR za pronalaženje hiperparametara postupka, rezultati na tom skupu predstavljaju optimističnu procjenu stvarnih rezultata. Zato je za realnu analizu učinkovitosti algoritma bolje razmatrati vrednovanje na skupu StackFAQ.

Zbog stohastičke prirode nekih od razmatranih postupaka važno je u obzir uzeti nestabilnost dobivenih rezultata. Kako bi se pravedno provelo vrednovanje i usporedba modela, korišten je statistički postupak *bootstrap* temeljen na ponovnom uzorkovanju. Iz podataka je generiran veći broj[5] bootstrap-uzoraka, te je na svakom od njih isproban svaki model. Rezultati navedeni u tablicama su sredine percentilnog bootstrap intervala povjerenja dobivenog na ovaj način. Dobiveni bootstrap-uzorci omogućuju i prikladno statističko testiranje razlika u učinkovitosti različitih modela.

Rezultati evaluacije za scenarij $K < K^*$ prikazani su u tablici 7.4. U ovom scenariju temeljni postupci, koji koriste točno K grupa, daju vrlo loše rezultate na oba skupa podataka. Njihov odziv je loš zato što broj grupa K nije dovoljan da pokrije sve informacijske potrebe prisutne u podacima. Razlog za njihova lošu preciznost jest to što nedovoljan broj grupa uzrokuje nakupljanje upita različitih informacijskih potreba u istu grupu, koja tada postaje nečista i

[5] U provedenom istraživanju korišteno je 100 uzoraka. Kako bismo bili sigurni da je ovaj broj uzoraka dovoljan, intervali povjerenja su prvo izračunati na prvih 50 i drugih 50 uzoraka, te je utvrđeno da su razlike u tako dobivenim procjenama vrlo male (manje od 0,001). Ovo indicira da je bootstrap empirijska procjena distribucije uzorkovanja stabilna te da je ukupno 100 uzoraka dovoljno za ispravno vrednovanje.

Tablica 7.1: Isprobane implementacije generičkih funkcija potrebnih za rad algoritma. Konačno odabrane postavke su navedene podebljano.

Funkcija	Implementacija	Opis
$dist(q_i, q_j)$	$\|q_i - q_j\|_2$	euklidska udaljenost
	1 - $\frac{q_i^T q_j}{\|q_i\|_2 \|q_j\|_2}$	**kosinusna udaljenost**
$R(c)$	$\max_{q_i \in c, q_j \in c, q_i \neq q_j} (dist(q_i, q_j))$	udaljenost najdaljih upita
	$\max_{q_i \in c} (dist(q_i, centroid(c)))$	najveća udaljenost od središta
	$\frac{1}{\|c\|} \sum_{q_i \in c} (dist(q_i, centroid(c))^2)$	**varijanca grupe**
$S(c_i, c_j)$	$\min_{q_i \in c_i, q_j \in c_j} (dist(q_i, q_j))$	udaljenost najbližih upita
	$\max_{q_i \in c_i, q_j \in c_j} (dist(q_i, q_j))$	udaljenost najdaljih upita
	$\frac{1}{\|c_i \times c_j\|} \sum_{q_i \in c_i, q_j \in c_j} (dist(q_i, q_j))$	prosječna udaljenost upita
	$\frac{1}{\|c_i\| + \|c_j\|} \sum_{q \in c_i \cup c_j} (dist(q, centroid(c_i \cup c_j))^2)$	**varijanca unije**

Tablica 7.2: Prihvatljive vrijednosti hiperparametara utvrđene u preliminarnim pokusima.

Parametar	**K <K***	**K = K***	**K >K***
α	0,1	0,1	0,1
SP_{max}	2	2	2
SS_{max}	2	2	2
SB_{max}	2	2	2
GSP_{max}	5	25	50
GSS_{max}	50	25	5

u kontekstu mjere F_1^{FAQ} lažno pozitivna. Rezultati pokazuju da već samo prvi dio predloženog postupka popravlja rezultate preko dva puta, uz prosječno 90 odnosno 104 pitanja stručnjaku na skupovima FAQIR i StackFAQ. Pokazuje se da postupak popravlja i preciznost i odziv, no pomak u mjeri odziva je veći. Ovo je očekivano, jer postupak većinom radi podjele grupa, što povećava broj čistih grupa, a time i broj pokrivenih informacijskih potreba. Problem slabije preciznosti rješava, do neke mjere, drugi korak predloženog postupka, koji uklanja značajan dio grupa koje bi se inače brojile kao lažno pozitivne. Očekivano, u rezultatima nakon drugog dijela postupka vidi se da je preciznost značajno porasla. Cijena ovoga je manji pad odziva, jer drugi dio ponekad obriše i grupe koje su bile stvarno pozitivne. No, u smislu mjere F_1^{FAQ} drugi dio se isplati, što se može vidjeti kroz apsolutni porast od 14 odnosno 7 bodova ove mjere nakon drugog dijela u usporedbi s najboljom temeljnom metodom (HAC). Ova poboljšanja su statistički značajna uz $p < 0,05$.

U tablici 7.3 prikazani su rezultati za scenarij $K > K^*$. Rezultati temeljnih metoda, koje koriste točno K grupa, u ovom su slučaju mnogo bolji nego u prethodnom scenariju, gdje je K bio premalen. Zbog vrlo velikog broja grupa za skoro svaku informacijsku potrebu postoji barem jedna dovoljno čista grupa koja ju pokriva. Posljedica ovoga je iznimno visok odziv. S druge strane, velik broj informacijskih potreba je pokriven od strane više od jednog upita. Ovo se u kontekstu mjere F_1^{FAQ} broji kao lažno pozitivan primjer, stoga je u ovom slučaju preciznost slabija. Slično kao i u prethodnom scenariju, prvi dio predloženog postupka povećava odziv sustava, no osjetno manje nego u prethodnom slučaju. Pri tome se može primijetiti blagi pad preciznosti. Analogno prethodnom slučaju drugi dio donosi značajno povećanje preciznosti na oba skupa podataka uz pripadni blagi pad odziva. Skupni učinak je ipak povećanje performansi u odnosu na najbolju temeljnu metodu (HAC), od 8 bodova mjere F_1^{FAQ} na oba skupa podataka. Razlike u odnosu na najbolju temeljnu metodu (HAC) su statistički značajne uz $p < 0,05$

Rezultati za scenarij u kojem je $K = K^*$ prikazani su u tablici 7.5. U ovom scenariju, očekivano, već i temeljne metode pokazuju vrlo dobre rezultate. Kao i u prethodnim scenarijima, prvi dio predloženog postupka popravlja odziv, dok drugi dio popravlja preciznost. Suprotno očekivanjima, u smislu mjere F_1^{FAQ} na skupu FAQIR nije postignuto značajno poboljšanje: razlika najbolje temeljne metode (HAC) i predloženog postupka nije statistički značajna uz $p < 0,05$. Razmatranje izlaza predloženog postupka pokazuje da je drugi dio postupka prerano završio te nije uspio dovoljno popraviti preciznost kako bi se ukupan efekt postupka odrazio kao povećanje mjere F_1^{FAQ}. Ovo se može vidjeti i kroz činjenicu da je broj postavljenih pitanja u drugom dijelu mnogo manji nego u prethodnim scenarijima. Ovaj problem bi se mogao riješiti pažljivijim namještanjem hiperparametara modela za ovaj scenarij i skup podataka. Na skupu StackFAQ dobiveno je poboljšanje od 19 bodova mjere F_1^{FAQ}, koje je statistički značajno uz $p < 0,05$. Valja napomenuti da je u ovom slučaju broj postavljenih pitanja u drugoj fazi veći te je sumjerljiv s onim u prvoj fazi. Ovo je u skladu s rezultatima na prethodna dva scenarija, gdje

Tablica 7.3: Vrednovanje postupka za izgradnju zbirke u slučaju kada je inicijalni broj grupa veći od stvarnog.

	Skup FAQIR				Skup StackFAQ			
Model	P	R	F	Pitanja	P	R	F	Pitanja
HAC	0,477	**0,953**	0,636	–	0,521	**0,911**	0,663	–
KMeans	0,448	0,889	0,596	–	0,515	0,883	0,651	–
SC	0,460	0,920	0,614	–	0,528	0,881	0,661	–
P1	0,412	0,978	0,578	179,95	0,524	0,924	0,669	141,29
P2	**0,594**	0,933	**0,723**	277,56	**0,633**	0,889	**0,739**	518,72

se pokazalo da je dovoljan broj pitanja u drugoj fazi ključan za dobre performanse.

Prirodno se nameće pitanje koje su razlike u ponašanju predloženog postupka s obzirom na tri predložena scenarija te koje su pritom prednosti i mane postupka na koje je potrebno obratiti posebnu pažnju. Za zbirku FAQIR najbolji rezultat od $0,766$ bodova mjere F_1^{FAQ} postignut je u scenariju $K < K^*$. S druge strane, za zbirku StackFAQ, najbolji rezultat je iznosio $0,818$ bodova mjere F_1^{FAQ} te je postignut u scenariju $K = K^*$. Ipak, ako je cilj dobiti što robustniji model zanimljiv scenarij je zapravo $K > K^*$. U tom scenariju postignut je rezultat od $0,723$ i $0,739$ bodova mjere F_1^{FAQ}, za zbirke FAQIR odnosno StackFAQ. Ako zanemarimo scenarij $K = K^*$, koji je nerealističan, ovo je najbolji rezultat za skup StackFAQ i drugi najbolji rezultat za skup FAQIR. To upućuje na to da ovaj scenarij, iako ne vodi uvijek do najboljih mogućih rezultata, vodi do prilično dobrih rezultata bez obzira na skup podataka. Stoga taj scenarij možemo smatrati robustnim izborom najprikladnijim za praktične primjename.

7.3.5. Analiza složenosti

Za primjenu predloženog postupka vrlo je korisno imati informaciju o očekivanom broju pitanja koja će biti potrebna. Budući da sama struktura skupa podataka nad kojima postupak radi može dramatično utjecati na tijek izvođenja postupka, pa tako i na broj pitanja, teško je odrediti teorijske gornje i donje granice. Ipak, moguće je provesti empirijsku analizu na dva dostupna skupa podataka, kako bi se dobile smjernice za daljnju primjenu postupka u praksi. U svrhu ovakve analize postupak je proveden na poduzorku od 30% do 100% ukupno dostupnih podataka na skupovima FAQIR i StackFAQ. Pri tome je svaki put zabilježen broj postavljenih pitanja. Broj pitanja za svaki uzorak je stohastička veličina, zbog stohastičke prirode samog postupka. Kako bi se taj problem ublažio, za svaki poduzorak je postupak pokrenut 10 puta te je kao broj pitanja

Tablica 7.4: Vrednovanje postupka za izgradnju zbirke u slučaju kada je inicijalni broj grupa manji od stvarnog.

	Skup FAQIR				Skup StackFAQ			
Model	P	R	F	Pitanja	P	R	F	Pitanja
HAC	0,380	0,190	0,254	–	0,394	0,195	0,261	–
KMeans	0,279	0,139	0,186	–	0,342	0,170	0,227	–
SC	0,278	0,139	0,186	–	0,401	0,199	0,266	–
P1	0,489	0,860	0,620	89,95	0,578	0,628	0,599	104,33
P2	**0,759**	**0,787**	**0,766**	202,12	**0,849**	**0,564**	**0,671**	314,00

Tablica 7.5: Vrednovanje postupka za izgradnju zbirke u slučaju kada je inicijalni broj grupa jednak stvarnom.

	Skup FAQIR				Skup StackFAQ			
Model	P	R	F	Pitanja	P	R	F	Pitanja
HAC	**0,654**	0,654	**0,654**	–	0,619	0,616	0,617	–
KMeans	0,536	0,535	0,535	–	0,556	0,548	0,552	–
SC	0,580	0,580	0,580	–	0,606	0,603	0,605	–
P1	0,379	**0,996**	0,547	509,48	0,609	0,919	0,732	333,24
P2	0,456	0,994	0,624	140,32	**0,759**	**0,889**	**0,818**	324,96

uzet prosjek broja pitanja po svim pokretanjima. Rezultati za skup FAQIR prikazani su na slici 7.3, a za skup StackFAQ na slici 7.4.

Uspoređujući broj pitanja za dva skupa, može se utvrditi da je broj pitanja na skupu StackFAQ najčešće veći. Ovo je posljedica činjenice da u skupu StackFAQ ima više informacijskih potreba, pa stoga i više grupa. To čini problem grupiranja zahtjevnijim i broj *mogućih* pitanja većim, a time i broj pitanja koja na kraju budu postavljena stručnjaku.

Razmatranjem broja pitanja u ovisnosti o scenariju koji promatramo rezultati su prilično dosljedni za oba skupa podataka. Scenarij $K = K^*$, obično zahtjeva najveći broj pitanja. Drugi je scenarij $K > K^*$, dok scenarij $K < K^*$ zahtjeva najmanje pitanja. Ovo ima smisla jer scenarij $K < K^*$ pretežito postavlja pitanja podjele dok scenarij $K > K^*$ pretežito postavlja pitanja spajanja. Kako su pitanja spajanja definirana nad parovima, broj mogućih pitanja spajanja je mnogo veći nego broj mogućih pitanja podjele.

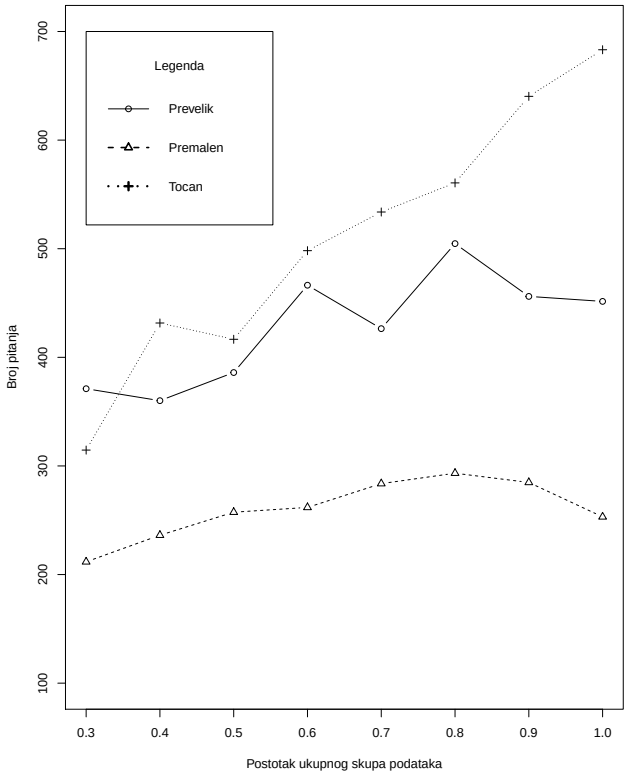
Konačno, sukladno očekivanjima, povećanjem količine podataka povećava se i broj pitanja koja postupak postavlja stručnjaku. Empirijska analiza pokazuje da je na korištenim skupovima podataka povećanje broja pitanja u odnosu na povećanje količine podataka često sublinearno ili, u najgorem slučaju, linearno. Ovakva složenost je prihvatljiva za izgradnju FAQ-zbirki reda veličine kakav imamo u našim skupovima podataka. Dodatno, velik broj FAQ-zbirki koje se pojavljuju u praksi sličnog su reda veličine, stoga postoje dobre empirijske indikacije da broj pitanja predloženog postupka nije prevelik za praktičnu upotrebu.

Ovim odjeljkom završili smo opis oba dijela postupka za grupiranje korisničkih upita aktivnim učenjem. Ovaj postupak čini tek prvu komponentu predloženog postupka za strojno-potpomognutu izgradnju FAQ-zbirke. Nakon primjene ove komponente, domenski stručnjak raspolaže skupom FAQ-parova u kojem je osmislio FAQ pitanja. Preostali posao je u svaki od FAQ-parova unijeti pripadni odgovor. Ovaj posao domenskom stručnjaku trebala bi ubrzati druga komponenta predloženog sustava – postupak za dohvat relevantnih odlomaka iz dokumentacije, koji je detaljno opisan u sljedećem odjeljku.

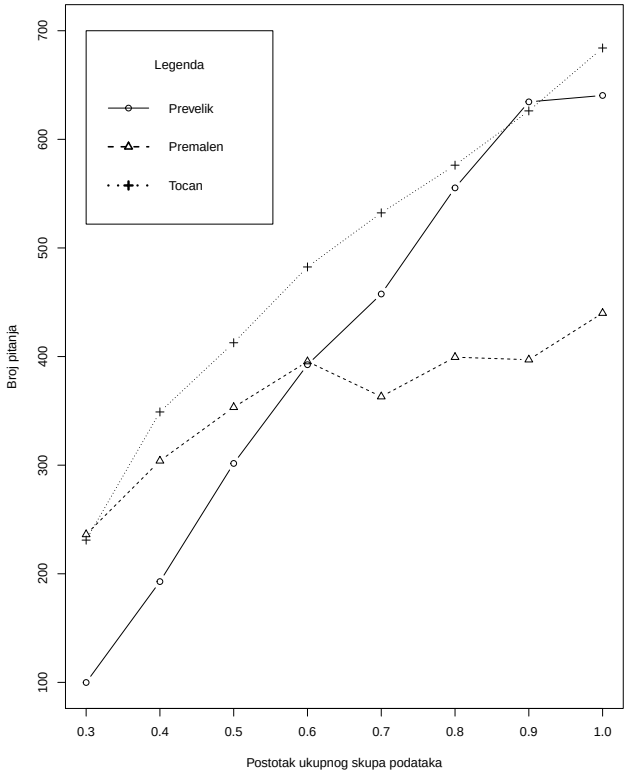
7.4. Postupak za dohvat relevantnih odlomaka

7.4.1. Pregled postupka

Druga komponenta postupka izgradnje FAQ-zbirke jest postupak za dohvat relevantnih odlomaka iz korisničke dokumentacije. Ovaj postupak gradi na rezultatu prethodne komponente. Točnije, postupak na ulazu ima skup korisničkih upita koji su grupirani u grupe. Pretpostavlja se da svaka grupa u ulaznim podacima pokriva informacijsku potrebu koja se često pojavljuje u pitanjima korisnika. Na izlazu postupka je, za svaku grupu iz ulaza, skup odlomaka iz dokumentacije sortiranih po relevantnosti za informacijsku potrebu te grupe. Cilj postupka jest da



Slika 7.3: Broj pitanja potreban na skupu FAQIR.



Slika 7.4: Broj pitanja potreban na skupu StackFAQ.

Tablica 7.6: Vrednovanje postupka za dohvat relevantnih odlomaka na temelju grupa upita.

	Skup FAQIR			Skup StackFAQ		
Scenarij	MRR	R_{OO1}	R_{OO5}	MRR	R_{OO1}	R_{OO5}
$K < K^*$	0,495	0,366	0,657	0,620	0,495	0,778
$K = K^*$	0,456	0,325	0,615	0,597	0,464	0,761
$K > K^*$	0,465	0,336	0,626	0,596	0,461	0,758

se omogući stručnjaku koji koristi sustav za izgradnju FAQ-zbirke brz i učinkovit pristup informacijama potrebnima za osmišljavanje odgovora prilikom pisanja novog FAQ-para na temelju grupe upita.

U svrhu postizanja ovog cilja postupak mora moći za grupu upita $Q = \{q_1, ..., q_N\}$ i skup odlomaka $P = \{p_1, ..., p_N\}$ dohvatiti one koji su najrelevantniji po nekom kriteriju. Pretpostavimo da je ovaj kriterij zadan preko funkcije relevantnosti $R(q, p)$, koja je proporcionalna relevantnosti odlomka p za upit q. Relevantnost odlomka p za čitavu grupu upita Q se tada računa kao:

$$R^*(Q, p) = \max_{q \in Q} R(q, p) \tag{7.4.1}$$

Izlaz modela sada je, za svaku grupu upita Q, popis svih odlomaka p silazno sortiran po veličini $R^*(Q, p)$. U preliminarnim pokusima isprobane su i alternative funkciji maksimuma, kao što su minimum, prosjek i medijan. No, funkcija maksimuma pokazala se kao najučinkovitija za dohvat relevantnih odlomaka.

7.4.2. Vrednovanje

Vrednovanje postupka provedeno je tako da je za svaku grupu upita iz grupiranja koje je izlaz prve komponente proveden opisan postupak dohvata relevantnih odlomaka. Skup odlomaka iz korisničke dokumentacije u ovom se pokusu simulira tako što se iz svakog FAQ-para u zbirci uzima samo dio s odgovorom te se smatra odlomkom dokumentacije. Tako skup odlomaka ukupno ima onoliko odlomaka koliko imamo FAQ-parova u skupu podataka. Točnije, 4.313 odlomaka za skup FAQIR i 720 odlomaka za skup StackFAQ.

Potom se za svaku grupu upita Q dohvaća niz relevantnih odlomaka kako je opisano u prethodnom odjeljku. Kao vektorski prikaz svih tekstova koristi se prosjek PARAGRAM prikaza (Wieting i dr., 2015) svih sadržajnih riječi u tekstu, slično kao u odjeljku 7.3.4. (također, v. odjeljak 2.2.4.). Kao funkcija $R(q, p)$ koristi se kosinusna sličnost ovako dobivenih vektorskih prikaza q i p. Dohvaćeni odlomci uspoređuju se sa stvarno relevantnim odlomcima za

Q. Stvarno relevantni odlomci definirani su kao oni odlomci koji su izvedeni iz FAQ-parova relevantnih za informacijsku potrebu koja je najčešća među upitima iz Q.[6]

Usporedba se provodi kroz tri mjere vrednovanja. Prvo, srednji recipročni rank (MRR), koji je već opisan u odjeljku 2.4. Dodatno, posebno prikladna je i mjera R_{ooN} – binarni odziv u prvih N rezultata (engl. *recall out of N*), predloženu u (McCarthy i Navigli, 2009). Ona se računa kao postotak grupa za koje je dohvaćen barem jedan relevantan odlomak u prvih N rezultata. Ova mjera je posebno prikladna za evaluaciju ovog zadatka jer se može intuitivno interpretirati kao mjera koliko često agent pronalazi koristan odlomak u prvih N rezultata. Računamo ovu mjeru za $N = 1$ i $N = 5$.

