EkoSmart asistent za iskanje po integracijski platformi

David Golob

IJS
Jamova 39, 1000 Ljubljana
+386 1 477 3147

david.golob@ijs.si

Matjaž Gams

IJS
Jamova 39, 1000 Ljubljana
matjaz.gams@ijs.si

Aleš Tavčar

IJS
Jamova 39, 1000 Ljubljana
ales.tavcar@ijs.si

**Povzetek**

V tem prispevku opišemo delovanje asistenta za iskanje po EkoSmart integracijski platformi. Namen asistenta je, da omogoči lažje in hitrejše iskanje informacij po integracijski platformi. Asistent je še posebej priročen za starejše in manj vešče uporabnike internetnih tehnologij. Asistent uporabi lematizator za določitev samostalnikov v uporabnikovem vprašanju. Asistent nato uporabi dobljene samostalnike za iskanje po integracijski platformi. Če asistent ne pridobi nobenih zadetkov za statistične podatke ali API-je, se obrne na starega EkoSmart asistenta za odgovor.

**Ključne besede**

Asistent, EkoSmart, integracijska platforma.

# Uvod

Namen asistenta za iskanje po EkoSmart integracijski platformi, je da omogoči poenostavljeno iskanje po EkoSmart integracijski platformi, kjer so na voljo različni statistični podatki ter API-ji (npr. API za občinskega asistenta ter API za zdravstvene čakalne vrste).

Namen projekta EkoSmart je razviti ekosistem pametnega mesta z vsemi podpornimi mehanizmi, ki so potrebni za učinkovito, optimizirano in postopno integracijo posameznih področij v enovit in povezan sistem vrednostnih verig. Projekt se je začel 1. Avgusta 2016 ter končal 31. Julija 2019. Vrednost projekta ja 8,66 mio EUR. V projektu je sodelovalo 25 partnerjev.

Integracijska platforma je sestavljena iz treh delov: podatkov, produktov ter storitev. Kot rečeno, se z našim asistentom osredotočimo na iskanje po podatkih ter storitvah (API-jih). V času pisanja, so bili na integracijski platformi dostopni 4 nabori podatkov (z imeni »Transport«, »Slovenski kraji«, »Zdravje« in »Okolje«) ter 13 storitev (API-jev). Kot platforma za dostop do API-kev se uporablja odprto-kodna platforma **wso2** [1]. Le ta platforma omogoča objavljanje ter prenos API-jev. Prav tako platforma omogoča upravljanje z objavljenimi API-ji v smislu monetizacije API-jev (i.e. koliko API-jev za koliko denarja). Za podatke se uporablja odprto-kodna **ckan** [2] platforma, ki omogoča objavi ter prenos podatkov.

Uporabnik se pogovarja z asistentom preko uporabniškega vmesnika, ki je dostopen na [3]. Uporabnik v uporabniški vmesnik vnese stavek/vprašanja o stvari, ki ga zanima (npr. »zanima me slovenija«) nato asistenta s pomočjo razumevanja naravnega jezika obdela uporabnikov