Pokus je proveden na izlazima modela iz prethodne komponente postupka, posebno za svaki od tri scenarija – $K < K^*$, $K = K^*$ i $K > K^*$. Rezultati pokusa prikazani su u tablici 7.6. Zbog stohastičke prirode prve komponente, oba koraka su provedena na više bootstrap-uzoraka, te je u tablicama dana sredina percentilnog bootstrap intervala povjerenja.

Općenito se može primijetiti da su rezultati prihvatljivi za praktičnu upotrebu. Na skupu FAQIR relevantan se dokument nalazi u prvih pet za preko dvije trećine grupa, dok se na skupu StackFAQ to događa za čak tri četvrtine grupa. Ovaj je rezultat dosljedan po svim scenarijima, te upućuje na to da je drugi korak predloženog postupka robustan s obzirom inicijalni broj grupa s kojim smo pokrenuli algoritam. Pri tome je vrlo važno napomenuti da za različite scenarije izlazno grupiranje prvog koraka predloženog postupka nije isto. Zato brojevi u redcima tablice 7.6 nisu izravno usporedivi. Iz istog razloga, zaključak o praktičnoj primjenjivosti drugog dijela postupka valjan je tek nakon što smo pokazali da je skup grupa na ulazu druge komponente dovoljno dobar za praktične svrhe. Da to zaista jest slučaj, već je pokazano kroz vrednovanje prve komponente kroz prethodne pokuse opisane u odjeljku 7.3.

7.5. Rasprava

Predložen je postupak izgradnje FAQ-zbirke koji se sastoji od dvije komponente. Prvo, upiti korisnika se grupiraju te se tako izdvajaju informacijske potrebe koje se često javljaju. Drugo, za svaku grupu koja odgovara čestoj informacijskoj potrebi dohvaća se skup odlomaka iz dostupne dokumentacije. Ovi podaci dovoljni su domenskom stručnjaku da za svaku čestu informacijsku potrebu napravi FAQ-par koji se odnosi na tu dotičnu potrebu. Ovim se postupkom domenskom stručnjaku olakšava i ubrzava izgradnja FAQ-zbirke koja dobro pokriva najčešće upite korisnika.

Svi algoritmi potrebni za ovaj dio istraživanja implementirani su u programskom jeziku Python. Za potrebne numeričke proračune korištena je biblioteka `numpy`.[7] Za grupiranje korištene su implementacije postupka K srednjih vrijednosti, hijerarhijskog aglomerativnog grupiranja i

[6] Smatra se da cijela grupa Q predstavlja upravo tu informacijsku potrebu.
[7] `http://www.numpy.org/`

spektralnog grupiranja, koje su dostupne u biblioteci `scikit-learn` (Pedregosa i dr., 2011).[8]

Provedeno je vrednovanje oba koraka postupka na dva skupa podataka u više scenarija korištenja. Utvrđeno je da su rezultati statistički značajno bolji od rezultata temeljnih metoda te da su prihvatljive kvalitete za praktičnu primjenu u izgradnji FAQ-zbirki.

Jedan od mogućih smjerova poboljšanja postupka kroz buduće istraživanje jest detaljnija analiza utjecaja pojedinih hiperparametara na ponašanje modela. Primjerice, mogla bi se uvesti adaptivna optimizacija hiperparametara tijekom izvođenja postupka. Takav bi pristup, osim boljih rezultata, mogao dovesti i do smanjenja broja pitanja koja sustav mora postaviti stručnjaku. Nadalje, prostor za poboljšanja može se naći i u implementaciji pojedinih operacija algoritma. Npr., sličnost dvaju korisničkih upita mogla bi se računati pomoću semantički zaglađenih jezgrenih funkcija nad sintaktičkim stablima (engl. *smoothed syntax tree kernel* - SSTK), predstavljenih u (Croce i dr., 2011), koje pri izračunu sličnosti uzimaju u obzir i sintaktička svojstva upita. Konačno, još jedan smjer mogućih poboljšanja je spajanje različitih scenarija. Postupak bi mogao paralelno razmatrati scenarije $K > K^*$ i $K < K^*$ te na taj način spojiti prednosti obje varijante, kako bi se dobili još bolji rezultati.

Ovim odjeljkom završen je opis postupka za izgradnju FAQ zbirke. U praktičnim primjenama, jednom izgrađena FAQ-zbirka je relativno statična, t.j. informacijske potrebe korisnika zbirke relativno se rijetko mijenjaju. Ipak, ponekad se može pojaviti potreba za proširenjem zbirke. Otkrivanjem takvih slučajeva, u svrhu održavanja zbirke, bavi se sljedeće poglavlje.

[8] `http://scikit-learn.org/stable/`

Poglavlje 8.

Pronalaženje nepokrivenih korisničkih upita u zbirci pitanja i odgovora

8.1. Motivacija i opis problema

Postoje slučajevi kada se skup informacijskih potreba korisnika može proširiti sa, do tog trenutka neviđenim, novim informacijskim potrebama. Primjerice, u slučaju FAQ-zbirke koja se bavi proizvodima i uslugama neke tvrtke, takav slučaj bi se dogodio u trenutku kada je uveden novi proizvod ili usluga. Takva nova informacijska potreba bila bi nepokrivena u postojećoj FAQ-zbirci.

Dok je zadatak pretraživanja FAQ-zbirki dobro obrađen u literaturi, postoji vrlo malo istraživanja na zadatku otkrivanja je li korisnički upit uopće pokriven od strane nekog FAQ-para u FAQ-zbirci. Ovaj zadatak nije trivijalan te je od posebne važnosti u kontekstu pretraživanja informacija nad domenski specifičnim FAQ-zbirkama. Razlog za to je što takve zbirke po svojoj prirodi pokrivaju ograničenu količinu informacijskih potreba. Zato nije rijedak slučaj da su rezultati pretraživanja nezadovoljavajući jer korisnička informacijska potreba nije pokrivena FAQ-zbirkom. To znatno smanjuje zadovoljstvo korisnika sustava za pretraživanje.

Ovo poglavlje razmatra problem otkrivanja slučajeva kada korisnički upit nije pokriven FAQ-zbirkom. Točnije, zadatak je klasificirati korisnički upit u jedan od dva razreda: *pokriven* ili *nepokriven*. Kvalitetno rješenje ovog zadatka donosi tri prednosti. Prvo, smanjuje broj pogrešaka zbog nepokrivenosti korisničkog upita. Drugo, pouzdano otkrivanje nepokrivenih upita može uštediti posao agentima korisničkih službi jer bi se u automatiziranoj korisničkoj službi samo oni upiti koji su nepokriveni morali proslijeđivati agentima, dok bi se ostali upiti mogli automatski odgovarati.[1] Treće, otkrivanje nepokrivenih upita bi omogućilo učinkovito

[1] U ovom slučaju postupak za otkrivanje nepokrivenih upita koristio bi se kao predfiltar u sustavu korisničke službe temeljenom na e-mail porukama. Ulazne e-mail poruke za koje sustav utvrdi da su pokrivene mogle bi biti odgovorene automatski. Ovakav bi postav istovremeno poboljšao korisničko iskustvo (vrlo brz odgovor za pokrivene upite) i učinkovitost agenata (manje ručne obrade upita).

održavanje FAQ-zbirki: model za klasifikaciju upita mogao bi se iskoristiti za otkrivanje nepokrivenih upita u logovima korisničkih upita. To bi omogućilo pravovremeno proširenje FAQ zbirke dodatnim FAQ-parovima koji adresiraju nepokrivene informacijske potrebe.

Za rješavanje ovog problema razmotreno je više postupaka temeljenih na strojnom učenju s različitim razinama nadzora. Pokazano je da je ovaj zadatak težak, ali su identificirani postupci koji daju obećavajuće rezultate.

8.2. Pregled literature

Koliko je autoru poznato, ovo je prvo istraživanje usredotočeno specifično na problem otkrivanja nepokrivenih upita u domenski specifičnim FAQ-zbirkama. Sa stanovišta strojnog učenja, ovaj je zadatak usko povezan s vrlo dobro istraženim problemom otkrivanja novina (engl. *novelty detection*). Jedan od najuspješnijih postupaka za detekciju novina je jednorazredni SVM (Schölkopf i dr., 2001). Dobar sveobuhvatan pregled postupaka za otkrivanje novina može se pronaći u (Pimentel i dr., 2014).

U pregledu literature važno je spomenuti postupke za otkrivanje pitanja u FAQ zbirci koja su duplikati. Ovaj zadatak je u osnovi dualan zadatku otkrivanja nepokrivenih upita. Dualnost proizlazi iz toga što je koncept podvostručenog pitanja zapravo inverz koncepta nepokrivenog upita. Dok prvi predstavlja nepoželjan višak u FAQ zbirci, drugi predstavlja nepoželjan manjak. Slijedeći ovu intuiciju može se vidjeti da se otkrivanje nepokrivenog upita može, do neke mjere, svesti na problem otkrivanja podvostručenog pitanja. Npr., upit bi se mogao smatrati nepokrivenim u FAQ-zbirci ako za njega nema duplikata u skupu FAQ pitanja iz zbirke. Za zadatak otkrivanja upita nepokrivenog zadanom FAQ-zbirkom nije dovoljno usporediti upit sa samo jednim FAQ pitanjem iz nje, već sa svima. Zbog toga je za određivanje pokrivenosti upita potreban dodatan korak koji agregira rezultate takvih usporedbi.

Jednostavan ali učinkovit sustav za detekciju podvostručenih pitanja predložili su Ahasanuzzaman i dr. (2016), koji koriste SVM sa značajkama temeljenim na n-gramima i leksičkosemantičkoj bazi WordNet (Miller, 1995) za klasifikaciju para pitanja sa stranice StackOverflow[2] u razrede *duplikati* ili *ne duplikati*. Napredniji postupak detekcije duplikata predlažu dos Santos i dr. (2015). Postupak se temelji na korištenju konvolucijske neuronske mreže za učenje vektorskih prikaza pitanja takvih da semantički slična pitanja imaju slične vektorske prikaze. Sličan, proširen pristup opisali su Bogdanova i dr. (2015), koji uspoređuju modele temeljene na SVM s konvolucijskim neuronskim mrežama, uz detaljne pokuse s različitim vektorskim prikazima i dodatnim prilagodbama modela konkretnoj domeni primjene.

U ovom istraživanju opisan je postupak za detekciju nepokrivenih pitanja koji prvenstveno koristi pristupe iz područja otkrivanja novina, budući da pristupi temeljeni na otkrivanju dupli-

[2] `http://stackoverflow.com`

ciranih pitanja zahtijevaju više podataka nego što je tipično dostupno u domenski specifičnim FAQ-zbirkama. Vrednuju se i uspoređuju različite varijante pristupa kako bi se odredio najbolji pristup za slučaj domenski specifičnih FAQ-zbirki kakvima se bavi cijeli ovaj rad.

8.3. Opis istraženih postupaka

Zadatak postupka za otkrivanje nepokrivenih pitanja u FAQ-zbirci je da, uz zadan korisnički upit q_{novi}, odredi spada li on u razred *pokriven* or *nepokriven*, s obzirom na FAQ-parove prisutne u FAQ-zbirci. Provode se pokusi s više inačica modela za rješavanje ovog problema. One se mogu razvrstati u (1) modele temeljene na pretraživanju informacija i (2) modele temeljene na grupiranju.

Modeli temeljeni na pretraživanju informacija. Ovo je jednostavan pristup u kojem se koristi postupak za pretraživanje informacija kako bi se dohvatio rangiran popis potencijalno relevantnih FAQ-parova za korisnički upit q_{novi}. Potom se korisnički upit svrstava u razred *pokriven* ako je stupanj relevantnosti najviše rangiranog dokumenta iznad praga t. Kao postupak pretraživanja informacija korišten je poznati postupak BM25 (Robertson i dr., 1995) i postupak pretraživanja temeljen na tf-idf utežanom vektorskom prostoru (engl. *vector space* – VS) (Manning i dr., 2008). Opisi oba ova modela mogu se pronaći u poglavlju 2.2. Prag t je optimiziran na malom izdvojenom skupu za provjeru. Ovaj pristup je jednostavan i ne zahtijeva mnogo ljudskog označavanja, pa je korišten kao temeljni postupak s kojim će se uspoređivati napredniji pristupi.

Modeli temeljeni na grupiranju podataka. U ovoj skupini pristupa podrazumijevano je da je dostupan skup upita $Q = \{q_1, \ldots, q_N\}$, za koje znamo da su pokriveni FAQ-zbirkom. Nadalje, pretpostavlja se da se zna koji od dostupnih upita zapravo pokrivaju istu informacijsku potrebu. Stoga, je upite moguće grupirati u grupe $\{C_1, \ldots, C_M\}$, gdje upiti iz iste grupe C_i predstavljaju parafraze jedinstvene informacijske potrebe, koja je pokrivena FAQ-zbirkom. Prednost ovakvog postava je što je moguće utvrditi je li korisnički upit q_{novi} pokriven zbirkom tako što se razmatra sličnost njega i njemu najsličnije grupe C_i. Ako je ta sličnost iznad praga t, korisnički upit svrstava se u razred *pokriven*, a inače se svrstava u razred *nepokriven*. Formalnije, q_{novi} je pokriven ako i samo ako $\max_i \left\{sim(q_{novi}, C_i)\right\} \geq t$, gdje je *sim* mjera sličnosti između upita i grupe upita. Isto kao i za skupinu modela temeljenih na pretraživanju informacija, prag t se optimira na malom izdvojenom skupu za provjeru.

U realističnom slučaju, nije razumno očekivati da će biti dostupne parafrazirane inačice upita koji se odnose na istu informacijsku potrebu. No, one se mogu jednostavno napraviti parafraziranjem FAQ pitanja koja se već nalaze u FAQ-zbirci. Kao što je pokazano u (Karan

i Šnajder, 2016), ovo nije vremenski zahtjevan zadatak označavanja. Za domenski specifične FAQ-zbirke, kakve tipično koriste velike tvrtke pružatelji usluga, a čija veličina tipično ne prelazi nekoliko stotina FAQ pitanja, označavanje parafraza za cijelu zbirku ostvariv je zadatak. Dodatno, performanse modela koji koriste ovu dodatnu informaciju trebale bi biti bolje od temeljnih modela. Jedno od pitanja na koje treba odgovoriti ovo istraživanje jest isplati li se dodatan trud označavanja s obzirom na dobiveno povećanje kvalitete rezultata modela.

Za rješavanje ovako postavljenog zadatka potrebno je imati (1) vektorske prikaze tekstova kao točaka u prostoru značajki i (2) postupak za klasifikaciju korisničkog upita na temelju usporedbe s grupama upita. U ovom istraživanju razmotreno je više inačica za obje ove komponente, što je detaljno izloženo u nastavku.

Pokusi su provedeni s dva moguća vektorska prikaza teksta:

- *tf-idf* – Tekstovi su prikazani kao tf-idf utežani vektori (Manning i dr., 2008). Idf komponente za riječi izračunate su na FAQ-zbirci;
- *PARAGRAM* – Vektorski prikazi teksta dobiveni su uprosječavanjem semantičkih vektora za sve sadržajne riječi u tekstu. Korišteni su semantički vektori izgrađeni u (Wieting i dr., 2015), slično kao u odjeljcima 7.3.4. i 7.4.2. (također, v. odjeljak 2.2.4.)

U kombinaciji s dva vektorska prikaza teksta razmotreno je šest pristupa za klasifikaciju korisničkog upita q_{novi}:

- COS – Sličnost korisničkog upita i grupe C_i računa se u dva koraka. Prvo, računa se kosinusna sličnost između vektorskog prikaza q_{novi} i vektorskog prikaza svakog upita u C_i. Drugo, dobivene sličnosti se agregiraju korištenjem agregacijske funkcije te se tako dobiva konačna sličnost. Operacije koje su isprobane kao agregacijska funkcija su minimum, maksimum i prosjek. Agregirana sličnost uspoređuje se s pragom t te se na temelju toga donosi klasifikacijska odluka. Odabir agregacijske funkcije i iznos praga t hiperparametri su ovog postupka;
- SVMdist – za svaku grupu uči se klasifikacijski model temeljen na jednorazrednom SVM (Schölkopf i dr., 2001), pri čemu je pozitivna klasa *pokriven*. Vektorski prikazi teksta koriste se kao vektori značajki za klasifikacijski model. Sličnost q_{novi} i C_i je definirana kao udaljenost q_i i razdvajajuće hiperravnine jednorazrednog SVM-a koji pripada grupi C_i. Motivacija za ovu mjeru jest očekivanje da će se svaki SVM specijalizirati za otkrivanje upita koji odgovaraju jednoj informacijskoj potrebi. Posljedica toga bit će da, ako q_{novi} zaista pripada u grupu C_i, pripadni SVM model će dati veliku pozitivnu udaljenost[3] na izlazu, dok će u protivnom udaljenost će biti mala ili čak negativna. Korisnički upit se klasificira kao *pokriven* ako je sličnost njega i njemu najsličnije grupe upita iznad praga t. Tip i parametri jezgrene funkcije jednorazrednih modela SVM-a za sve grupe te iznos praga t su hiperparametri te se optimiraju na izdvojenom skupu za provjeru;

[3]Primjer će biti duboko s pozitivne strane razdvajajuće hiperravnine SVM-a.

- SVMvote – Jednorazredni modeli SVM naučeni su na identičan način kao i model SVMdist. Klasifikacijska odluka razmatra odluke SVM modela za sve grupe C_i. Korisnički upit q_{novi} klasificira se kao *pokriven* ako i samo ako barem jedan od razmatranih SVM modela svrsta q_{novi} u pozitivnu klasu, tj. prepozna ga kao pripadnika grupe koju modelira;
- Gaussova mješavina (GM) – Na temelju upita svake pojedine grupe procjenjuje se multivarijatna Gaussova razdioba (MG) za tu grupu. Sličnost između q_{novi} i C_i definirana je kao iznos funkcije gustoće vjerojatnosti (engl. *probability density function* – PDF) MG procijenjene iz C_i u točki definiranoj s q_{novi}. Intuicija za ovu mjeru je ta što je očekivano da, ako q_{novi} zaista pripada u grupu C_i, tada će vjerojatnost q_{novi} u pripadnoj MG biti veća. Posljedica ovoga bit će velika vrijednost pripadne PDF u točki q_{novi};
- BSVMdist – Prilikom učenja modela SVM za pojedinu grupu C_i, primjeri iz drugih grupa mogu se smatrati negativnim primjerima u binarnom klasifikacijskom problemu. Takav model SVM-a u usporedbi s jednorazrednim modelom SVM ima na raspolaganju više primjera za učenje, pa bi zbog toga mogao postići bolje rezultate. Model BSVMdist ekvivalentan je SVMdist, ali umjesto učenja jednorazrednih modela SVM, koriste se binarni modeli SVM-a. Pozitivni primjeri učenje za model koji odgovara grupi C_i su upiti iz te grupe, dok su svi ostali upiti negativni primjeri za učenje;
- BSVMvote – istovjetan princip kojim je model SVMdist pretvoren u model SVMvote koristi se za pretvaranje modela BSVMdist u model BSVMvote.

Slično kao u prethodnom poglavlju, svi postupci potrebni za ovaj dio istraživanja implementirani su u programskom jeziku Python. Numerički proračuni implementirani su uz pomoć biblioteke `numpy`, dok su za postupke strojnog učenja korištene implementacije iz biblioteke `scikit-learn` (Pedregosa i dr., 2011).

8.4. Vrednovanje

8.4.1. Postav vrednovanja

U svrhu vrednovanja razvijenih modela prilagođen je skup FAQIR, koji je detaljno opisan u odjeljku 3.4. Izvorni skup FAQIR ima 4.313 FAQ-parova i 1.233 upita, koji predstavljaju 50 jedinstvenih informacijskih potreba. Dodatno, dostupne su binarne oznake relevantnosti za sve upite. Skup je prilagođen na sljedeći način. Prvo, izbačeni su svi FAQ-parovi koji nisu relevantni za barem jedan upit. Nakon ovog koraka u skupu je ostalo 779 FAQ-parova. Ovaj korak je potreban jer, po svojoj definiciji, FAQ-zbirke sadrže FAQ-parove koji pokrivaju *često postavljana* pitanja. Zato je malo vjerojatno da bi FAQ-par koji nije relevantan za niti jedan korisnički upit, odnosno za koji nisu postavljana korisnička pitanja, uopće postao dio FAQ-zbirke. Izbacivanjem takvih FAQ-parova skup podataka postaje realističniji. Sljedeći korak je

Tablica 8.1: Rezultati vrednovanja svih modela. Preciznost, odziv i mjera F_1 su prosjeci deset mjerenja.

Model	P	R	F_1
BM25	0,635	0,902	0,738
VS	0,635	0,896	0,737
COS-T	0,734	0,846	0,782
COS-P	0,741	0,874	**0,794**
SVMdist	0,588	**0,994**	0,733
SVMvote	0,821	0,416	0,551
BSVMdist	0,882	0,613	0,714
BSVMvote	**0,894**	0,606	0,718
GM	0,698	0,785	0,735

simulacija nepokrivenosti korisničkih upita. Od ukupno 50 informacijskih potreba koliko ih ima u skupu, za slučajno odabranih 25 su uklonjeni svi relevantni FAQ-parovi. Nakon ovog koraka, otprilike pola od početnih 1.233 upita više nije pokriveno preostalim skupom FAQ-parova. Tim upitima dodjeljuje se oznaka *nepokriven*, dok se preostalim upitima dodjeljuje oznaka *pokriven*. Konačan rezultat je skup od 1.233 upita s pripadnim oznakama pokrivenosti.

Skup upita je podijeljen na skup za učenje, skup za provjeru i skup za ispitivanje, koristeći ugniježđenu unakrsnu provjeru s pet preklopa u unutarnjoj i deset preklopa u vanjskoj petlji. Hiperparametri svih modela optimirani su tako da maksimiziraju mjeru F_1 na skupu za provjeru.[4] Nakon što su pronađeni optimalni hiperparametri modela, oni se koriste za učenje konačnog modela na uniji skupa za učenje i skupa za provjeru. Takav konačni model potom se primijenjuje na skup za ispitivanje.

8.4.2. Rezultati

Rezultati vrednovanja modela za otkrivanje nepokrivenih pitanja u FAQ-zbirkama prikazani su u tablici 8.1. Prikazan je prosjek evaluacijskih mjera po deset vanjskih preklopa ugniježđene unakrsne provjere. Model COS imao je dobre rezultate s oba razmatrana vektorska prikaza teksta, pa su za njega u tablici navedene obje inačice – COS-T koja koristi vektorski prikaz tf-idf i COS-P koja koristi vektorski prikaz PARAGRAM. Za sve ostale modele navedene su samo inačice koje koriste vektorski prikaz PARAGRAM, jer su one imale značajno bolje rezultate od onih koje su koristile vektorski prikaz tf-idf.

[4] Za modele temeljene na SVM-u, jezgrena funkcija i parametri jezgrene funkcije su optimirani za svaki model SVM-a posebno. Nakon toga je prag t optimiran združeno koristeći skup naučenih modela SVM-a.

Može se primijetiti da nenadzirani postupci temeljeni na pretraživanju informacija rade neočekivano dobro, pri čemu imaju bolji odziv nego preciznost. Između različitih pristupa iz ove porodice (BM25 i VS) nema značajnih razlika.

Razmatrajući skupinu postupaka temeljenih na grupiranju otkriva se da model SVMdist ostvaruje iznimno visok odziv, no uz cijenu vrlo niske preciznosti. Upravo suprotno je slučaj kod modela BSVMdist, koji tijekom učenja ima pristup negativnim primjerima za svaku pojedinu grupu. Sveukupno najbolja preciznost ostvarena je modelima temeljenim na binarnom SVM-u. Ovaj rezultat posljedica je toga što je kod ovih modela skup za učenje modela sadrži mnogo više negativnih nego pozitivnih primjera, što potiče modele na visoku preciznost.[5] Iz istih razloga, kod tih je modela odziv vrlo nizak, što im narušava mjeru F_1. Vrijedi napomenuti da su iznosi preciznosti i odziva navedene u tablici 8.1 dobiveni maksimizacijom mjere F_1. Prema potrebi, odnos preciznosti i odziva mogao bi se jednostavno mijenjati korištenjem drugačijeg praga t.[6] Najbolji omjer preciznosti i odziva u smislu mjere F_1 postiže model COS-P, dok je drugi najbolji model COS-T. Svi drugi modeli (uz iznimku SVMvote) imaju podjednake, ali malo slabije, performanse.

Konačno, ako bismo usporedili postupke temeljene na pretraživanju informacija i postupke temeljene na grupiranju, pokazuje se da postupci iz druge skupine rade mnogo bolje. Usporedba najboljih modela iz obiju skupina (BM25 i COS-T) pokazuje da su razlike statistički značajne ($p<0{,}01$, upareni t-test). Ovaj rezultat upućuje na to da se isplati uložiti vrijeme u označavanje parafraza pitanja u FAQ-zbirci jer njihova dostupnost značajno poboljšava rezultate.

8.5. Rasprava

Razmotreno je više modela za otkrivanje nepokrivenih upita u domenski specifičnim FAQ-zbirkama. Najbolje rezultate postigao je model temeljen na grupiranju koristeći vektorske prikaze teksta temeljene na modelu PARAGRAM. Ovaj se pristup pokazao značajno boljim od jednostavnijih pristupa temeljenih na pretraživanju informacija, ali i od složenijih pristupa temeljenih na SVM modelima. Važno je napomenuti da model traži označene parafraze za sve informacijske potrebe u FAQ-zbirci. No, dodatan trud potreban za ovo označavanje opravdan je boljim rezultatima.

Ovi rezultati čine osnovu za daljnja istraživanja zadatka otkrivanja nepokrivenih upita u FAQ-zbirkama. Postoji mnoštvo smjerova u kojima bi se istraživanje moglo nastaviti. Jedna očita mogućnost je razmatranje šireg skupa vektorskih prikaza teksta i modela klasifikacije. Nadalje, moglo bi se razmotriti kombiniranje modela visoke preciznosti i modela visokog odziva, kako bi do izražaja došle prednosti obiju vrsta modela. Konačno, moguće je spojiti ove

[5] Kada negativnih primjera ima više od pozitivnih onda će u smislu mjere F_1 biti dobar onaj model koji je dobar na negativnim primjerima. Takav model ima malo lažnih-pozitiva, tj. visoku preciznost.

[6] Iznimka su modeli temeljeni na glasanju, koji nemaju taj prag.

pristupe s pristupima za otkrivanje podvostručenih upita, kako bi se u mnoštvu novih korisničkih upita otkrili oni koji su istovremeno nepokriveni i često se javljaju. Na temelju takvih upita FAQ-zbirka bi se mogla nadopuniti prikladnim odgovorima.

Ovim odjeljkom završen je opis postupaka za otkrivanje nepokrivenih pitanja. Sljedeće poglavlje pretpostavlja da je dostupna izgrađena FAQ-zbirka te se bavi razvojem modela za semantičko pretraživanje takve zbirke.

Poglavlje 9.

Pretraživanje zbirke pitanja i odgovora

9.1. Motivacija i opis problema

FAQ-zbirke imaju brojne prednosti kao što je povećanje kvalitete korisničkog iskustva i učinkovitosti agenata korisničke službe. Tri najveće prednosti već su raspravljene u odjeljku 1.2. Prvo, FAQ-zbirke poboljšavaju korisničko iskustvo tako što korisnicima olakšavaju i ubrzavaju pristup odgovorima na njihova pitanja. Drugo, one smanjuju količinu posla za agente korisničke službe jer omogućavaju djelomičnu automatizaciju odgovaranja na korisnička pitanja. Treće, FAQ-zbirke povećavaju učinkovitost korisničke službe tako što omogućavaju agentu korisničke službe da brže pronađe odgovor na pitanje koje je slično nekom od pitanja koja su korisnici često prije postavljali.

Temelj za ostvarenje ovih prednosti jest mogućnost kvalitetnog pretraživanja FAQ zbirke. Taj zadatak je vrlo težak zbog općenitih problema s analizom prirodnog jezika kao što su njegova višeznačnost i potreba za vanjskim znanjem i logičkim zaključivanjem, kao što je opisano u odjeljku 1.1. No, jedan od problema specifičnih za područje pretraživanja tekstnih informacija, pa tako i pretraživanje FAQ-zbirki, jest *leksički jaz* (Lee i dr., 2008; Berger i dr., 2000).[1] Radi se o nedostatku preklapanja riječi između korisničkog upita i relevantnog FAQ-para. Razmotrimo sljedeći primjer.

Upit: *"Kako mogu zakrpati rupu u rezervaru svog auta?"*
FAQ pitanje: *"Popravljanje pukotine na spremniku za gorivo automobila?"*
FAQ odgovor: *"Trebat će vam sljedeći alati ......"*

Iako je FAQ-par visokorelevantan za korisnički upit, preklapanje riječi iznimno je nisko. Općenito, problemi uzrokovani leksičkim jazom izraženiji su za kraće tekstove. Još jedan čimbenik koji povećava ove probleme jest specifičan stil pisanja kakav korisnici upotrebljavaju za pisanje upita, a koji se znatno razlikuje od stila kojim se pišu FAQ-parovi (Barr i dr., 2008).

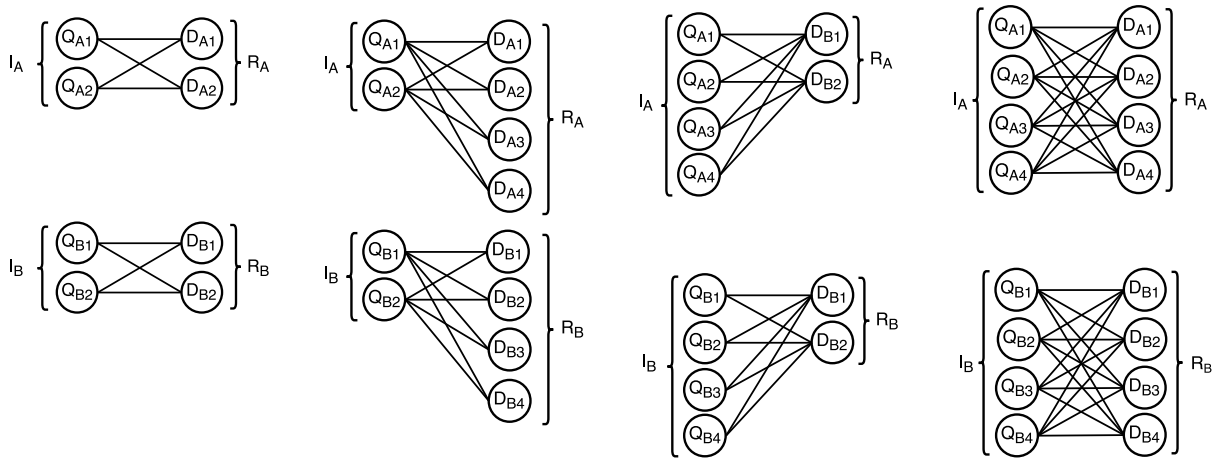
U literaturi je opisano mnogo istraživanja usredotočenih na ublažavanje problema leksič-

[1] Ovaj pojam ne bi trebalo miješati sa pojmom *leksičkog jaza* iz područja lingvistike, koji predstavlja nedostatak prikladne riječi u jeziku za nešto o čemu govornik želi govoriti (Lehrer, 1974).

kog jaza. Prema (Jeon i dr., 2005), pristupe je moguće podijeliti u sljedeće skupine: pristupe koji koriste baze znanja (Burke i dr., 1997), pristupe temeljene na ručno izgrađenim pravilima (Sneiders, 2002) te statističke pristupe (Berger i dr., 2000). Statističke pristupe dalje je moguće podijeliti u nadzirane i nenadzirane. Od nabrojanih skupina, statistički pristupi, posebno oni temeljeni na nadziranome strojnom učenju, pokazali su najbolje rezultate. Modeli statističkog strojnog učenja mogu učinkovito naučiti kako procijeniti relevantnost na temelju prisutnosti specifičnih riječi u korisničkom upitu ili u FAQ-paru. Mnogo truda uloženo je u istraživanja nadziranih modela temeljenih na *učenju rangiranja* (Agarwal i dr., 2012), posebno za zadatak odgovaranja na pitanja pomoću rada zajednice (engl. *community question answering* – CQA). Taj je zadatak povezan s pretraživanjem FAQ-zbirki, ali su za njega tipično dostupni mnogo veći skupovi podataka. Suvremeni modeli za CQA najčešće koriste bogat skup značajki za prikaz teksta, kao što su značajke generirane putem dubokih neuronskih mreža (dos Santos i dr., 2015; Severyn i Moschitti, 2015). No, koliko je autoru poznato, ne postoje istraživanja koja se bave primjenom takvih modela na zadatak pretraživanja FAQ-zbirki, gdje su podaci specifični za domenu, a ocjena relevantnosti ima malo ili ih uopće nema – situacija kakva je tipična je za FAQ-zbirke velikih pružatelja usluga.

U osnovi, nadzirani modeli za pretraživanje informacija, uključujući i modele za pretraživanje FAQ-zbirki, koriste dvije vrste zalihosti u podacima kako bi naučili preslikavanje između korisničkog upita i dokumenta (FAQ-para):

1. *Redundancija dokumenata (engl.* document redundancy – *DR)* – Nakon što su modelu bili predočeni primjeri više različitih relevantnih dokumenata za korisnički upit, model može zaključiti koje su riječi (i njima pripadajući koncepti) ključne za informacijsku potrebu tog upita. Tu informaciju model može iskoristiti kod usporedbe upita i dokumenta. Na primjer, neka je korisnički upit: *"Kako promijeniti stare pante na vratima moje spavaće sobe?"*, a relevantni FAQ-parovi (u svojim FAQ pitanjima) sadrže: *"Ugradnja šarki na kuhinjska vrata"* i *"Zamjena panti na ulazu u stan"*. Nerelevantni FAQ-parovi sadrže: *"Ugradnja zidnog ormara u spavaću sobu"* i *"Promjena kvake na vratima"*. Model može naučiti da se relevantnost za informacijsku potrebu iz korisničkog upita u FAQ-paru najviše očituje kroz prisutnost koncepta *promjene/zamjene/ugradnje* i istovremeno koncepta *šarke/panti*, dok su ostali koncepti iz upita (*stare* i *spavaća soba*) manje ključni;
2. *Redundancija upita (engl.* query redundancy – *QR)* – Ako model ima na raspolaganju parafraze istog upita, može zaključiti koje su riječi (odnosno pripadajući koncepti) ključni za informacijsku potrebu tog upita, a koje predstavljaju šum. Slično kao u prethodnom slučaju, te informacije model može iskoristiti za optimizaciju usporedbe korisničkog upita i dokumenta. Na primjer, neka je korisnički upit: *"Kako skinuti naljepnicu s prozora?"*. Neke od mogućih parafraza za ovaj upit su: *"Uklanjanje etikete s prazne staklenke mrkve"*, *"Kako ukloniti ljepljive trake sa staklene površine?"*, ili *"Kako maknuti oznaku*



(a) QR↓, DR↓ (b) QR↓, DR↑ (c) QR↑, DR↓ (d) QR↑, DR↑

Slika 9.1: Primjeri različitih vrsta redundancije koja se javlja između upita i dokumenata (FAQ-parova): redundancija upita (QR) redundancija dokumenata (DR), a mogu biti niske (↓) ili visoke (↑).

koju je moj prijatelj zalijepio za ogledalo?", Model sada može zaključiti da su, za informacijsku potrebu parafraziranog upita, najbitniji koncepti (1) *neka vrsta ljepljivog predmeta*, (2) *staklena površina* i (3) *micanje/uklanjanje/skidanje*, dok ostali koncepti iz parafraziranog upita, kao što su *prijatelj* ili *mrkva*, nisu ključni za relevantnost.

Kako bismo ilustrirali utjecaj obje vrste redundancije na modele pretraživanja, razmotrimo skup parafraziranih upita, Q_{Ai}, koji svi izražavaju istu informacijsku potrebu I_A koja je povezana s skupom relevantnih dokumenata $R_A = \{D_{A1}, \ldots, D_{AN}\}$. Nadalje, razmotrimo još jedan skup parafraziranih upita Q_{Bi}, koji odgovaraju drugoj informacijskoj potrebi I_B koja je povezana sa drugim skupom relevantnih dokumenata $R_B = \{D_{B1}, \ldots, D_{Bn}\}$. Situacija bi se mogla interpretirati kao graf, kao što je prikazano na slici 9.1, gdje prisutnost brida između čvora upita i dokumenta upućuje na relevantnost tog dokumenta za dotični upit. Nadzirani model za pretraživanje u osnovi iskorištava informaciju sadržanu u bridovima ovog grafa. Najviše informacije prenose bridovi koji su *prisutni* u grafu – nadziranom modelu oni predstavljaju pozitivne primjere za učenje, tj. parove upit-dokument gdje je dokument relevantan za upit. No, u manjoj mjeri *nedostajući* bridovi također prenose nešto informacije. Takvi bridovi nadziranom modelu predstavljaju negativne primjere za učenje, odnosno parove upit-dokument u kojima dokument *nije* relevantan za upit. U skoro svim primjenama većina parova upit-dokument će predstavljati negativne primjere za učenje, pa će oni biti dostupni u velikim količinama. Zadatak modela za pretraživanje je, u osnovi, naučiti kako uspješno razlučiti malen broj pozitivnih primjera od većine ostalih koji su negativni. Kako bi model mogao uspješno naučiti rješavati ovaj zadatak, ključno je da postoji dovoljan broj pozitivnih primjera, što se izravno očituje kao povećanje broja bridova u grafu. To se može postići ili označavanjem više parafraza za upite (povećanje QR-a), ili označavanjem više relevantnih dokumenata za svaku informacijsku potrebu (povećanje DR-a).

Obje su vrste redundancije prisutne, do neke mjere, u svim zadacima iz područja pretraživanja informacija, no redundanciju upita često je teže izravno iskoristiti. Konkretno, u većini praktičnih primjena pretraživanja informacija parafraze nisu eksplicitno dostupne. Iako ponekad u skupovima podataka za pretraživanje informacija može biti slučaj da su dva ili više upita parafraze jedan drugoga, ta informacija nije nigdje eksplicitno označena, pa je stoga modeli pretraživanja ne mogu iskoristiti. Nadalje, u rijetkim slučajevima kada u skupu podataka postoje oznake parafraza, npr., u skupu podataka CQA iz (Hoogeveen i dr., 2015), broj parafraza za pojedinu informacijsku potrebu vrlo je nizak. Ovo je očekivana pojava jer većina velikih CQA stranica aktivno potiče izbjegavanje pitanja koja su duplikati.

Postoje dvije važne razlike između pretraživanja domenski specifičnih FAQ-zbirki i općenitog pretraživanja teksta. Prvo, ograničena domena i relativno mala veličina FAQ zbirki čine skup jedinstvenih informacijskih potreba relativno malenim. Zahvaljujući tome, označavanje parafraza za sve informacijske potrebe iz zbirke (u smislu uloženog ljudskog rada) je izvedivo. Drugo, informacijske potrebe korisnika relativno su statične. Zbog ovog su parafraze upita vrlo *praktične* jer ih nije potrebno često osvježavati. Zadatak pretraživanja FAQ-zbirki u ovom je smislu usko povezan sa zadatkom CQA, no od njega se razlikuje tako što su FAQ-zbirke najčešće usredotočene na mnogo užu domenu od CQA, kao i po tome što su FAQ pitanja obično kraća od pitanja u CQA.

Gornje razmatranje daje motivaciju za tri vrste strategija za označavanje podataka s obzirom na količinu ljudskog truda koju je potrebno uložiti.

- *Potpuna strategija* – Rezultira velikim količinama oznaka relevantnosti (engl. *relevance judgments*) i parafraza upita. Tako izgrađen skup podataka ima visoku razinu obje vrste redundancije, kao što je prikazano na slici 9.1d. S obzirom na trud označivača, ova strategija zahtijeva najviše vremena;
- *Strategija usmjerena na relevantnost* – Kod ove strategije označava se malo parafraza ili se one uopće ne označavaju. No, označava se velik broj ocjena relevantnosti, što rezultira visokom redundancijom dokumenata. Ova je situacija prikazana na slici 9.1b. Ova strategija zahtijeva marginalno manje vremena nego prethodna, ali je i dalje vrlo vremenski zahtjevna;
- *Strategija usmjerena na parafraze* – Ova strategija stavlja naglasak na označavanje velikog broja parafraza i manjeg broja ocjena relevantnosti. Rezultat je skup s visokom razinom redundancije upita i manjom razinom redundancije dokumenata, kao što je ilustrirano na slici 9.1c.

Rezultat sve tri strategije jest označen skup podataka koji služi za učenje modela pretraživanja temeljenih na nadziranome strojnom učenju. Nadalje, sve tri strategije unose obje vrste redundancije u skup podataka, no one se razlikuju po omjeru redundancije upita i redundancije dokumenata.

Model pretraživanja učen koristeći skup podataka označen potpunom strategijom trebao bi davati najbolje rezultate. Ipak, za većinu praktičnih primjena izgradnja takvog skupa trajala bi predugo. Druge dvije strategije smanjuju vrijeme izgradnje, pa su zato u praksi zanimljivije. Međutim, strategija usmjerena na relevantnost je i dalje vrlo vremenski zahtjevna: označavanje ocjena relevantnosti uključuje pažljivo čitanje i razumijevanje duge liste FAQ-parova za svaki upit, često od strane više od jednog označivača, što je dugotrajno i skupo. S druge strane, strategija usmjerena na parafraze za označivače je osjetno manje zahtjevna: umjesto naglaska na označavanju relevantnosti, više pažnje posvećeno je označavanju parafraza, što je nedvojbeno mnogo jednostavniji zadatak. Ovo je potvrđeno u (Karan i Šnajder, 2016), gdje je označavanje parafraza za informacijsku potrebu bilo u prosjeku deset puta brže nego označavanje ocjena relevantnosti. Dakle, izgradnja modela pretraživanja učenog na skupu podataka označenom strategijom usmjerenom na parafraze ima jasne praktične prednosti. Pritom je uvjet da takav model zadrži performanse usporedive s modelima učenim na podacima koji su nastali primjenom potpune strategije. Jedan način implementacije strategije usmjerene na parafraze mogao bi biti da se provede označavanje parafraza jednako kao kod potpune strategije, a da se istovremeno uvedu mehanizmi za smanjenje broja oznaka relevantnosti koje treba označiti.

Ovaj dio rada istražuje potencijal korištenja nadziranih modela pretraživanja temeljenih na učenju rangiranja u kombinaciji sa strategijama usmjerenim na parafraze. Pritom je cilj povećanje učinkovitosti pretraživanja domenski specifičnih FAQ-zbirki uz male količine dodatnog truda uloženog u označavanje podataka. Razmatraju se sljedeća istraživačka pitanja:

1. Može li model pretraživanja temeljen na nadziranom učenju rangiranja i učen na skupu podataka označenom potpunom strategijom doseći u praksi značajna poboljšanja učinkovitosti u usporedbi s nenadziranim pristupima za pretraživanje;
2. Može li se usporediva učinkovitost postići ako se koristi skup podataka označen strategijom usmjerenom na parafraze, koja zahtijeva manje oznaka relevantnosti.

Vrijedi spomenuti da, u kontekstu označavanja podataka za učenje rangiranja, postoji mnogo alternativnih pristupa za smanjenje rada označivača. Glavni pristupi su aktivno učenje (Brinker, 2004; Huang i dr., 2008), polunadzirano učenje (Yang i dr., 2012; Duh i Kirchhoff, 2008) i označavanje radom mnoštva (engl. *crowdsourcing*) (Alonso i dr., 2008; Kazai i dr., 2013; Abraham i dr., 2016). Za razliku od ovih pristupa, u ovom radu naglasak je stavljen na strategiju označavanja koja jedan zadatak označavanja (označavanje relevantnosti) mijenja drugim, lakšim zadatkom (parafraziranje upita). Predložena strategija je u osnovi komplementarna navedenim alternativama.

U ovom dijelu istraživanja prvo je pokazano da, uz skup podataka označen potpunom strategijom, dva nadzirana modela pretraživanja – LambdaMART i konvolucijska neuronska mreža (engl. *convolutional neural network* – CNN) – daju u praksi značajna poboljšanja u usporedbi s nenadziranim temeljnim modelima pretraživanja. Nakon toga, istražene su dvije strategije

usmjerene na parafraze, koje su zatim isprobane u kombinaciji s modelima pretraživanja. Pokazano je da je učinkovitost modela pretraživanja učenih na podacima označenim ovim strategijama sumjerljiva učinkovitosti istih modela učenih na podacima označenim potpunom strategijom. Ovi rezultati pokazuju da modeli pretraživanja FAQ-zbirki temeljeni na nadziranom učenju rangiranja mogu dati značajno kvalitetnije rezultate od nenadziranih temeljnih modela pretraživanja. Ovo je slučaj čak i uz strategije označavanja koje zahtijevaju vrlo malo truda označivača. Svi su pokusi provedeni koristeći dva skupa podataka za pretraživanje domenski specifičnih FAQ-zbirki, koji su izgrađeni da budu reprezentativni za situacije kakve se javljaju u praktičnoj primjeni.

U nastavku ovog poglavlja prvo je dan pregled literature na temu pretraživanja FAQ-zbirki. Potom slijedi detaljan opis i tehnički detalji svih korištenih modela. Nakon toga opisan je postav pokusa te su dani rezultati i dobiveni zaključci. Poglavlje završava kratkom raspravom i smjernicama za moguć nastavak istraživanja.

9.2. Pregled literature

9.2.1. Pretraživanje FAQ-zbirki

Zadatak pretraživanja FAQ-zbirki nalazi se na presjecištu područja pretraživanja informacija (engl. *information retrieval* – IR) i područja odgovaranja na pitanja (engl. *question answering* – QA), pa mu se može pristupiti korištenjem tehnika iz oba područja. Za razliku od tradicionalnog QA (Hirschman i Gaizauskas, 2001), kod kojega treba generirati precizan odgovor na pitanje, u pretraživanju FAQ-zbirki dovoljno je dohvatiti FAQ-par iz zbirke koji u sebi sadrži odgovor. Zato većina pristupa pretraživanju FAQ-zbirki naginju više prema tehnikama iz područja IR. Blisko povezan zadatak je odgovaranje na nečinjenična pitanja (engl. *non-factoid question answering*)[2] (Surdeanu i dr., 2008), gdje je zadatak dohvatiti odsječak teksta koji u sebi sadržava odgovor na pitanje korisnika. No, postoje dvije razlike odgovaranja na nečinjenična pitanja i pretraživanja FAQ-zbirki: (1) odgovaranje na nečinjenična pitanja ostvaruje preslikavanje s upita na odgovor, dok pretraživanje FAQ-zbirki ostvaruje preslikavanje s upita na FAQ-par i (2) tipična FAQ-zbirka je manja i usredotočena je na specifičniju domenu nego što je to slučaj kod zbirki koje se koriste za odgovaranje na nečinjenična pitanja. Nadalje, tipični sustav za nečinjenično QA mora pronaći relevantne odsječke teksta u većem tekstu, dok su kod pretraživanja FAQ-zbirki takvi odsječci unaprijed pripremljeni u obliku FAQ-parova.

Svojstvo specifično za pretraživanje FAQ-zbirki jest da se FAQ-par sastoji od dva dijela – FAQ pitanja i FAQ odgovora. Ovo omogućava više načina izračuna sličnosti između FAQ-para i

[2]Za razliku od činjeničnih pitanja, npr. *Kad je rođen J.R.R. Tolkien?*, čiji je odgovor činjenica (u ovom slučaju datum – 3. siječnja 1892.), nečinjenična pitanja nemaju jednostavan činjenični odgovor. Primjeri nečinjeničnih pitanja su *Kako ispeći štrukle?* ili *Zašto je nezaposlenost visoka?*

korisničkog upita. Korisnički upit može se usporediti (1) samo s FAQ pitanjem, (2) samo s FAQ odgovorom, (3) konkatenacijom FAQ pitanja i FAQ odgovora ili (4) posebno s FAQ pitanjem i s FAQ odgovorom uz naknadnu agregaciju dvije dobivene sličnosti. U okviru ovog istraživanja razmatrane su mogućnosti (1) i (3), koje su najčešće korištene u srodnim istraživanjima (Wu i dr., 2005; Sneiders, 2009; Contractor i dr., 2010; Karan i Šnajder, 2016).

9.2.2. Rana istraživanja

Jedan od prvih sustava za pretraživanje FAQ-zbirki bio je FAQFinder (Burke i dr., 1997). Sustav je preslikavao korisničke upite na FAQ-parove koristeći vektorske prikaze teksta temeljene na shemi tf-idf (v. odjeljak 2.2.4.). komplementirane informacijama iz leksičkosemantičke baze WordNet (Miller, 1995). U ranim natjecanjima iz QA, koja su organizirana u sklopu konferencije *Text Retrieval Conference* – TREC (Voorhees, 1999) pokazalo se da temeljni modeli iz područja IR daju dobre rezultate. Različiti pristupi temeljeni na IR uključivali su jezično modeliranje (engl. *language modeling*) (Jeon i dr., 2005), udaljenost uređivanja (engl. *edit distance*) (Kothari i dr., 2009), modeliranje vektorskim prostorom (Jijkoun i de Rijke, 2005), sličnost temeljenu na analizi korpusa (engl. *corpus based similarity*) (Marom i Zukerman, 2009) i statističko strojno prevođenje (Riezler i dr., 2007). Karan i Šnajder (2016) daju pregled nenadziranih temeljnih IR modela za pretraživanje FAQ-zbirki.

Problemom leksičkog jaza u kontekstu pretraživanja FAQ-zbirki bavi se Sneiders (1999), koji je kombinirao kontrolirani rječnik (engl. *controlled vocabulary*) s pravilima koja su za svaki FAQ-par određivala koje riječi upit *mora*, a koje *ne smije* sadržavati da bi bio relevantan. Ovi su metapodaci korišteni za kvalitetniju usporedbu upita i FAQ-parova. Pristup je dalje razrađen u (Sneiders, 2009) i prilagođen za automatsko odgovaranje na poruke e-pošte u (Sneiders, 2010). Iako učinkovit, ovakav pristup zahtijeva ručnu izradu pravila za svaki FAQ-par, što može biti vrlo vremenski zahtjevno. Problem su, do neke mjere, zaobišli Moreo i dr. (2013), koji koriste parafraze upita kako bi automatski inducirali regularne izraze. Ti izrazi zapravo predstavljaju jednostavna pravila koja se koriste u modelu pretraživanja FAQ-zbirki. Alternativan pristup predložen je u (Kim i Seo, 2006), gdje je primijenjeno grupiranje na logovima klikova (engl. *click logs*) kako bi se generirala pravila za poboljšanje sustava.

U ovom istraživanju koriste se nenadzirani IR modeli tipični za rana istraživanja kao temeljni modeli s kojima se uspoređuju predloženi pristupi pretraživanju FAQ-zbirki. Ne koristi se nikakva specifična tehnika za ublažavanje leksičkog jaza, već se umjesto toga očekuje da nadzirani modeli iz označenih podataka nauče kako ga najučinkovitije premostiti.

9.2.3. Pristupi temeljeni na značajkama sličnosti

U novijoj literaturi pristupi temeljeni na nadziranom učenju rangiranja (Agarwal i dr., 2012) pokazali su na zadatcima odgovaranja na pitanja izvrsne rezultate. Takvi su sustavi često temeljeni na nadziranom modelu pretraživanja učenom na velikom broju raznovrsnih, često lingvistički motiviranih značajki sličnosti između korisničkog upita i dokumenta (u slučaju pretraživanja FAQ-zbirki dokument je FAQ-par). Pregled najčešće korištenih značajki može se naći u (Agirre i dr., 2013; Han i dr., 2013).

Jednostavniji QA sustavi obično se oslanjaju na značajke temeljene na sličnosti površinskih oblika riječi. Jedan takav primjer je sustav opisan u (Bickel i Scheffer, 2004), gdje je prikaz teksta temeljen na shemi tf-idf korišten u kombinaciji s klasifikacijskim modelom SVM. Još jedan primjer je sustav opisan u (Arora i dr., 2015), koji koristi BM25 (Robertson i dr., 1995), jezične modele i tf-idf prikaze teksta za dohvat sličnih pitanja u CQA.

Dobar primjer sustava temeljenog na nadziranom učenju rangiranja za nečinjenično QA je (Surdeanu i dr., 2008). Kao modeli za pretraživanje koriste se klasifikacijski model SVM-a i varijanta perceptrona prilagođena rangiranju. Korišten je bogat skup značajki od kojih su neke lingvistički motivirane a neke opisuju površinsku sličnost. One uključuju preklapanje riječi, podudaranje ovisnosnih sintaktičkih relacija, značajke temeljene na strojnom prevođenju, značajke temeljene na frekvencijama riječi i značajke temeljene na korelaciji rezultata web-tražilice. Sustav postiže značajna poboljšanja u usporedbi s nenadziranim temeljnim modelom. Sličan skup značajki koristi se u kombinaciji s logističkom regresijom i stablima odluke u (Yih i dr., 2013) te u kombinaciji sa SVM-om u (Bunescu i Huang, 2010). Sustav za CQA temeljen na sintaktičkim značajkama u kombinaciji s modelima pretraživanja ListNET (Cao i dr., 2007), LambdaRank (Burges i dr., 2007) i novim modelom online SVMRank predložili su Carmel i dr. (2014).

Alternativan smjer istraživanja usmjeren je na poboljšanje sposobnosti sustava za odgovaranja na pitanja da prepoznaju parafraze upita. Figueroa i Neumann (2013) predložili su nadzirani sustav temeljen na učenju rangiranja za CQA. Sustav rangira parafraze upita po njihovoj učinkovitosti da dohvate relevantne odgovore. Model za rangiranje koristi značajke temeljene na površinskoj sličnosti (npr., udaljenost stringova), ali i složenije lingvističke značajke temeljene na leksičkosemantičkoj bazi WordNet (Miller, 1995) i sintaktičkoj analizi. U nastavku tog istraživanja Figueroa i Neumann (2014) otkrili su da izgradnja modela pretraživanja specijaliziranih za pojedine kategorije na stranicama CQA može još više poboljšati rezultate pretraživanja. Ovaj rezultat potiče istraživanja u području prilagodbe modela pretraživanja užoj domeni. Nedavno, Figueroa (2017) je predložio sustav koji automatski generira parafraze upita iz leksikaliziranih ovisnosnih sintaktičkih stabala pitanja. Generirane parafraze tada rangira nadzirani model koji, uz prethodno nabrojane značajke, koristi i značajke temeljene na prepoznavanju imenovanih entiteta i analizi sentimenta. Cilj nadziranog modela jest odrediti najinformativniju parafrazu te

tako efektivno prepraviti korisnički upit u alternativnu formulaciju koja će dati bolje rezultate pretraživanja.

9.2.4. Pristupi temeljeni na jezgrama stabala

Pristupi temeljeni na jezgrama stabala (engl. *tree kernel*) kombiniraju jezgre stabala s jezgrenim inačicama algoritama strojnog učenja, najčešće SVM-om. Jezgre stabala zapravo su mjere sličnosti koje uspoređuju korisnički upit i FAQ-par tako što parsaju oba teksta i računaju sličnost na dobivenim sintaktičkim stablima. Sličnost uključuje sintaktičku informaciju iz stabla, ali postupak se može nadograditi tako da uzima u obzir semantičku sličnost riječi. Za razliku od pristupa koji koriste ograničen skup lingvističkih značajki, sintaktička i semantička informacija uključena u izračun jezgre stabala daje ovim modelima implicitan pristup iznimno bogatom skupu značajki. Posljedično, ovi modeli uče na vrlo bogatim prikazima teksta, što se često očituje kroz osjetno bolje rezultate. Jezgre stabala uspješno su primijenjene za preslikavanje pitanja na odgovore u (Moschitti i Quarteroni, 2011). Sustav koji su razvili Filice i dr. (2016), a koji koristi kombinaciju raznih jezgara stabala i dodatnih značajki specifičnih za zadatak, osvojio je prvo mjesto na zadatku CQA natjecanja Semeval 2015 (Moschitti i dr., 2015). Sličan algoritam temeljen na usporedbi sintaktičkih stabala predložen je i u (Wang i dr., 2009).

9.2.5. Pristupi temeljeni na neuronskim mrežama

Neuronske mreže u okviru paradigme dubokog učenja (engl. *deep learning*) (Goodfellow i dr., 2016) pokazale su se kao vrlo uspješne u velikom broju zadataka strojnog učenja. Takvi zadaci uključuju obradu slike (Park i Kim, 2013; Alonso-Weber i dr., 2014) i obradu prirodnog jezika (Moraes i dr., 2013; Ghiassi i dr., 2012). Od mnogih predloženih arhitektura najpopularnije i najuspješnije su konvolucijske neuronske mreže (engl. *convolutional neural networks* – CNN) (Krizhevsky i dr., 2012) i dugačka mreža kratkotrajnog pamćenja (engl. *long short-term memory networks* – LSTM) (Hochreiter i Schmidhuber, 1997). Posebno se ističu arhitekture CNN, koje su pokazale izvrsne rezultate za zadatke klasifikacije i rangiranja tekstnih podataka (Severyn i Moschitti, 2015; Feng i dr., 2015; Qiu i Huang, 2015).

Postoji velik broj radova koji opisuju pristupe za odgovaranje na pitanja i srodne zadatke temeljene na neuronskim mrežama. Rekurzivna neuronska mreža za preslikavanje pitanja na odgovore predložena je u (Iyyer i dr., 2014), dok je dvosmjerna mreža LSTM uspješno primijenjena u (Nassif i dr., 2016) za isti zadatak, ali na skupu podataka iz područja CQA. Za ovaj zadatak mogu se pronaći i brojni pristupi temeljeni na arhitekturama mreže CNN (dos Santos i dr., 2015; Severyn i Moschitti, 2015; Qiu i Huang, 2015). Najčešće se koriste dva odvojena konvolucijska sloja kako bi se za pitanje i potencijalni odgovor dobili vektorski prikazi fiksne duljine. Daljnji slojevi mreže rade operaciju usporedbe nad tim prikazima. Operacije usporedbe

koje se mogu naći u literaturi uključuju jednostavnu kosinusnu sličnost vektora (dos Santos i dr., 2015), matricu sličnosti kombiniranu sa skrivenim slojem (Severyn i Moschitti, 2015) i tenzorski sloj (Qiu i Huang, 2015).

Lei i dr. (2016) nedavno su predložili postupak za pronalaženja pitanja koja su parafraze jedni drugih temeljen na neuronskim mrežama. Predloženi pristup koristi povratnu (engl. *recurrent*) arhitekturu CNN kako bi preslikali pitanja u vektorske prikaze takve da semantički slična pitanja imaju slične prikaze. Model uči u dva koraka. U prvom koraku model uči prepoznavati sličnost između tijela pitanja i naslova tog istog pitanja na vrlo velikom skupu pitanja. U drugom koraku model dodatno uči prepoznavati parafraze pitanja na manjem skupu pitanja koja su ručno označena kao parafraze. Lei i dr. utvrdili su da predložena arhitektura mreže za ovaj zadatak radi bolje od svih ostalih razmatranih arhitektura.

U ovom istraživanju, razmatramo CNN kao jedan od više mogućih modela temeljenih na nadziranom učenju rangiranja. Najbliži srodni rad je (Severyn i Moschitti, 2015), s dvije razlike. Prvo, taj model preslikava korisnički upit na odgovor, dok razmatrani model preslikava korisnički upit na FAQ-par. Drugo, i mnogo važnije, naglasak u (Severyn i Moschitti, 2015) jest optimizacija samog modela, dok je naglasak ovog istraživanja optimizacija postupka označavanja podataka (u smislu smanjenja potrebnog vremena)..

9.2.6. Semantička sličnost i vektorski prikazi riječi

U osnovi, dohvat relevantnog FAQ-para za zadani upit svodi se na zadatak izračunavanja semantičke sličnosti između dvaju tekstova. Zadatak izračunavanja semantičke sličnosti tekstova dobro je istražen problem u obradi prirodnog jezika (Agirre i dr., 2012). Suvremeni pristupi ovom problemu grade semantičke vektorske prikaze riječi iz velikih korpusa (Turney i Pantel, 2010). Takvi modeli predstavljaju svaku riječ kao vektor koji se računa iz konteksta u kojima se riječ pojavila, kao što je već opisano u odjeljcima 2.2.3. i 2.2.4.

Postoje dva glavna načina na koje se vektorski prikazi riječi mogu iskoristiti za zadatke odgovaranja na pitanja i pretraživanja FAQ-zbirki. Prvo je oslanjanje na aditivnu kompozicionalnost vektorskih prikaza riječi, koja kaže da se vektor većeg teksta može izgraditi zbrajanjem vektora pojedinih riječi u tekstu. Takvi vektorski prikazi korisničkog upita i FAQ-para tada se mogu usporediti kosinusnom sličnosti. Drugo, vektorski prikazi teksta mogu biti ulazne značajke za neki od nadziranih modela pretraživanja. Na primjer, Yang i dr. (2016) i Sharp i dr. (2015) koriste uparene sličnosti riječi iz upita i dokumenata kao značajke za zadatak nečinjeničnog QA. Vektorski prikazi riječi također su tipično ulaz u kompleksnije neuronske modele kao što su CNN i LSTM. Dodatna prednost takvog pristupa jest što se vektorski prikazi riječi mogu prilagođavati tijekom učenja zadatka, pa tako nastaju vektorski prikazi riječi posebno prilagođeni zadatku.

U ovom radu koriste se semantički vektorski prikazi riječi kao ulaz u model pretraživanja

zasnovan na CNN-u i za izračunavanje nekih značajki za ostale modele pretraživanja. Također, koristi se i kosinusna sličnost između agregiranih vektorskih prikaza riječi kao jedan od nenadziranih temeljnih modela pretraživanja (vidjeti poglavlje 2.2.4.).

9.3. Nadzirani postupci za pretraživanje FAQ-zbirke

U ovom odjeljku opisani su modeli pretraživanja korišteni u pokusima. Neka je Q skup svih upita a F skup svih FAQ-parova. Model pretraživanja FAQ-zbirki kao ulaz prima korisnički upit $q_i \in Q$ i FAQ-par $f_j \in F$, a na izlazu daje ocjenu relevantnosti $h(q_i, f_j)$ za f_j s obzirom na q_i.

9.3.1. Nenadzirani modeli pretraživanja

Nenadzirani modeli pretraživanja ne trebaju označene podatke. Koriste se kao temeljni modeli i za agregiranje rezultata više modela pretraživanja kod strategije usmjerene na parafraze. Pokusi uključuju sljedeće modele pretraživanja:

- *BM25* – Model koji koristi poznatu BM25 formulu (Robertson i dr., 1995) za izračun ocjena relevantnosti $h(q_i, f_j)$. Više detalja o ovom modelu može se naći u odjeljku 2.2.2.;
- *VS* – Tradicionalan model pretraživanja temeljen na vektorskom prostoru (Salton i dr., 1975), kakav je opisan u odjeljku 2.2.1. Model koristi tf-idf utežani vektorski prikaz za q_i i f_j. Ocjena relevantnosti $h(q_i, f_j)$ računa se kao kosinusna sličnost između tih vektorskih prikaza. Pritom su IDF komponente za sve riječi izračunate na skupu svih FAQ-parova;
- *SG* – Model koji koristi semantičke vektorske prikaze riječi kako bi izgradio vektorske prikaze q_i i f_j. Koriste se 300-dimenzijski semantički vektori dobiveni u (Mikolov i dr., 2013) skip-gram postupkom na korpusu GoogleNews.[3] Postupak je detaljno opisan u odjeljku 2.2.4. U skladu sa standardnom praksom (Wieting i dr., 2015) i slično kao u srodnim istraživanjima (Karan i Šnajder, 2016), koristi se aditivna kompozicionalnost semantičkih vektora. Vektori većih tekstova dobivaju se zbrajanjem vektora sadržajnih riječi u njima. Ovako se dobivaju 300-dimenzijski vektorski prikazi q_i i f_j, pa se ocjena relevantnosti $h(q_i, f_j)$ može izračunati kao kosinusna sličnost ovih vektorskih prikaza;
- *RC* – Jednostavan ansambl koji kombinira izlaze prethodno opisana tri modela. Ocjena relevantnosti $h(q_i, f_j)$ je za ovaj model jednaka zbroju rangova koje je f_j imao za q_i prema prethodna tri modela. O ovom slučaju niži iznos ocjene relevantnosti znači veću relevantnost. Vrijedi napomenuti da korištenje rangova umjesto apsolutnih iznosa ocjena relevantnosti rješava problem razlike skala na kojima se nalaze ocjene relevantnosti za različite modele.

[3] Dostupno na poveznici `https://code.google.com/archive/p/word2vec/`

9.3.2. Nadzirani modeli pretraživanja

Modeli temeljeni na nadziranom učenju rangiranja zahtijevaju postojanje oznaka relevantnosti. Neka je $y_{ij} \in \{0,1\}$ binarna oznaka relevantnosti za FAQ par f_j s obzirom na korisnički upit q_i. Ako je f_j relevantan za q_i, onda je $y_{ij} = 1$, inače je $y_{ij} = 0$. Kada postoje ovakve oznake, učenje rangiranja može se provesti na tri načina: *po točkama* (engl. *pointwise*), *po parovima* (engl. *pairwise*) i *po listama* (engl. *listwise*) (Agarwal i dr., 2012).

- *po točkama* – Kod ovog načina učenja rangiranja ulaz u model je par upita i dokumenta, a željeni izlaz je binarna odluka o tome je li upit relevantan za dokument. Ovaj način učenja najčešće se implementira kao klasifikator koji klasificira par upita i dokumenta u klase *relevantan* ili *nerelevantan*. Popis rezultata za upit se gradi tako da se dokumenti poredaju po vjerojatnosti koju im je klasifikator pridijelio za klasu *relevantan*;
- *po parovima* – U ovom slučaju ulaz u model je trojka upita i dva dokumenta, dok je željeni izlaz odluka o tome koji je od dva dokumenta relevantniji za upit. Za razliku od prethodnog, ovaj način učenja uzima u obzir informaciju o relativnom međusobnom položaju dokumenata u željenom ispravnom poretku. Popis rezultata se gradi na način da se prikupe odluke modela za sve moguće parove dokumenata uz ulazni upit. Ocjena za dokument definirana je kao broj puta koliko ga je model proglasio relevantnijim. Potom se dokumenti poredaju silazno po ocjenama. Ovakvi modeli tipično optimiraju funkciju cijene temeljenu na kažnjavanju inverzija u poretku po svim mogućim trojkama upita i dva dokumenta;
- *po listama* – Ovaj način učenja rangiranja je najnapredniji. Ulaz u model je upit, a željeni izlaz je cijeli ispravno poredani popis dokumenata. Za razliku od prethodnog načina, koji uključuje informaciju o relativnom položaju samo za parove dokumenata, ovaj način učenja združeno razmatra cijeli izlazni popis. To se najčešće ostvaruje kroz optimiranje funkcije gubitka koja je definirana nad cijelim izlaznim popisom.

U pokusima provedenim u ovom radu razmatramo tri modela pretraživanja temeljena na nadziranome strojnom učenju rangiranja. Dva modela temeljena su na lingvistički motiviranim značajkama sličnosti tekstova, dok je jedan model temeljen na dubokoj neuronskoj mreži. Prva dva modela pripadaju u skupinu koja uči *po listama*, dok treći pripada u skupinu koja uči *po točkama*. Ovi su modeli detaljnije opisani u nastavku.

Modeli temeljeni na značajkama sličnosti

Zbog njihovih dobrih rezultata u srodnim istraživanjima (Surdeanu i dr., 2011; Figueroa i Neumann, 2013, 2014; Carmel i dr., 2014; Yih i dr., 2013; Bunescu i Huang, 2010) u pokuse su uključeni modeli pretraživanja temeljeni na nadziranom učenju rangiranja nad raznolikim skupom lingvistički motiviranih značajki. Prvi model je ListNET (Cao i dr., 2007), koji je te-

meljen na jednostavnim neuronskim mrežama. Drugi model je LambdaMART (Wu i dr., 2010), koji je varijanta modela LambdaRank (Burges i dr., 2007) temeljena na gradijentnom poticanju (engl. *gradient boosting*). Oba modela uče *po listi*, a više informacija o njima može se naći u odjeljku 2.3. Implementacije ovih modela pretraživanja su javno dostupne.[4]

Skup korištenih značajki uključuje mjere površinske sličnosti tekstova, kao i složenije lingvistički motivirane značajke. Značajke se stvaraju za svaki par korisničkog upita q_i i potencijalno relevantnog FAQ-para f_j, a navedene su u nastavku.

1. Rangovi koje ima f_j za upit q_i prema četiri nenadzirana temeljna modela pretraživanja (BM25, VS, SG, i RC);
2. značajke vreće riječi (engl. *bag-of-words* – BoW) – broj pojavljivanja za svaku riječ u rječniku. Prije izračuna ove značajke riječi se korjenuju. Postoje dva broja za svaku riječ: broj pojavljivanja u q_i i broj pojavljivanja u f_j. Zbog vrlo velikog broja ovih značajki (red veličine više tisuća), proveden je odabir značajki temeljen na filtru (engl. *filter feature selection*). Za to je iskorišten statistički test χ^2, na način da su izbačene sve značajke osim njih 100 s najvećom vrijednosti statistike χ^2. Razmatrano je i korištenje više značajki ovog tipa. No, daljnje povećanje njihovog broja nije dovelo do boljih rezultata, ali je ozbiljno produžilo vrijeme potrebno za učenje modela;
3. Levenshteinova udaljenost (Levenshtein, 1966) između q_i i f_j;
4. Četiri značajke sličnosti implementirane u paketu SEMILAR (Rus i dr., 2013): (i) pohlepno uparivanje riječi na razini rečenice[5] kombinirano s mjerama za sličnost riječi temeljenima na leksičkosemantičkoj bazi WordNet, (ii) optimalno uparivanje kombinirano sa sličnošću riječi temeljenoj na LSA, (iii) BLEU (Papineni i dr., 2002) sličnost između q_i i q_j i (iv) kosinusna sličnost između LSA vektora dokumenata (v. odjeljak 2.2.3.) za q_i i f_j;
5. Sličnost temeljena na jezgrama stabala – računa se sličnost na razini rečenice između q_i i f_j korištenjem jezgrenih funkcija nad sintaktičkim stablima implementiranih u biblioteci KeLP (Filice i dr., 2015). Za dobivanje ovisnosnih sintaktičkih stabala rečenica korišten je Stanfordov parser (Chen i Manning, 2014). Primijenjene jezgrene funkcije uključuju jezgrenu funckiju temeljenu na podstablima (engl. *subtree kernel* – STK) (Smola i Vishwanathan, 2003), jezgrenu funkciju temeljenu na podskupovima stabala (engl. *subset tree kernel* – SSTK) (Collins i Duffy, 2002) i jezgrenu funkciju temeljenu na parcijalnim stablima (engl. *partial tree kernel* – PTK) (Moschitti, 2006). Ova moćna skupina značajki opisuje sličnost između q_i i f_j tako što mjeri preklapanje u strukturi njihovih ovisnosnih sintaktičkih stabala. Nadalje, vrijedi naglasiti da je u ovaj skup uključena i zaglađena

[4] Koristi se biblioteka RankLib, dostupna za preuzimanje na `https://sourceforge.net/p/lemur/wiki/RankLib/`.

[5] Sve značajke sličnosti definirane na razini rečenice računaju ukupnu sličnost između dva teksta kao maksimum sličnosti po svim mogućim parovima rečenica iz tih tekstova.

jezgrena funkcija temeljena na djelomičnim stablima (engl. *smoothed partial tree kernel* – SPTK) (Croce i dr., 2011),koja uz razmatranje preklapanja sintaktičkih stabala, uzima u obzir i semantičku sličnost leksikaliziranih čvorova stabala.[6] Za mjeru semantičke sličnosti riječi ta jezgrena funkcija koristi semantičke prikaze riječi temeljene na modelu skip-gram (Mikolov i dr., 2013), opisanom u odjeljku 2.2.4.

U pokusima, proveden je stupnjeviti unaprijedni odabir značajki (engl. *sequential forward feature selection*) (Whitney, 1971) iterativnim dodavanjem skupina značajki 1–5 i promatranjem rezultata na izdvojenom skupu za provjeru. Najbolje je rezultate imala inačica modela u kojoj su u model bile uključene sve grupe značajki.

Model pretraživanja temeljen na konvolucijskoj neuronskoj mreži

S obzirom na uspješnu primjenu konvolucijske neuronske mreže (engl. *convolutional neural network* – CNN) u (Severyn i Moschitti, 2015) taj je pristup preuzet kao jedan od modela pretraživanja razmotrenih u ovom istraživanju. Kao što je već obrazloženo u odjeljku 2.3.3., modeli CNN pružaju dobre rezultate za velik broj zadataka analize i pretraživanja teksta, uključujući i pretraživanje FAQ-zbirki. Ovaj model pretraživanja koristi pristup učenju rangiranja *po točkama*. Specifično, zadatak rangiranja formuliran je kao klasifikacijski zadatak, $(q_i, f_j) \mapsto y_{ij}$, gdje je oznaka klase y_{ij} jednaka 1 ako i samo ako je FAQ-par f_j relevantan za korisnički upit f_j.

Ovaj model pretraživanja (u daljnjem tekstu: CNN-rank) pojednostavljena je inačica modela predstavljenog u (Severyn i Moschitti, 2015), te je detaljnije opisan u odjeljku 2.3.3. Model se uči klasičnim stohastičkim gradijentnim spustom (Rumelhart i dr., 1988), minimizirajući pogrešku unakrsne entropije po svim parovima korisničkih upita i FAQ-parova. Slično kao što je slučaj u (Kim, 2014), ne postoji dovoljno primjera za učenje kako bi se učili semantički vektorski prikazi riječi specifični za zadatak, pa su semantički prikazi riječi fiksirani tijekom učenja. Hiperparametri ovog modela uključuju broj filtara za upit i za FAQ-par, broj neurona u skrivenom sloju, postotak neurona izostavljanih tijekom učenja (engl. *dropout*) i konstantu učenja (ukupno pet hiperparametara).

Tri tehnička detalja zaslužuju posebno detaljan opis:

Regularizacija. Budući da su dostupni skupovi podataka prilično mali za primjenu dubokih neuronskih mreža, posebna pažnja posvećena je suzbijanju prenaučenosti kroz regularizaciju. Jedna jednostavna, no vrlo učinkovita, tehnika regularizacije jest izostavljanje neurona (Srivastava i dr., 2014). Izostavljanje neurona se temelji na zanemarivanju slučajno izabranih neurona tijekom svake iteracije učenja. To otežava međusobnu prilagodbu neurona, koja je tipična za prenaučene modele. Izostavljanje se koristi u svim slojevima mreže. Dodatno, koristi se i rano zaustavljanje (engl. *early stopping*) učenja kroz praće-

[6] Listovi ovisnosnog sintaktičkog stabla koji predstavljaju same riječi iz rečenice.

nje empirijske pogreške na izdvojenom skupu za provjeru. Onaj skup težina koji je imao najbolje rezultate na skupu za provjeru se koristi kao konačna inačica modela kojom se označava ispitni skup;

Uravnotežavanje klasa. Karakteristika dostupnih podataka jest da su primjeri za učenje modela CNN-rank vrlo neuravnoteženi s obzirom na klasu. Razlog tomu je što je za bilo koji upit većina FAQ-parova zapravo nerelevantna. Ova činjenica je problematična, jer je poznato da neuravnoteženost klasa kvari ponašanje algoritama učenja temeljenih na gradijentnom spustu (Oquab i dr., 2014). Kako bi se riješio problem, pozitivna klasa je naduzorkovana[7] kako bi skup za učenje postao uravnotežen. Naravno, ova prilagodba je provedena samo na podacima za učenje i provjeru, jer bi u stvarnoj primjeni skup za ispitivanje zaista bio neuravnotežen;

Rerangiranje. Kako bi se izbjegla primjena modela pretraživanja učenog na uravnoteženom skupu za učenje na neuravnotežen skup za ispitivanje, korišteno je rješenje predloženo u (Moschitti i Quarteroni, 2011) i (Jansen i dr., 2014). Prvo se koristi nenadzirani model pretraživanja (u ovom istraživanju BM25) kako bi se dohvatio skup FAQ-parova koji su dobri kandidati da budu relevantni. Potom se pomoću modela CNN-rank rerangira najgornjih N_{CNN} FAQ-parova s tog popisa. Kako bi se razumjela intuicija iza ove strategije, potrebno je razmotriti na koji način broj N_{CNN} utječe na uravnoteženost skupa podataka koji rerangira model CNN-rank. Za velike vrijednosti N_{CNN}, očekivano je da su skoro svi FAQ-parovi nerelevantni, dok je za vrlo male vrijednosti očekivano da bi skoro svi FAQ-parovi bili relevantni. Negdje između – vjerojatno bliže manjim vrijednostima – postoji točka gdje je skup FAQ-parova na koje se primjenjuje CNN-rank model dovoljno uravnotežen da odražava uravnoteženu situaciju kakva je vrijedila u skupu za učenje. Ovo omogućava modelu CNN-rank da radi na neuravnoteženom ispitnom skupu. Osim što rješava problem neuravnoteženosti, rerangiranje ima dodatnu prednost vremenske učinkovitosti. Dohvaćanje liste FAQ-parova koji su dobri kandidati vrlo je učinkovito, dok se računski složen CNN-rank model mora primijeniti tek na manji broj N_{CNN} najbolje rangiranih FAQ-parova. Zbog pravednosti vrednovanja, prag N_{CNN} je smatran hiperparametrom modela te je optimiran na izdvojenom skupu za provjeru.

9.4. Implementacija

Za temeljne postupke pretraživanja i vrednovanje korištene su iste implementacije u programskom jeziku C# koje su već spomenute u odjeljku 5.2. Za implementaciju ListNET i LambdaMART postupaka korištena je biblioteka RankLib.[8] Za izračun značajki kod modela temeljenih

[7] Pozitivni primjeri su ponovljeni prikladan broj puta, kako bi njihov broj postao sumjerljiv broju negativnih primjera.

[8] `https://sourceforge.net/p/lemur/wiki/RankLib/`

na značajkama sličnosti korištene su biblioteke SEMILAR (Rus i dr., 2013) za površinske značajke, odnosno KeLP (Filice i dr., 2015) za značajke temeljene na jezgrama stabala. Opisana arhitektura konvolucijske neuronske mreže u potpunosti je implementirana je u programskom jeziku Python koristeći biblioteku Theano (Theano Development Team, 2016). Pokusi s modelom CNN-rank provedeni su koristeći NVidia Titan X grafičku karticu.

9.5. Vrednovanje

U ovom odjeljku opisan je niz pokusa kojima su vrednovani različiti modeli pretraživanja i razmotrena istraživačka pitanja postavljena u odjeljku 9.1. U pokuse su uključeni temeljni postupci pretraživanja (BM25, VS, SG, i RC) te postupci temeljeni na nadziranom učenju rangiranja (ListNET, LambdaMART i CNN-rank). Prvo istraživačko pitanje razmatra se kroz usporedbu povećanja kvalitete rezultata koju nude modeli pretraživanja temeljeni na nadziranom učenju rangiranja u usporedbi s temeljnim modelima pretraživanja.

Nakon toga je razmotreno drugo istraživačko pitanje: ispitano je hoće li najbolji modeli temeljeni na nadziranom učenju davati podjednako dobre rezultate kada su učeni na podacima označenima pomoću strategije usmjerene na parafraze. Zbog iscrpnosti, vrednovanje je provedeno na skupovima FAQIR i StackFAQ opisanima u odjeljcima 3.4. i 3.5. Za nenadzirane modele provodi se jednostavna predobradba koja uključuje opojavničenje (engl. tokenization), korjenovanje i uklanjanje zaustavnih riječi.[9] Modeli pretraživanja ListNET i LambdaRANK traže dodatnu predobradbu, koja je opisana u odjeljku 9.3., dok se kao ulaz neuronske mreže koriste semantički vektori riječi bez predobradbe.

9.5.1. Postav vrednovanja

Kako bi se dobile nepristrane procjene pogreške za modele pretraživanja, modeli su vrednovani postupkom unakrsne provjere s pet preklopa (engl. *five fold cross-validation*). U svakoj iteraciji postupka skup upita podijeljen je u omjeru 80% naprama 20%. Pri tome je 20% upita predstavljalo ispitni skup, dok je 80% upita dalje podijeljeno u skup za učenje i skup za provjeru u omjeru 3:1 (od 80% upita u svakoj iteraciji, 60% je korišteno za učenje a 20% za provjeru). Model pretraživanja učen je na skupu za učenje, dok je skup za provjeru korišten za odabir optimalnih vrijednosti hiperparametara modela. Rezultati iz tablica u nastavku poglavlja su rezultati tako izgrađenog modela na skupu za ispitivanje (prosjek po pet preklopa).

U ovom se slučaju pri implementaciji unakrsne provjere postavlja pitanje kako rasporediti upite po skupovima za učenje, provjeru i ispitivanje. Neka Q_i označava skup svih upita koji pokrivaju neku informacijsku potrebu I_i. Prva mogućnost je da svi upiti iz Q_i budu svrstani

[9] Za ovaj korak korišten je Natural Language Toolkit (NLTK) (Loper i Bird, 2002).

Tablica 9.1: Razmatrane vrijednosti hiperparametara za modele pretraživanja temeljene na nadziranom učenju rangiranja.

Model	Hiperparametar	Isprobane vrijednosti	Optimalna vrijednost
ListNET	Konstanta učenja	0,0001, 0,01	0,01
LambdaMART	Broj stabala	50, 500, 1000	500
	Broj listova u svakom stablu	5, 10, 20	20
CNN-rank	Broj filtara	30, 50, 70	70
	Broj neurona skrivenog sloja	15, 30	15
	Postotak izostavljenih neurona	20%, 50%	20%
	Konstanta učenja	0,01, 0,1 , 1	0,1
	Broj dokumenata za reangiranje	1–30	23

u isti podskup. Ovakav postav bi omogućio ispitivanje sposobnosti modela pretraživanja da generalizira na nove, još neviđene informacijske potrebe. Druga mogućnost jest da upiti iz Q_i budu ravnomjerno raspoređeni po skupovima za učenje, provjeru i ispitivanje. Suprotno prvom slučaju, ovakav postav omogućava ispitivanje sposobnosti modela pretraživanja da generalizira na različite inačice već viđene informacijske potrebe. Iako na prvi pogled prva mogućnost izgleda prikladnija, važno je napomenuti da je ona nerealna za slučaj pretraživanja domenski specifičnih FAQ-zbirki. Naime, ključna karakteristika tog zadatka jest da je broj različitih informacijskih potreba vrlo ograničen te ih korisnici izražavaju na različite načine. Zato bi idealan model pretraživanja FAQ-zbirki trebao naučiti iznimno dobro generalizirati na različite inačice manjeg broja poznatih informacijskih potreba. Točnije, model treba potaknuti da žrtvuje sposobnost generalizacije na nove informacijske potrebe. Zauzvrat, model dobiva znatno povećanu sposobnost generalizacije na različite inačice predefiniranog skupa informacijskih potreba. Uz ovu intuiciju, vrednovanje je provedeno tako da su upiti iz Q_i raspoređeni po sva tri podskupa podataka tijekom unakrsne provjere.

Hiperparametri nadziranih modela pretraživanja prvo su optimirani korištenjem pretrage po rešetci (engl. *grid search*) (Bergstra i dr., 2011) uz promatranje rezultata na skupu za provjeru. Potom je model s najboljim rezultatom na skupu za provjeru korišten za označavanje ispitnog skupa. Razmatrani rasponi hiperparametara modela pretraživanja i njihove optimalne vrijednosti dane su u tablici 9.1. Vrijedi napomenuti da stvarne optimalne vrijednosti hiperparametara variraju po preklopima; vrijednosti koje su navedene u tablici su one koje su u najviše preklopa bile odabrane kao optimalne.

Uz korisnički upit q i FAQ-par $f = (Q, A)$ nije odmah očito je li bolje usporediti q sa FAQ

pitanjem Q ili s cijelim FAQ-parom (Q,A). Prva opcija bi mogla biti poželjna u slučajevima kada A sadrži mnogo nevažnih riječi koje samo zbunjuju model pretraživanja. S druge strane, druga je opcija bolja u slučajevima kada A sadrži riječi važne za ispravno određivanje relevantnosti. Zbog iscrpnosti, u pokuse su uključene obje inačice, a one su označene kao *postav Q* i *postav QA*. Razmatrana je i treća opcija u kojoj se q uspoređuje samo s A, ali se pokazalo da ona radi osjetno slabije od preostale dvije opcije.Vrijedi napomenuti da niti jedna od ovih opcija ne radi usporedbu s tijelom[10] pitanja jer, za razliku od uobičajene situacije u CQA, FAQ-pitanje nema tijelo.

Učinkovitost modela pretraživanja vrednovana je tradicionalnim mjerama vrednovanja za zadatak pretraživanja informacija. Korištene su mjere MAP, MRR i P@5, koje su opisane u odjeljku 2.4. Mjera P@5 posebno je prikladna za vrednovanje pretraživanja FAQ-zbirki jer je realno očekivati da će većina korisnika smatrati pretraživanje neuspješnim ako ne pronađu relevantan FAQ-par pregledom prvih pet rezultata. Uz ovakvu pretpostavku mjera MRR je preblaga, dok je mjera MAP prestroga. No, mjera P@5 vrlo dobro opisuje kriterije za vrednovanja koji najbolje odražavaju očekivanje korisnika.

Postav pokusa za skup StackFAQ sličan je gore opisanom postupku za skup FAQIR uz sljedeće promjene. Prvo, koristi se fiksna podjela upita na skup za učenje (60%), skup za provjeru (20%) i skup za ispitivanje (20%). Drugo, razmatraju se samo najbolji nadzirani i temeljni modeli pretraživanja iz pokusa na skupu FAQIR. Ovaj pojednostavljeni postav vrednovanja je primijenjen prije svega zbog visoke računske složenosti izvođenja unakrsne provjere za sve modele pretraživanja. U ovom pogledu, pokusi na skupu StackFAQ imaju svrhu potvrđivanja iscrpnijih pokusa provedenih na skupu FAQIR.

9.5.2. Nadzirano učenje rangiranja u usporedbi s nenadziranim modelima pretraživanja

Cilj prvog pokusa je dvojak. Prvo, potrebno je odrediti koliko su učinkoviti temeljni modeli pretraživanja. Drugo, potrebno je odrediti kolika je gornja granica učinkovitosti nadziranih modela, koju oni postižu uz skup za učenje koji je označen potpunom strategijom (v. odjeljak 9.1.). Rezultati za sve modele u *postavu Q* i *postavu QA* prikazani su u tablici 9.2 i tablici 9.3 za skupove FAQIR odnosno StackFAQ.

Prva stvar koja se može primijetiti jest da su relativne razlike u rezultatima između modela dosljedne bez obzira na postav (*postav Q* ili *postav QA*). Nadalje, za skup podataka FAQIR, rezultati u *postavu QA* dosljedno su bolji od rezultata u *postavu Q*. Ovo upućuje na to da, za ovaj skup podataka, FAQ odgovori nose više korisne informacije za zadatak pretraživanja FAQ-zbirki nego slučajnog šuma koji bi zbunjivao modele. S druge strane, na skupu podataka StackFAQ,

[10] U CQA pitanja se tipično sastoje od naslova i tijela.

Tablica 9.2: MAP, MRR i P@5 rezultati četiri temeljna modela pretraživanja i tri nadzirana modela pretraživanja na skupu podataka FAQIR označenom potpunom strategijom, odvojeno za *postavu Q* i *postavu QA*.

	postav Q			postav QA		
Model	MAP	MRR	P@5	MAP	MRR	P@5
BM25	0,46	0,77	0,50	0,51	0,78	0,54
SG	0,45	0,76	0,52	0,49	0,76	0,55
VS	0,44	0,75	0,49	0,50	0,77	0,54
RC	0,49	0,80	0,56	0,53	0,80	0,58
ListNET	0,49	0,80	0,54	0,53	0,80	0,57
LambdaMART	0,52	0,81	0,57	0,57	0,84	0,61
CNN-rank	**0,53**	**0,83**	**0,64**	**0,58**	**0,85**	**0,66**

Tablica 9.3: MAP, MRR i P@5 rezultati RC temeljnog modela i tri nadzirana modela pretraživanja na StackFAQ skupu podataka označenom potpunom strategijom, odvojeno za *postav Q* i *postav QA*.

	postav Q			postav QA		
Model	MAP	MRR	P@5	MAP	MRR	P@5
RC	0,74	0,73	0,60	0,63	0,80	0,52
ListNET	0,73	0,68	0,56	0,54	0,70	0,51
LambdaMART	0,75	0,76	0,62	0,74	0,84	0,60
CNN-rank	**0,79**	**0,77**	**0,63**	**0,74**	**0,84**	**0,62**

modeli učeni u *postavu Q* rade bolje od onih učenih u *postavu QA* po mjerama MAP i P@5, ali ne i po mjeri MRR. Razlog ovakvog ponašanja vjerojatno se može pronaći u načinu na koji je ovaj skup izgrađen (v. odjeljak 3.5.). Točnije, kvaliteta FAQ odgovora u ovom skupu bila je provjerena neizravno, kroz broj glasova koje je pojedini odgovor dobio. Velik broj glasova nije nužno uvijek indikacija visoke relevantnosti. Posljedica ovoga jest da kvaliteta FAQ odgovora u ovom skupu varira mnogo više nego što je to slučaj za skup FAQIR. Ova pojava uzrokuje da su ponekad kao relevantni označeni oni FAQ-parovi čiji FAQ odgovor zapravo uopće nije relevantan za korisnički upit. Takve parove modeli pretraživanja vjerojatno će rangirati nisko, što će utjecati na iznose mjera vrednovanja. Najmanje će biti narušena mjera MRR, jer ona ovisi samo o jednom, najbolje rangiranom relevantnom dokumentu, koji vjerojatno neće biti pogođen opisanim problemom. No, mjere P@5 i MAP razmatraju cijeli popis dohvaćenih dokumenata, pa će opisani problem uzrokovati njihovo smanjenje u *postavu QA* u odnosu na *postav Q*.

Drugo opažanje jest da nenadzirani modeli imaju vrlo dobre rezultate. Model pretraživanja RC postiže rezultat od 0,58 i 0,68 bodova po mjeri P@5 na skupovima podataka FAQIR i StackFAQ. Ovaj rezultat je iznad svih drugih nenadziranih modela pretraživanja. Ovo je očekivan rezultat, budući da je model pretraživanja RC zapravo ansambl svih preostalih nenadziranih modela pretraživanja.

Treća stvar koja se može analizirati su međusobni odnosi po performansama između nadziranih modela pretraživanja. Na skupu podataka FAQIR postupak ListNET daje bolje rezultate od jednostavnijih temeljnih modela kao što su BM25 ili VS, ali radi otprilike podjednako kao njihova kombinacija RC. No, na skupu podataka StackFAQ model ListNET ima slabije rezultate od RC. Drugi najbolji model pretraživanja jest LambdaMART, koji na oba skupa podataka daje vrlo dobre rezultate po mjerama MAP i MRR. Najbolji nadzirani model je CNN-rank, koji ima rezultate od 0,66 i 0,63 bodova P@5 mjere na skupovima FAQIR odnosno StackAFAQ.

Konačno, važno je napomenuti da najbolji nadzirani model (CNN-rank) rezultira sa značajnim poboljšanjem u odnosu na nenadzirane temeljne modele pretraživanja. Ovaj pomak u rezultatima može se kvantificirati pomoću apsolutnih razlika u iznosu P@5 mjere između najboljeg nenadziranog modela pretraživanja (RC) i modela CNN-rank. Za skup podataka FAQIR, pomak je 8 (14,2%) odnosno 8 (13,7%) bodova P@5 mjere u *postavu Q* i *postavu QA*. Za skup podataka StackFAQ pomak iznosi 3 (3,9%) odnosno 10(19,2%) bodova P@5 mjere u *postavu Q* i *postavu QA*. Ove su razlike statistički značajne uz $p < 0,05$.[11]

U smislu jakosti učinka, opisane razlike mogu se smatrati i praktično značajnima. Točnije, povećanje P@5 mjere se može izravno interpretirati kao povećanje postotka korisničkih upita koji su odgovoreni u prvih pet rezultata. U kontekstu velikih pružatelja usluga, povećanje P@5 mjere od 5 do 10 bodova može imati osjetan utjecaj na poslovanje. Zbog toga se može zaklju-

[11] Za statističko testiranje korišten je dvostrani statistički test temeljen na ponovnom uzorkovanju (engl. *bootstrap resampling test*), gdje su uzorci izvučeni s ponavljanjem iz skupa upita.

čiti da je odgovor na prvo istraživačko pitanje izneseno u odjeljku 9.1. pozitivan: nadzirano učenje rangiranja za pretraživanje domenski specifičnih FAQ-zbirki uz podatke označene potpunom strategijom *može* dovesti do praktično relevantnih poboljšanja rezultata u usporedbi s nenadziranim modelima pretraživanja.

9.5.3. Analiza pogrešaka

Modeli pretraživanja koji su vrednovani u provedenom pokusu razlikuju se u načinu na koji pristupaju rješavanju problema u obradi prirodnog jezika. Zbog toga je zanimljivo analizirati i usporediti tipične uzroke pogrešaka koje rade ovi modeli. U tu svrhu provedena je analiza pogrešaka najboljeg temeljnog modela pretraživanja (RC) i najboljeg nadziranog modela pretraživanja (CNN-rank). Ovo je provedeno na jednom od pet preklopa za skup FAQIR, što se svodi na 228 upita. Analiza pokazuje da većina pogrešaka može biti pripisana jednom od sljedeća tri glavna uzroka:

1. Nediskriminativne riječi – Upit sadrži riječ koja se pojavljuje u velikom broju nerelevantnih FAQ-parova. Na primjer, za upit *"Repairing a broken vacuum cleaner"* (*"Popravljanje pokvarenog usisavača"*), riječ *"broken"* (*"pokvaren"*) pojavljuje se u FAQ-parovima koji su povezani sa konceptima kao što su *"broken switch"* (*"pokvarena sklopka"*) ili *"broken window"* (*"Razbijen (pokvaren) prozor."*);
2. Leksički jaz – Sinonimi ili povezane riječi koje se pojavljuju u korisničkom upitu i FAQ-paru nisu ispravno prepoznate kao semantički povezane. Na primjer, *"Repair an automobile"* (*"Popravljanje automobila."*) i *"Fix a car"* (*"Servis auta"*);
3. Znanje o svijetu – Primjerice, za upit *"How to get a sticker off glass surface?"* (*"Kako skinuti naljepnicu sa staklene površine."*) i FAQ-par koji sadrži pitanje *"Kako ukloniti etikete sa staklenki?"*, sustav mora znati da je staklenka tipično od stakla i da je etiketa u ovom kontekstu vrsta naljepnice.

Kako je kroz pokus utvrđeno da model CNN-rank poboljšava rezultate u usporedbi s nenadziranim modelima, zanimljivo je istražiti do koje je mjere to poboljšanje posljedica uspješnog rješavanja gornjih problema. Od 228 upita, najbolji nenadzirani model (RC) bio je bolji od modela CNN-rank prema mjeri P@5 za 20 upita (9%), rezultat je bio izjednačen za 114 upita (50%), dok je CNN-rank bio bolji od modela RC za preostalih 94 upita (41%). Daljnja analiza otkriva da za slučajeve gdje je model CNN-rank bolji to zaista jest posljedica boljeg rješavanja nekog od gornjih problema. Za ove sposobnosti modela CNN-rank u načelu su odgovorna dva mehanizma. Prvo, model ima sposobnost da iz podataka za učenje nauči koje riječi iz upita i relevantnog FAQ-para su važne za korisnikovu informacijsku potrebu. Na primjer, neka je informacijska potreba povezana s uklanjanjem mrlji na tepihu te neka je korisnički upit (*"Kako mogu ukloniti mrlje sa tepisona gdje je moje dijete prolilo sok."*). Model CNN-rank može naučiti da su riječi (*"mrlja"*), (*"ukloniti"*) i (*"tepih"*) važne za informacijsku potrebu, dok riječi

(*"dijete"*) i (*"sok"*) to nisu. Ovo efektivno rješava prvi problem. Drugi mehanizam jest korištenje semantičkih vektorskih prikaza riječi koji dobro modeliraju sinonime i povezane riječi, čime se ublažava drugi naveden problem. Treba napomenuti da neki od nenadziranih modela (SG) također koriste semantičke vektorske prikaze riječi. Ključna razlika između CNN-rank i tih modela jest da CNN-rank koristi konvolucijsku neuronsku mrežu kao mnogo napredniji način za kombiniranje vektora riječi nego što je to operacija zbrajanja kakvu koristi SG i tijekom koje se gubi velika količina informacije sadržana u vektorima.

Tablica 9.4 prikazuje primjere koji predstavljaju tipične slučajeve kada je model CNN-rank imao bolje rezultate od temeljnih modela. U prvom primjeru, koncepti *"squeaking"* (*"škripanje"*) i *"floor"* (*"pod"*) važni su za informacijsku potrebu. Model CNN-rank uspješno je naučio tu informaciju iz podataka za učenje. S druge strane, temeljni model RC preferira FAQ-parove s većim brojem pojavljivanja riječi *"floor"* nad onima koji sadrže oba pojma, ali su rijeđe spomenuta. U drugom primjeru, koncepti važni za informacijsku potrebu korisnika su *"bird"* (*"ptica"*) i *"chase away"* (*"otjerati"*). Temeljni model RC se ovdje suočava s dva problema: (1) najviše rangiran (lažno pozitivni) FAQ-par bavi se konceptom *"attic"* (*"tavan"*), koji nije ključan za informacijsku potrebu, i (2) FAQ-parovi na pozicijama četiri i pet su lažno pozitivni zbog riječi *"fly"* (*"muha"*), koja se pojavljuje u smislu različitom od onoga koji ima u korisničkom upitu. Model CNN-rank uspješno je zaobišao ove probleme tako što je naučio oba pojma važna za informacijsku potrebu.

U 9% slučajeva u kojima CNN-rank model ima slabije performanse od modela RC, svi su posljedica jedne od sljedeće dvije krajnosti. Prvo, jedan od pojmova važnih za informacijsku potrebu nije bio prepoznat kao važan. Drugo, jedan od važnih pojmova ima pretjerano velik utjecaj na rezultat. Na primjer, za informacijsku potrebu koja je usredotočena na koncepte *"odor"* (*"miris"*), *"remove"* (*"ukloniti"*) i *"car"* (*"auto"*), velik broj FAQ-parova koje dohvaća model CNN-rank povezani su isključivo s autima i dijelovima auta. Vjerojatan uzrok ovoga jest što je koncept (*"miris"*) nedovoljno prisutan u podacima za učenje, pa je stoga model CNN-rank podcijenio njegovu važnost za informacijsku potrebu.

9.5.4. Pokusi sa strategijama usmjerenim na parafraze

Prvi pokus je pokazao da se praktično značajna povećanja točnosti pretraživanja mogu dobiti korištenjem nadziranog modela pretraživanja. Cilj sljedećeg pokusa jest procijeniti mogu li se ista povećanja točnosti pretraživanja dobiti uz manje truda potrebnog za označavanje. Kao što je argumentirano u odjeljku 9.1., za ovo bi se mogla koristiti strategija označavanja usmjerena na relevantnost ili strategija označavanja usmjerena na parafraze. Pritom je druga strategija mnogo učinkovitija s obzirom na trud koji je potrebno uložiti u označavanje. Točnije, relativno je jednostavno izgraditi parafraze upita parafraziranjem FAQ pitanja prisutnih u FAQ-zbirci. Zbog toga je drugo istraživačko pitanje, izneseno u odjeljku 9.1., usredotočeno na istraživanje može

Tablica 9.4: Primjeri najbolje rangiranih FAQ-parova za modele RC i CNN-rank na skupu podataka FAQIR. Za FAQ-parove su, zbog sažetosti, navedena samo FAQ pitanja.

Upit: How to silence my squeaky floor?		
Model	Relevantan?	FAQ pitanje
RC	Da	1. How do I fix squeaky stairs?
	Ne	2. How do I fix a squeaky Herman Miller Aeron chair? It's making me crazy!? It squeaks when it rocks forward and back (the wheels are fine).
	Ne	3. How can I remove wax from a wooden floor? I dropped a pot of wax on my hardwood floor. How can I remove it without damaging the floor?
	Ne	4. How do you get glue off of a wooden floor from an old carpet without ruining the floor??
	Ne	5. How can you take off stains off wooden floors? i had carpet over my wooden floors and after a while of spilling fluids over the carpet my floor penetrated it and has marks how can i get them to go away?
CNN-rank	Da	1. How do you repair sweaky wooden floors? i have hard wood floors that sweak throughout the house, is there something i can do to stop it without removing the entire floor.
	Da	2. How do you stop floors from squeeking?
	Da	3. How do I fix squeaky stairs?
	Da	4. How do you fix a squeeky hardwood floor?
	Da	5. How do i repair the creaking of our hardwood flooring?
Upit: I have a bunch of pigeons stuck in my attic and don't know how to get them out – they refuse to fly out the window.		
Model	Relevantan?	FAQ pitanje
RC	No	1. How to cool attic -in cape code home-we have two rooms and no central airconditioning?
	Da	2. how do i keep pigeons away from my balcony?
	Da	3. How to keep birds from eating your gardens?
	Ne	4. How do you get rid of flies in the yard. hundreds of them?
	Ne	5. How do you get rid of flies in the kitchen and bathroom?
CNN-rank	Da	1. How do I get birds to stay off of my deck without using nail strips or tacky glue stuff?
	Da	2. How do you get rid of a bird that's made a nest in the stack on hot water heater/furnace? Who can you call?
	Da	3. How to keep birds from eating your gardens?
	Da	4. How to drive away the pigeons from my factory shed? they create lot of problem thro their droppings.
	Da	5. How to keep pigeons away from my balcony?

li označavanje strategijom usmjerenom na parafraze rezultirati označenim skupom podataka jednako kvalitetnim kao skup označen potpunom strategijom.

Razmotrene su dvije strategije usmjerene na parafraze. One su označene kao strategija *ručno-prvihK* i strategija *pseudorelevantna*. Obje strategije smanjuju broj ocjena relevantnosti koje označivači moraju označiti. S druge strane, obje se strategije oslanjanju na velik broj označenih parafraza upita, koji bi trebale kompenzirati smanjenje redundancije dokumenata. Ako sa q označimo korisnički upit, strategije rade kako slijedi.

Ručno-prvihK strategija. Ova strategija smanjuje količinu FAQ-parova koje je potrebno ručno označiti tako što se ljudskim označivačima prikazuje samo K najbolje rangiranih FAQ-parova u rezultatu nenadziranog modela pretraživanja. Za sve ostale FAQ-parove automatski se pretpostavlja da nisu relevantni za q. Drugim riječima, označivači zanemaruju sve osim spomenutih K FAQ-parova, što drastično smanjuje ukupnu količinu posla. Intuicija iza ovog pristupa jest da prvih K rezultata nenadziranog modela pretraživanja vrlo vjerojatno sadrži skoro sve dokumente koji su zaista relevantni, pa su ti dokumenti najbolji kandidati za ručno označavanje relevantnosti.[12] Kako bi se u pokusu oponašala ta strategija, koristi se najbolji nenadzirani model pretraživanja (RC), kojim se dohvaća rangiran popis FAQ-parova za upit q. Potom se za prvih K FAQ-parova ostavlja njihova stvarna oznaka relevantnosti za q, dok se preostalima zapisuje oznaka *nerelevantan* s obzirom na q. Idealno bi bilo da ovako označen skup podataka omogućava učenje kvalitetnih modela pretraživanja već za niske vrijednosti K jer biranje prevelike vrijednosti za K je kontradiktorno s idejom smanjenja truda označavanja. U ovom pokusu, K je postavljen na razumne vrijednosti između 10 i 50 FAQ-parova (1–7% ukupnog broja FAQ-parova u skupu podataka). U svim inačicama pokusa postojao je značajan broj relevantnih FAQ-parova koji nisu završili u prvih K rezultata nenadziranog modela. Iz ovoga se može zaključiti da je skup oznaka dobiven ovom strategijom zaista različit od onog koji je se dobiva potpunom strategijom;

Pseudorelevantna strategija. Ova se strategija temelji na načelu pseudorelevantne povratne informacije (engl. *pseudo-relevance feedback*), koja je često korištena tehnika za proširenje upita (Manning i dr., 2008). Slično kao u *ručno-prvihK* strategiji, u *pseudorelevantnoj* strategiji prvi je korak pretraživanje informacija pomoću nenadziranog modela pretraživanja. Nakon toga, oznake relevantnosti potpuno se automatski generiraju. Prvih K dohvaćenih FAQ-parova označavaju se kao relevantni za q, dok se svi ostali FAQ-parovi označavaju kao nerelevantni. Opravdanje za ovo jest pretpostavka da bi prvih K dohvaćenih FAQ-parova zaista trebalo biti relevantno. Ključna razlika ovog modela i potpuno nenadziranih modela pretraživanja jest što ovaj model ima kapacitet da nauči implicitne

[12] Strategija je srodna strategiji agregiranja rezultata više modela pretraživanja (engl. *pooling*), koja se često koristi za smanjenje posla kod označavanja relevantnosti u području pretraživanja informacija (Manning i dr., 2008).

Tablica 9.5: MAP, MRR i P@5 rezultati CNN-rank i ListMART modela pretraživanja na skupu podataka FAQIR označenom strategijom usmjerenom na parafraze, odvojeno za *postav Q* i *postav QA*. Podebljano u svakom stupcu su najbolji rezultati modela ListMART i CNN-rank uz podatke označene na reduciran način.

			postav Q			postav QA		
Model	Strategija	K	MAP	MRR	P@5	MAP	MRR	P@5
RC	–	–	0,49	0,80	0,56	0,53	0,80	0,58
ListMART	*potpuna*	–	0,52	0,81	0,57	0,57	0,84	0,61
	ručno-prvihK	10	0,41	**0,81**	0,55	0,49	0,82	0,59
	ručno-prvihK	30	0,49	**0,81**	**0,57**	0,54	0,82	**0,60**
	ručno-prvihK	50	**0,50**	**0,81**	**0,57**	**0,55**	**0,83**	**0,60**
	pseudorelevantna	3	0,35	0,77	0,43	0,30	0,79	0,45
	pseudorelevantna	5	0,33	0,65	0,50	0,38	0,74	0,55
	pseudorelevantna	10	0,23	0,59	0,40	0,26	0,58	0,46
CNN-rank	*potpuna*	–	0,53	0,83	0,64	0,58	0,85	0,66
	ručno-prvihK	10	0,42	0,74	0,48	0,33	0,64	0,34
	ručno-prvihK	30	0,52	0,83	**0,63**	**0,58**	**0,84**	**0,65**
	ručno-prvihK	50	**0,53**	**0,85**	**0,63**	0,56	0,83	0,63
	pseudorelevantna	3	0,31	0,63	0,30	0,30	0,61	0,30
	pseudorelevantna	5	0,28	0,55	0,26	0,32	0,61	0,31
	pseudorelevantna	10	0,33	0,63	0,39	0,32	0,62	0,32

poveznice između upita i dokumenata, što može dovesti do bolje sposobnosti generalizacije za različite inačice iste informacijske potrebe. Važno je naglasiti da učinkovitost ovog modela u velikoj mjeri ovisi o pretpostavci da je prvih K rezultata nenadziranog modela visoke kvalitete, što u praksi ne mora nužno biti slučaj.

Rezultati obje strategije u *postavu Q* i u *postavu QA* za dva najbolja nadzirana modela (LambdaMART i CNN-rank) prikazani su u tablicama 9.5 i 9.6 za skup FAQIR odnosno skup StackFAQ.

Može se primijetiti da su rezultati oba modela, kada su oni učeni na podacima označenima *pseudorelevantnom* strategijom, vrlo nedosljedni s obzirom na različite vrijednosti K. Također, ta strategija vodi do znatno slabijih rezultata nego kada su podaci označeni potpunom strategijom. Ovaj rezultat dosljedan je za *postav Q* i *postav QA* za sve modele pretraživanja i sve

Tablica 9.6: MAP, MRR i P@5 rezultati CNN-rank i ListMART modela pretraživanja na skupu podataka StackFAQ označenom strategijom usmjerenom na parafraze, odvojeno za *postav Q* i *postav QA*. Podebljano u svakom stupcu su najbolji rezultati modela ListMART i CNN-rank uz podatke označene na reduciran način.

			postav Q			postav QA		
Model	Strategija	K	MAP	MRR	P@5	MAP	MRR	P@5
RC	–	–	0,74	0,73	0,60	0,63	0,80	0,52
ListMART	*potpuna*	–	0,75	0,76	0,62	0,74	0,84	0,60
	ručno-prvihK	10	0,74	0,75	**0,61**	0,72	0,83	**0,61**
	ručno-prvihK	30	0,76	**0,77**	**0,63**	**0,74**	**0,84**	**0,61**
	ručno-prvihK	50	**0,77**	0,74	0,61	**0,74**	0,83	0,61
CNN-rank	*potpuna*	–	0,79	0,77	0,63	0,74	0,84	0,62
	ručno-prvihK	10	0,79	**0,76**	0,64	0,72	**0,84**	0,60
	ručno-prvihK	30	**0,80**	**0,76**	**0,66**	**0,73**	**0,84**	**0,61**
	ručno-prvihK	50	**0,80**	**0,76**	0,64	0,70	0,83	0,59

skupove podataka. Zbog toga se može zaključiti da *pseudorelevantna* strategija nije učinkovita.

Za strategiju *ručno-prvihK* situacija je mnogo bolja. Nadzirani modeli učeni na podacima označenima ovom strategijom pokazuju vrlo dobre rezultate za oba skupa podataka i u oba postava za sve modele. Očekivano, kako raste broj dokumenata K koji se upućuju na ručno označavanje, tako raste i kvaliteta rezultata. No, već i uz relativno male vrijednosti K modeli pretraživanja i dalje rade dobro. Na $K = 30$, oba modela imaju rezultate usporedive s istim modelima u slučaju kada su oni učeni na podacima označenima potpunom strategijom. Za taj K oba modela pretraživanja imaju statistički značajan ($p < 0,05$, dvostrani test temeljen na ponovnom uzorkovanju upita) pomak u odnosu na nenadzirani RC model. Radi usporedbe, korisno je napomenuti da su označivači prilikom izgradnje skupa podataka FAQIR označili oznake relevantnosti za prosječno 202 dokumenta po informacijskoj potrebi (Karan i Šnajder, 2016). Strategija *ručno-prvihK* omogućava učenje nadziranih modela na podacima koji imaju označeno samo 30 FAQ-parova po informacijskoj potrebi. Ovo je gotovo sedmerostruko smanjenje truda označavanja relevantnosti, dok (znatno manji) trud označavanja parafraza ostaje isti. Smanjenje u redundantnosti dokumenata (o ovom slučaju FAQ-parova) nema ozbiljan negativan utjecaj na rezultate naučenog modela. Preostala, manja količina redundancije dokumenta, u spoju s redundancijom upita koju osiguravaju označene parafraze upita, omogućava učenje kvalitetnog nadziranog modela pretraživanja. Pokusi na skupu podataka StackFAQ potvrđuju ove trendove te pokazuju da je označavanje samo $K = 10$ FAQ-parova dovoljno da se dobije značajno poboljšanje rezultata pretraživanja, uz uvjet da postoje parafraze upita.

Rezultati provedenog pokusa navode na zaključak da je odgovor na drugo postavljeno istraživačko pitanje također potvrdan: strategija usmjerena na parafraze može se upotrijebiti za izgradnju skupa podataka koji će nadziranim učenjem dovesti do modela koji su podjednako kvalitetni kao i oni učeni na podacima označenima mnogo zahtjevnijom potpunom strategijom.

9.6. Rasprava

Ovo poglavlje bavilo se zadatkom pretraživanja FAQ-zbirki. To je posebna vrsta pretraživanja informacija koja koristi znanja iz područja odgovaranja na pitanja. Cilj je bio istražiti potencijal primjene nadziranih modela pretraživanja, temeljenih na učenju rangiranja. Nedostatak takvih modela, u odnosu na nenadzirane modele pretraživanja, jest što iziskuju prisutnost označenih podataka za učenje. Oznake u podacima zapravo prenose informaciju na dva načina: redundancija upita (dostupna kroz parafraze upita) i redundancija dokumenata (dostupna kroz oznake relevantnosti). Drugi cilj ovog istraživanja bio je pronalazak strategije označavanja podataka koja bi smanjila potreban trud označivača, a da se pritom ne smanji značajno kvaliteta modela koji bi bio naučen na tako označenim podacima.

U skladu s ovim ciljevima, provedena su dva pokusa. Prvi je ispitivao potencijal nadzira-

nih modela za pretraživanje FAQ-zbirki. Pokus je proveden na dva reprezentativna domenski specifična skupa podataka, na kojima su ispitana tri suvremena nadzirana modela pretraživanja temeljena na učenju rangiranja. Pokazano je da dva modela – algoritam LambdaMART kombiniran s raznovrsnim lingvističkim značajkama i konvolucijska neuronska mreža – mogu u usporedbi s nenadziranim temeljnim modelima pretraživanja postići poboljšanja točnosti koja su od praktičkog značaja. Nakon toga istraženo je mogu li se jednaka poboljšanja ostvariti u slučaju kada su nadzirani modeli učeni na podacima označenima strategijom usmjerenom na parafraze. Ovdje glavnu ulogu igra pretpostavka da parafraze upita mogu nadomjestiti smanjenje broja oznaka relevantnosti. Rezultati drugog pokusa ukazuju na to da jednostavna strategija, u kojoj se za svaku informacijsku potrebu označava manji broj parafraza (u provedenim pokusima 10–22), dok se oznake relevantnosti označavaju samo za FAQ-parove najviše rangirane (u provedenim pokusima 30) nenadziranim modelom, ima jednako dobar rezultat kao i model pretraživanja učen na podacima označenima potpunom strategijom. Također, ovi rezultati upućuju na to da bi prelazak s nenadziranih na nadzirane modele pretraživanja mogao biti isplativa investicija uz prikladnu strategiju označavanja.

Postoji niz smjerova u kojima bi se ovo istraživanje moglo nastaviti. Iako se pokazalo da modeli pretraživanja LamdaMART i CNN-rank daju dobre rezultate, još uvijek nije razmotren prostor svih mogućih modela pretraživanja, pa bi se isplatilo ispitati učinkovitost alternativnih, potencijalno moćnijih modela, kao što su neuronske mreže temeljene na nizovima (engl. *sequence-based neural models*). Drugi obećavajući smjer jest razmatranje postupaka za generiranje posebno teških negativnih primjera, koji bi mogli pomoći nadziranom modelu pretraživanja da nauči bolje razlikovati relevantne od nerelevantnih dokumenata. Slično, mogao bi se osmisliti pristup za generiranje dodatnih parafraza upita, sličan onome opisanom u (Figueroa, 2017). Nadalje, bilo bi zanimljivo prilagoditi arhitekturu neuronske mreže specifičnostima zadatka pretraživanja FAQ-zbirki. Na primjer, neuronskoj mreži bi se moglo predstaviti pitanje i odgovor iz FAQ-para kroz dva odvojena konvolucijska sloja, što bi omogućilo mreži da za njih nauči odvojene vektorske prikaze za usporedbu s upitom. Konačno, dodatno smanjenje obima označavanja moglo bi se postići kombiniranjem strategije usmjerene na parafraze s nekim od postupaka aktivnog učenja.

Poglavlje 10.

Zaključak

U ovom doktorskom istraživanju razmatrano je više zadataka analize i pretraživanja teksta definiranih nad zbirkama često postavljanih pitanja i odgovora (engl. *frequently asked questions collections* – FAQ-zbirke). Zadaci kojima se ovaj rad bavi odabrani su kao najznačajniji nad FAQ-zbirkama. Takve zbirke često se koriste u praksi, pa bi bolje rješavanje ovih zadataka značajno poboljšalo korisničko iskustvo velikog broja korisnika. U svrhu istraživanja stvorena su tri FAQ skupa podataka na engleskom jeziku. Dodatno, korišten je još jedan gotov skup podataka na hrvatskom jeziku. Ovi skupovi su javno dostupni za buduća istraživanja. Kroz predistraživanja pokazano koji su skupovi prikladne veličine i kvalitete za primjenu modela temeljenih na nadziranome strojnom učenju. Predistraživanja, opisana u poglavljima 4., 5. i 6., uključivala su tekstove na hrvatskom i engleskom jeziku, ali nastavak istraživanja proveden je na skupovima podataka na engleskome jeziku. Ipak, važno je naglasiti da su najbolji predloženi modeli skoro u potpunosti jezično neovisni, pa njihova prilagodba za hrvatski jezik ne bi bila previše složen zadatak. U ovom radu predstavljena su sljedeća tri izvorna znanstvena doprinosa.

Prvi ostvareni doprinos, opisan u poglavlju 7., jest postupak za strojno potpomognutu izgradnju zbirki često postavljanih pitanja i odgovora. Predloženi postupak sastoji se od dvije komponente. Prva komponenta temeljena je na pronalasku čestih upita korisnika koristeći grupiranje s ograničenjima kombinirano s aktivnim učenjem. Druga komponenta temelji se na pretraživanju tekstnih informacija kako bi se u dostupnoj dokumentaciji pronašlo odgovore na česta pitanja. Predloženi postupak pomaže domenskom stručnjaku da poveća brzinu izgradnje i konačnu kvalitetu FAQ-zbirke.

Drugi ostvareni doprinos, opisan u poglavlju 8., bavi se postupkom za otkrivanje korisničkih upita koji nisu pokriveni FAQ-zbirkom. Ovo olakšava održavanje FAQ-zbirke na način da omogućava strojno potpomognutu nadgradnju zbirke novim, važnim FAQ-parovima.

Treći doprinos ostvaren u okviru ovog rada, opisan u poglavlju 9., jest model za semantičko pretraživanje FAQ-zbirki. Predloženo je više modela temeljenih na postupcima nadziranog strojnog učenja rangiranja. Pri tome su razmotrena dva načina prikaza teksta za modele

pretraživanja. Prvi prikaz uključivao je više statističkih značajki za semantičku sličnost tekstova dobivenih postupcima obrade prirodnog jezika. Drugi prikaz sastojao se od semantičkih vektorskih prikaza riječi koji su izravno bili ulaz u model pretraživanja temeljen na konvolucijskoj neuronskoj mreži.

Svi predloženi pristupi iscrpno su vrednovani te su se pokazali kao poboljšanje u usporedbi sa odgovarajućim referentnim pristupima. Pored toga, učinkovitost predloženih pristupa dovoljno je visoka za primjenu na FAQ-zbirkama kakve se tipično javljaju u praksi. Stoga možemo zaključiti da su kao rezultat ovog rada nastali praktično primjenjivi modeli i postupci.

Dodatan važan zaključak proizlazi iz činjenice da se na više mjesta u radu koristi skup označenih parafraza upita. Ovo je resurs specifičan za male, domenski specifične FAQ-zbirke jer je na takvim zbirkama ostvarivo njegovo označavanje. Gdje god je bilo moguće (drugi i treći doprinos), ova je prednost bila ugrađena u postupke i modele kako bi poboljšala njihovu učinkovitost. Vrednovanje je potvrdilo da je to korisno – pristupi koji koriste parafraze upita pokazali su značajno bolje rezultate od pristupa koji ih ne koriste.

Postoje mnoge mogućnosti nastavka istraživanja. Neke od njih ocrtane su u raspravnim dijelovima dotičnih poglavlja. Na razini sveukupnog istraživanja mora se naglasiti da, iako je ono iscrpno, još uvijek nisu isprobani svi dostupni postupci iz područja strojnog učenja. U budućnosti bi se isplatilo isprobati alternativne postupke koji bi možda dali bolje rezultate. Nadalje, kako su modeli i postupci zamišljeni tako da ne budu suviše ovisni o jeziku, jedan neposredan smjer budućeg istraživanja jest primjena i vrednovanje predloženih modela i postupaka na hrvatskim tekstovima, ali također i na tekstovima pisanima na raznim drugim jezicima. Drugi smjer istraživanja usmjeren na jezik primjene bio bi da se odmakne od jezične neovisnosti te da se iskoriste svi dostupni resursi za velike jezike. Na primjer, za engleski bi se jezik u modele moglo uključiti ljudsko znanje modelirano u velikim ontologijama kao što su Cyc[1] ili DBpedia.[2] Nadalje, još jedna zanimljiva mogućnost jest kombiniranje različitih zadataka, odnosno izgradnja modela koji združeno obavljaju zadatke koje su rješavali doprinosi rada. Motivacija za ovo jest činjenica da kod tih zadataka postoji stanovito preklapanje. Na primjer, ako je model pretraživanja vrlo loš za neku informacijsku potrebu, to bi moglo upućivati na to da je kod izgradnje FAQ-zbirke ta potreba loše definirana. Drugi je primjer skup upita koji su nakon izgradnje FAQ-zbirke pronađeni kao nepokriveni, a koji bi mogao biti ulaz za drugu iteraciju postupka izgradnje, koja bi nadopunila zbirku. Postoje brojne ovakve međuovisnosti koje bi se mogle iskoristiti za poboljšanje sveukupnog sustava. Konačno, potrebno je detaljnije istražiti načine za osiguranje skalabilnosti svih dijelova ovog istraživanja kroz odgovarajuću predobradu jednostavnijim ali bržim postupcima. Ovo je vrlo važno jer osigurava da predloženi pristupi upravljanju FAQ-zbirkama budu primjenjivi na različitim domenama koje se javljaju u praksi.

[1] `http://www.opencyc.org/`
[2] `http://wiki.dbpedia.org/`

Popis slika

Popis tablica

Literatura

Abraham, I., Alonso, O., Kandylas, V., Patel, R., Shelford, S. i Slivkins, A. 2016. How Many Workers to Ask?: Adaptive Exploration for Collecting High Quality Labels. *Str. 473–482 u: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. New York, NY, USA: ACM.

Agarwal, A., Raghavan, H., Subbian, K., Melville, P., Lawrence, R. D., Gondek, D. C. i Fan, J. 2012. Learning to Rank for Robust Question Answering. *Str. 833–842 u: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012)*. ACM.

Agić, Ž. i Merkler, D. 2013. Three Syntactic Formalisms for Data-driven Dependency Parsing of Croatian. *Str. 560–567 u: International Conference on Text, Speech and Dialogue*. Springer.

Agirre, E., Diab, M., Cer, D. i Gonzalez-Agirre, A. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. *Str. 385–393 u: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A. i Guo, W. 2013. *SEM 2013 Shared Task: Semantic Textual Similarity, Including a Pilot on Typed-similarity. *U: *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*.

Ahasanuzzaman, M., Asaduzzaman, M., Roy, C. K. i Schneider, K. A. 2016. Mining Duplicate Questions in Stack Overflow. *Str. 402–412 u: Proceedings of the 13th International Conference on Mining Software Repositories*. MSR '16. New York, NY, USA: ACM.

Alonso, O., Rose, D. E. i Stewart, B. 2008. Crowdsourcing for relevance evaluation. *Str. 9–15 u: ACM SigIR Forum*, vol. 42. ACM.

Alonso-Weber, J. M., Sesmero, M. i Sanchis, A. 2014. Combining Additive Input Noise Annealing and Pattern Transformations for Improved Handwritten Character Recognition. *Expert Systems with Applications*, **41**(18), 8180–8188.

Arora, P., Ganguly, D. i Jones, G. J. 2015. The Good, the Bad and their Kins: Identifying Questions with Negative Scores in Stackoverflow. *Str. 1232–1239 u: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. Association for Computing Machinery.

Barr, C., Jones, R. i Regelson, M. 2008. The Linguistic Structure of English Web-Search Queries. *Str. 1021–1030 u: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Basu, S., Bilenko, M. i Mooney, R. J. 2004. A probabilistic framework for semi-supervised clustering. *Str. 59–68 u: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Berger, A., Caruana, R., Cohn, D., Freitag, D. i Mittal, V. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. *Str. 192–199 u: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Bergstra, J. S., Bardenet, R., Bengio, Y. i Kégl, B. 2011. Algorithms for Hyper-Parameter Optimization. *Str. 2546–2554 u: Advances in Neural Information Processing Systems*.

Bickel, S. i Scheffer, T. 2004. Learning From Message Pairs for Automatic Email Answering. *Str. 87–98 u: European Conference on Machine Learning*. Springer.

Bilenko, M., Basu, S. i Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. *Str. 81–88 u: Proceedings of the 21st International Conference on Machine Learning (ICML-04)*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. springer.

Bogdanova, D., dos Santos, C. N., Barbosa, L. i Zadrozny, B. 2015. Detecting Semantically Equivalent Questions in Online User Forums. *Str. 123–131 u: Proceedings of the 19th Conference on Computational Natural Language Learning, CoNLL 2015, Beijing, China, July 30-31, 2015*.

Brinker, K. 2004. Active learning of label ranking functions. *Str. 17 u: Proceedings of the twenty-first international conference on Machine learning*. ACM.

Bunescu, R. i Huang, Y. 2010. Learning the Relative Usefulness of Questions in Community QA. *Str. 97–107 u: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. i Hullender, G. 2005. Learning to Rank Using Gradient Descent. *Str. 89–96 u: Proceedings of the 22nd international conference on Machine learning*. ACM.

Burges, C. J., Ragno, R. i Le, Q. V. 2007. Learning to Rank with Non-Smooth Cost Functions. *Str. 193–200 u: Advances in neural information processing systems*.

Burke, R. D., Hammond, K. J., Kulyukin, V., Lytinen, S. L., Tomuro, N. i Schoenberg, S. 1997. Question Answering from Frequently Asked Question Files: Experiences with the FAQ Finder System. *AI Magazine*, **18**(2), 57–66.

Campello, R. J., Moulavi, D., Zimek, A. i Sander, J. 2013. A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. *Data Mining and Knowledge Discovery*, **27**(3), 344–371.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F. i Li, H. 2007. Learning to Rank: from Pairwise Approach to Listwise Approach. *Str. 129–136 u: Proceedings of the 24th International Conference on Machine learning*. Association for Computing Machinery.

Carmel, D., Mejer, A., Pinter, Y. i Szpektor, I. 2014. Improving Term Weighting for Community Question Answering Search Using Syntactic Analysis. *Str. 351–360 u: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. Association for Computing Machinery.

Carpineto, C. i Romano, G. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Computing Surveys (CSUR)*, **44**(1), 1–50.

Chang, C.-C. i Lin, C.-J. 2011. LIBSVM: A Library for Support Vector Machines. *Transactions on Intelligent Systems and Technology*, **2**(3), 1–27.

Chen, D. i Manning, C. D. 2014. A Fast and Accurate Dependency Parser using Neural Networks. *Str. 740–750 u: Emnlp*.

Collins, M. i Duffy, N. 2002. New ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. *Str. 263–270 u: Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics.

Contractor, D., Kothari, G., Faruquie, T. A., Subramaniam, L. V. i Negi, S. 2010. Handling Noisy Queries in Cross Language FAQ Retrieval. *Str. 87–96 u: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Cortes, C. i Vapnik, V. 1995. Support-vector Networks. *Machine learning*, **20**(3), 273–297.

Croce, D., Moschitti, A. i Basili, R. 2011. Structured Lexical Similarity via Convolution Kernels on Dependency Trees. *Str. 1034–1046 u: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11.

Cui, H., Wen, J.-R., Nie, J.-Y. i Ma, W.-Y. 2002. Probabilistic query expansion using query logs. *Str. 325–332 u: Proceedings of the 11th international conference on World Wide Web*. ACM.

Davis, J. V., Kulis, B., Jain, P., Sra, S. i Dhillon, I. S. 2007. Information-theoretic metric learning. *Str. 209–216 u: Proceedings of the 24th international conference on Machine learning*. ACM.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. i Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, **41**(6), 391.

dos Santos, C., Barbosa, L., Bogdanova, D. i Zadrozny, B. 2015. Learning Hybrid Representations to Retrieve Semantically Equivalent Questions. *Str. 694–699 u: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J. i Vapnik, V. 1997. Support vector regression machines. *Str. 155–161 u: Advances in neural information processing systems*.

Duh, K. i Kirchhoff, K. 2008. Learning to rank with partially-labeled data. *Str. 251–258 u: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Feng, M., Xiang, B., Glass, M. R., Wang, L. i Zhou, B. 2015. Applying Deep Learning to Answer Selection: A Study and an Open Task. *Str. 813–820 u: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., i dr. 2010. Building Watson: An Overview of the DeepQA Project. *AI magazine*, **31**(3), 59–79.

Figueroa, A. 2017. Automatically Generating Effective Search Queries Directly from Community Question-Answering Questions for Finding Related Questions. *Expert Systems with Applications*, **77**, 11 – 19.

Figueroa, A. i Neumann, G. 2013. Learning to Rank Effective Paraphrases from Query Logs for Community Question Answering. *Str. 1099–1105 u: Proceedings of the 27th AAAI Conference on Artificial Intelligence*, vol. 13.

Figueroa, A. i Neumann, G. 2014. Category-Specific Models for Ranking Effective Paraphrases in Community Question Answering. *Expert Systems with Applications*, **41**(10), 4730–4742.

Filice, S., Castellucci, G., Croce, D. i Basili, R. 2015. KeLP: a Kernel-based Learning Platform for Natural Language Processing. *Str. 19–24 u: ACL (System Demonstrations)*.

Filice, S., Croce, D., Moschitti, A. i Basili, R. 2016. KeLP at SemEval-2016 Task 3: Learning Semantic Relations Between Questions and Answers. *Str. 1116–1123 u: Proceedings of SemEval-2016*.

Fleiss, J. L. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological bulletin*, **76**(5), 378–382.

Friedman, J. H. 2001. Greedy Function Approximation: a Gradient Boosting Machine. *Annals of statistics*, 1189–1232.

Ganitkevitch, J., Van Durme, B. i Callison-Burch, C. 2013. PPDB: The Paraphrase Database. *Str. 758–764 u: Proceedings of NAACL-HLT*. Atlanta, Georgia: Association for Computational Linguistics.

Ghiassi, M., Olschimke, M., Moon, B. i Arnaudo, P. 2012. Automated Text Classification Using a Dynamic Artificial Neural Network Model. *Expert Systems with Applications*, **39**(12), 10967–10976.

Goodfellow, I., Bengio, Y. i Courville, A. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Han, L., Kashyap, A., Finin, T., Mayfield, J. i Weese, J. 2013. UMBC EBIQUITY-CORE: Semantic Textual Similarity Systems. *Str. 44–52 u: Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, vol. 1.

Hirschman, L. i Gaizauskas, R. 2001. Natural language question answering: the view from here. *Natural Language Engineering*, **7**(04), 275–300.

Hochreiter, S. i Schmidhuber, J. 1997. Long Short-term Memory. *Neural Computation*, **9**(8), 1735–1780.

Hoogeveen, D., Verspoor, K. M. i Baldwin, T. 2015. CQADupStack: A Benchmark Data Set for Community Question-Answering Research. *Str. 1–8 u: Proceedings of the 20th Australasian Document Computing Symposium*. ACM.

Huang, T. S., Dagli, C. K., Rajaram, S., Chang, E. Y., Mandel, M. I., Poliner, G. E. i Ellis, D. P. 2008. Active learning for interactive multimedia retrieval. *Proceedings of the IEEE*, **96**(4), 648–667.

Iyyer, M., Boyd-Graber, J. L., Claudino, L. M. B., Socher, R. i Daumé III, H. 2014. A Neural Network for Factoid Question Answering over Paragraphs. *Str. 633–644 u: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Jansen, P., Surdeanu, M. i Clark, P. 2014. Discourse Complements Lexical Semantics for Non-factoid Answer Reranking. *Str. 977–986 u: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Jeon, J., Croft, W. B. i Lee, J. H. 2005. Finding Semantically Similar Questions Based on their Answers. *Str. 617–618 u: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Jeon, J., Croft, W. B., Lee, J. H. i Park, S. 2006. A Framework to Predict the Quality of Answers with Non-textual Features. *Str. 228–235 u: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Jijkoun, V. i de Rijke, M. 2005. Retrieving Answers from Frequently Asked Questions Pages on the Web. *Str. 76–83 u: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. ACM.

Joachims, T. 1998. *Making Large-scale SVM Learning Practical*. Tech. rept. Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

Joachims, T. 2002. Optimizing Search Engines Using Clickthrough Data. *Str. 133–142 u: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Karan, M. i Šnajder, J. 2016. FAQIR – A Frequently Asked Questions Retrieval Test Collection. *Str. 74–81 u: International Conference on Text, Speech, and Dialogue*. Springer.

Karan, M., Šnajder, J. i Dalbelo Bašić, B. 2012. Distributional Semantics Approach to Detecting Synonyms in Croatian Language. *Str. 111–116 u: Information Society 2012 - Eighth Language Technologies Conference*.

Kazai, G., Kamps, J. i Milic-Frayling, N. 2013. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information retrieval*, **16**(2), 138–178.

Kekäläinen, J. 2005. Binary and Graded Relevance in IR evaluations — Comparison of the Effects on Ranking of IR Systems. *Information processing & management*, **41**(5), 1019–1033.

Kim, H. i Seo, J. 2006. High-performance FAQ Retrieval Using an Automatic Clustering Method of Query Logs. *Information Processing & Management*, **42**(3), 650–661.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. *Str. 1746–1751 u: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Kothari, G., Negi, S., Faruquie, T. A., Chakaravarthy, V. T. i Subramaniam, L. V. 2009. SMS Based Interface for FAQ Retrieval. *Str. 852–860 u: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics.

Krizhevsky, A., Sutskever, I. i Hinton, G. E. 2012. Imagenet Classification with Deep Convolutional Neural Networks. *Str. 1097–1105 u: Advances in Neural Information Processing Systems*.

Kullback, S. i Leibler, R. A. 1951. On Information and Sufficiency. *The annals of mathematical statistics*, **22**(1), 79–86.

Latiri, C. C., Yahia, S. B., Chevallet, J. i Jaoua, A. 2003. Query Expansion Using Fuzzy Association Rules between Terms. *Proceedings of JIM*.

Lavie, A. i Denkowski, M. J. 2009. The METEOR Metric for Automatic Evaluation of Machine Translation. *Machine translation*, **23**(2-3), 105–115.

LeCun, Y., Bottou, L., Bengio, Y. i Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

Lee, J.-T., Kim, S.-B., Song, Y.-I. i Rim, H.-C. 2008. Bridging lexical gaps between queries and questions on large online Q&A collections with compact translation models. *Str. 410–418 u: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Lehrer, A. 1974. *Semantic Fields and Lexical Structures*. Amsterdam: North-Holland Publishing Co.

Lei, T., Joshi, H., Barzilay, R., Jaakkola, T., Tymoshenko, K., Moschitti, A. i Màrquez, L. 2016. Semi-supervised Question Retrieval with Gated Convolutions. *Str. 1279–1289 u: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Lelis, L. i Sander, J. 2009. Semi-supervised Density-Based Clustering. *Str. 842–847 u: Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*. IEEE Computer Society.

Levenshtein, V. I. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Str. 707–710 u: Soviet physics doklady*, vol. 10.

Liu, T.-Y., i dr. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval*, **3**(3), 225–331.

Ljubešić, N. i Erjavec, T. 2011. HrWaC and SlWaC: Compiling Web Corpora for Croatian and Slovene. *Str. 395–402 u: Text, Speech and Dialogue*. Springer.

Lombarović, T., Šnajder, J. i Bašić, B. D. 2011. Question Classification for a Croatian QA System. *Str. 403–410 u: Text, Speech and Dialogue*. Springer.

Loper, E. i Bird, S. 2002. NLTK: The Natural Language Toolkit. *Str. 63–70 u: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics.

Lytinen, S. i Tomuro, N. 2002. The Use of Question Types to Match Questions in FAQ Finder. *Str. 46–53 u: AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*.

Mahalanobis, P. C. 1936. On the Generalised Distance in Statistics. *Proceedings of the National Institute of Sciences of India*, **2**(1), 49–55.

Mallapragada, P. K., Jin, R. i Jain, A. K. 2008. Active query selection for semi-supervised clustering. *Str. 1–4 u: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE.

Manning, C. D., Raghavan, P. i Schütze, H. 2008. *Introduction to information retrieval*. Cambridge University Press.

Marom, Y. i Zukerman, I. 2009. An Empirical Study of Corpus-based Response Automation Methods for an E-mail-based Help-desk Domain. *Computational Linguistics*, **35**(4), 597–635.

McCarthy, D. i Navigli, R. 2009. The English lexical substitution task. *Language resources and evaluation*, **43**(2), 139–159.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., i dr. 2011. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, **331**(6014), 176–182.

Mikolov, T., Chen, K., Corrado, G. i Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. *Workshop at ICLR*.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, **38**(11), 39–41.

Mitchell, J. i Lapata, M. 2008. Vector-based Models of Semantic Composition. *Str. 236–244 u: ACL.*

Moraes, R., Valiati, J. F. i Neto, W. P. G. 2013. Document-level Sentiment Classification: An Empirical Comparison Between SVM and ANN. *Expert Systems with Applications*, **40**(2), 621–633.

Moreo, A., Eisman, E. M., Castro, J. i Zurita, J. M. 2013. Learning Regular Expressions to Template-based FAQ Retrieval Systems. *Knowledge-Based Systems*, **53**, 108–128.

Moschitti, A. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. *Str. 318–329 u: Proceedings of the 17th European Conference on Machine Learning*, vol. 4212. Springer.

Moschitti, A. i Quarteroni, S. 2011. Linguistic Kernels for Answer Re-ranking in Question Answering Systems. *Information Processing & Management*, **47**(6), 825–842.

Moschitti, A., Nakov, P., Marquez, L., Magdy, W., Glass, J. i Randeree, B. 2015. Semeval-2015 Task 3: Answer Selection in Community Question Answering. *Str. 269–281 u: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015).*

Nassif, H., Mohtarami, M. i Glass, J. 2016. Learning Semantic Relatedness in Community Question Answering Using Neural Models. *Str. 137–147 u: Proceedings of the 1st Workshop on Representation Learning for NLP*.

Ng, A. Y., Jordan, M. I. i Weiss, Y. 2002. On Spectral Clustering: Analysis and an Algorithm. *Str. 849–856 u: Advances in neural information processing systems*.

Nogueira, B. M., Jorge, A. M. i Rezende, S. O. 2012. Hierarchical confidence-based active clustering. *Str. 216–219 u: Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM.

Oquab, M., Bottou, L., Laptev, I. i Sivic, J. 2014. Learning and transferring mid-level image representations using convolutional neural networks. *Str. 1717–1724 u: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Papineni, K., Roukos, S., Ward, T. i Zhu, W.-J. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *Str. 311–318 u: Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics.

Park, J.-G. i Kim, K.-J. 2013. Design of a Visual Perception Model with Edge-adaptive Gabor Filter and Support Vector Machine for Traffic Sign Detection. *Expert Systems with Applications*, **40**(9), 3679–3687.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. i Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Pena, J. M., Lozano, J. A. i Larranaga, P. 1999. An Empirical Comparison of Four Initialization Methods for the K-means Algorithm. *Pattern recognition letters*, **20**(10), 1027–1040.

Pimentel, M. A., Clifton, D. A., Clifton, L. i Tarassenko, L. 2014. A Review of Novelty Detection. *Signal Processing*, **99**, 215–249.

Platt, J. 1998. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. Tech. rept.

Porter, M. F. 2001. *Snowball: A Language for Stemming Algorithms*.

Punj, G. i Stewart, D. W. 1983. Cluster Analysis in Marketing Research: Review and Suggestions for Application. *Journal of marketing research*, 134–148.

Qiu, X. i Huang, X. 2015. Convolutional Neural Tensor Network Architecture for Community-Based Question Answering. *Str. 1305–1311 u: Proceedings of the International Joint Conference on Artificial Intelligence*.

Rangapuram, S. S. i Hein, M. 2012. Constrained 1-Spectral Clustering. *Str. 1143–1151 u: International Conference on Artificial Intelligence and Statistics*.

Resnik, P. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *Str. 448–453 u: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 1.

Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V. i Liu, Y. 2007. Statistical Machine Translation for Query Expansion in Answer Retrieval. *Str. 464 u: Annual Meeting-Association For Computational Linguistics*.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M. i Gatford, M. 1995. Okapi at TREC-3. *Str. 109–126 u: NIST Special Publication Sp*.

Rumelhart, D. E., McClelland, J. L., Group, P. R., i dr. 1988. *Parallel distributed processing*. Vol. 1. IEEE.

Rus, V., Lintean, M. C., Banjade, R., Niraula, N. B. i Stefanescu, D. 2013. SEMILAR: The Semantic Similarity Toolkit. *Str. 163–168 u: ACL (Conference System Demonstrations)*.

Salton, G., Wong, A. i Yang, C.-S. 1975. A vector space model for automatic indexing. *Communications of the ACM*, **18**(11), 613–620.

Šarić, F., Glavaš, G., Karan, M., Šnajder, J. i Bašić, B. D. 2012. TakeLab: Systems for Measuring Semantic Text Similarity. *Str. 441–448 u: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J. i Williamson, R. C. 2001. Estimating the Support of a High-dimensional Distribution. *Neural computation*, **13**(7), 1443–1471.

Severyn, A. i Moschitti, A. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. *Str. 373–382 u: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Sharp, R., Jansen, P., Surdeanu, M. i Clark, P. 2015. Spinning Straw into Gold: Using Free Text to Train Monolingual Alignment Models for Non-factoid Question Answering. *Str. 231–237 u: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Shental, N., Bar-Hillel, A., Hertz, T. i Weinshall, D. 2004. Computing Gaussian mixture models with EM using equivalence constraints. *Str. 465–472 u: Advances in neural information processing systems*.

Shi, J. i Malik, J. 2000. Normalized Cuts and Image Segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, **22**(8), 888–905.

Smola, A. J. i Vishwanathan, S. 2003. Fast Kernels for String and Tree Matching. *Str. 585–592 u: Advances in neural information processing systems*.

Šnajder, J., Bašić, B. D. i Tadić, M. 2008. Automatic Acquisition of Inflectional Lexica for Morphological Normalisation. *Information Processing & Management*, **44**(5), 1720–1731.

Sneiders, E. 1999. Automated Answering: Continued Experience with Shallow Language Understanding. *Str. 97–107 u: Question Answering Systems. Papers from the 1999 AAAI Fall Symposium*.

Sneiders, E. 2002. Automated question answering using question templates that cover the conceptual model of the database. *Str. 235–239 u: International Conference on Application of Natural Language to Information Systems*. Springer.

Sneiders, E. 2009. Automated FAQ Answering with Question-specific Knowledge Representation for Web Self-service. *Str. 298–305 u: 2009 2nd Conference on Human System Interactions*. IEEE.

Sneiders, E. 2010. Automated Email Answering by Text Pattern Matching. *Str. 381–392 u: International Conference on Natural Language Processing*. Springer.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. i Salakhutdinov, R. 2014. Dropout: a Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15**(1), 1929–1958.

Surdeanu, M., Ciaramita, M. i Zaragoza, H. 2008. Learning to Rank Answers on Large Online QA Collections. *U: Proceedings of the Association for Computational Linguistics*.

Surdeanu, M., Ciaramita, M. i Zaragoza, H. 2011. Learning to Rank Answers to Non-factoid Questions from Web Collections. *Computational Linguistics*, **37**(2), 351–383.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, **abs/1605.02688**(May).

Tomuro, N. i Lytinen, S. 2004. Retrieval Models and Q and A Learning with FAQ Files. *New Directions in Question Answering*, 183–194.

Turney, P. D. i Pantel, P. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of artificial intelligence research*, **37**, 141–188.

Van Craenendonck, T., Dumancic, S. i Blockeel, H. 2017. COBRA: A fast and simple method for active clustering with pairwise constraints. *U: Proceedings of the 26th International Joint Conference on Artificial Intelligence*.

Voorhees, E. M. 1999. The TREC-8 Question Answering Track Report. *Str. 77–82 u: Proceedings of the Text Retrieval Conference*, vol. 99.

Voorhees, E. M. i Harman, D. K. 2005. TREC: Experiment and evaluation in information retrieval. *U: Proceedings of the Text Retrieval Conference*, vol. 1. MIT Press Cambridge.

Voorhees, E. M. i Tice, D. M. 2000. Building a Question Answering Test Collection. *Str. 200–207 u: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., i dr. 2001. Constrained k-means clustering with background knowledge. *Str. 577–584 u: ICML*, vol. 1.

Wang, K., Ming, Z. i Chua, T.-S. 2009. A Syntactic Tree Matching Approach to Finding Similar Questions in Community-based QA Services. *Str. 187–194 u: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Wang, X. i Davidson, I. 2010. Flexible constrained spectral clustering. *Str. 563–572 u: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Wang, X., Qian, B. i Davidson, I. 2014. On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, 1–30.

Wei, J., Bressan, S. i Ooi, B. C. 2000. Mining Term Association Rules for Automatic Global Query Expansion: Methodology and Preliminary Results. *Str. 366–373 u: Proceedings of WISE*, vol. 1. IEEE.

Whitney, A. W. 1971. A Direct Method of Nonparametric Measurement Selection. *IEEE Transactions on Computers*, **100**(9), 1100–1103.

Wieting, J., Bansal, M., Gimpel, K., Livescu, K. i Roth, D. 2015. From Paraphrase Database to Compositional Paraphrase Model and Back. *Transactions of the Association for Computational Linguistics*, **3**, 345–358.

Wu, C.-H., Yeh, J.-F. i Chen, M.-J. 2005. Domain-specific FAQ Retrieval Using Independent Aspects. *ACM Transactions on Asian Language Information Processing (TALIP)*, **4**(1), 1–17.

Wu, C.-H., Yeh, J.-F. i Lai, Y.-S. 2006. Semantic segment extraction and matching for internet FAQ retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, **18**(7), 930–940.

Wu, Q., Burges, C. J., Svore, K. M. i Gao, J. 2010. Adapting Boosting for Information Retrieval Measures. *Information Retrieval*, **13**(3), 254–270.

Xing, E. P., Jordan, M. I., Russell, S. J. i Ng, A. Y. 2003. Distance metric learning with application to clustering with side-information. *Str. 521–528 u: Advances in neural information processing systems*.

Xu, J. i Croft, W. B. 1996. Query Expansion Using Local and Global Document Analysis. *Str. 4–11 u: Proceedings of ACM SIGIR*. ACM.

Xu, Q., desJardins, M. i Wagstaff, K. 2005. Active constrained clustering by examining spectral eigenvectors. *Str. 294–307 u: Discovery Science*. Springer.

Yang, L., Ai, Q., Spina, D., Chen, R.-C., Pang, L., Croft, W. B., Guo, J. i Scholer, F. 2016. Beyond factoid QA: effective methods for non-factoid answer sentence retrieval. *Str. 115–128 u: European Conference on Information Retrieval*. Springer.

Yang, Y., Nie, F., Xu, D., Luo, J., Zhuang, Y. i Pan, Y. 2012. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(4), 723–742.

Yang, Y. i Liu, X. 1999. A Re-examination of Text Categorization Methods. *Str. 42–49 u: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Yih, W.-t., Chang, M.-W., Meek, C. i Pastusiak, A. 2013. Question Answering Using Enhanced Lexical Semantic Models. *Str. 1744–1753 u: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Žmak, L. 2009. Sustav za pretraživanje zbirke često postavljenih pitanja na hrvatskom jeziku. *Diplomski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu*, 1–67.

Životopis

Mladen Karan rođen je 28. lipnja 1987. godine u Čakovcu u Hrvatskoj. Preddiplomski studij računarstva završio je 2009. godine na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Na istome je fakultetu završio i diplomski studij računarstva (smjer računarska znanost) s temom diplomskog rada "Mogućnosti korištenja paralelnih algoritama u linearnoj optimizaciji".

Od svibnja 2011. godine zaposlen je na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva kao stručni suradnik na više projekata s privredom. Od ožujka 2015. godine zaposlen je kao asistent na istom zavodu. Bio je uključen u nastavne aktivnosti zavoda na predmetima "Strojno Učenje", "Neizrazito, evolucijsko i neuroračunarstvo", "Umjetna inteligencija", "Analiza i pretraživanje teksta", "Statistička analiza podataka", "Formalne metode u oblikovanju sustava" i "Oblikovanje programske potpore" te je asistirao u vođenju četiri završna i diplomska rada.

Njegovi istraživački interesi obuhvaćaju područja pretraživanja informacija, obrade prirodnoga jezika i računalne lingvistike. U suautorstvu je objavio 15 radova na međunarodnim znanstvenim skupovima i dva rada u časopisima s međunarodnom recenzijom, od kojih jedan u časopisu indeksiranom u bazi Current Contents. Član je strukovne udruge ACL (Association for Computational Linguistics). Govori engleski i njemački jezik.

Popis objavljenih radova

Radovi u časopisima

1. Karan, M., Šnajder, J., "Paraphrase-focused Learning to Rank for Domain-specific Frequently Asked Questions Retrieval", Expert Systems with Applications, Vol. 91, siječanj 2018., str. 418–433
2. Karan, M., Glavaš, G., Šarić, F., Šnajder, J., Dalbelo Bašić, B., "CroNER: Recognizing Named Entities in Croatian Using Conditional Random Fields", Informatics, Vol. 37, prosinac 2013., str. 165–172

Radovi na međunarodnim znanstvenim skupovima

1. Šaina F., Kukurin T., Puljić L., Karan M., Šnajder J., "TakeLab-QA at SemEval-2017 Task 3: Classification Experiments for Answer Retrieval in Community QA", Eleventh International Workshop on Semantic Evaluation (SemEval), 2017., str. 339–343
2. Karan, M., Šnajder, J., "Detecting Non-covered Questions in Frequently Asked Questions Collections", International Conference on Applications of Natural Language to Information Systems, 2017., str. 387 –390
3. Karan, M., Šnajder, J., "FAQIR – A Frequently Asked Questions Retrieval Test Collection", Text, Speech and Dialogue, 2017., str 74–81
4. Karan M., Šnajder J., Širinić D., Glavaš G. "Analysis of Policy Agendas: Lessons Learned from Automatic Topic Classification of Croatian Political Texts.", Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2016), ACL 2016., str 12–21
5. Tutek M., Sekulić I., Gombar P., Paljak I., Čulinović F., Boltužić F., Karan M., Alagić D., and Šnajder J., "TakeLab at SemEval-2016 Task 6: Stance Classification in Tweets Using a Genetic Algorithm Based Ensemble.", Tenth International Workshop on Semantic Evaluation (SemEval), 2016., str. 464–468
6. Karan, M., Glavaš, G., Šnajder, J., Dalbelo Bašić, B., Vulić, I., Moens, M.F., "TKL-BLIIR: Detecting Twitter Paraphrases with TweetingJay", 9th International Workshop on Semantic Evaluation (SemEval), Association for Computational Lingustics, 2015., str 70-74
7. Karan, M., Šnajder, J., "Evaluation of Manual Query Expansion Rules on a Domain Specific FAQ Collection", Conference and Labs of the Evaluation Forum, Volume 9283 of the series Lecture Notes in Computer Science., 2015., str. 248-253
8. Karan, M., Pintar D., Skočir Z., Vranić M., Alajković, A., Milojević, J., Pleša, M., "The impact of training data tailoring on demand forecasting models in retail", 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014., str 1636-1641
9. Zuanović, L., Karan, M., Šnajder, J., "Experiments with Neural Word Embeddings for Croatian", Ninth Language Technologies Conference, Information Society (IS-JT), 2014., str 69-72
10. Karan, M., Žmak, L., Šnajder, J., "Frequently Asked Questions Retrieval for Croatian Based on Semantic Textual Similarity", 4th Biennial International Workshop on Balto-Slavic Natural Language Processing (BSNLP), Association for Computational Linguistics, 2013. str. 24-33
11. Glavaš, G., Karan, M., Šarić, F., Šnajder, J., Mijić, J., Šilić, A., Dalbelo Bašić, B., "Cro-NER: A State-of-the-Art Named Entity Recognition and Classification for Croatian", Eig-

hth Language Technologies Conference, Information Society (IS-LTC), 2012., str. 73–78

12. Karan, M., Šnajder, J., Dalbelo Bašić, B., “Distributional Semantics Approach to Detecting Synonyms in Croatian Language”, Eighth Language Technologies Conference, Information Society (IS-LTC), 2012., str. 111-116
13. Karan, M., Šnajder, J., Dalbelo Bašić, B., “Evaluation of Classification Algorithms and Features for Collocation Extraction in Croatian”, Eight International Conference on Language Resources and Evaluation (LREC), 2012., str. 657-662
14. Šarić, F., Glavaš, G., Karan, M., Šnajder, J., Dalbelo Bašić, B., “TakeLab Systems for Measuring Semantic Text Similarity”, The First Joint Conference on Lexical and Computational Semantics (*SEM), 2012., str. 441–448
15. Karan, M., Skorin-Kapov N., “A Branch and Bound Algorithm for the Sequential Ordering Problem”, 34th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2011., str. 452-457

Biography

Mladen Karan was born on June 28, 1987 in Čakovec, Croatia. He received his B.Sc. in Computing from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2009 and M.Sc. in Computer Science from the same university in 2011 (thesis title: "Possibilities for Using Parallel Algorithms in Linear Optimization").

From May 2011 he was employed as a project associate at the Department of Electronics, Microelectronics, Computer and Intelligent Systems at the Faculty of Electrical Engineering and Computing on several commercial projects. From March, 2015 he is employed as a teaching assistant at the same department. He was inolved in the Department's educational activities within the following courses: Machine Learning, Fuzzy, Evolutionary, and Neurocomputing, Artificial Intelligence, Text Analysis and Retrieval, Statistical Data Analysis, Formal Methods in System Design, and Software Design. He also assisted in supervising four bachelor and masters theses.

His research interests include information retrieval, natural language processing, and computational linguistics. He has co-authored 15 conference papers and two journal papers, one of which in a journal indexed in the Current Contents. He is a member of the ACL (Association for Computational Linguistics). He is fluent in English and German.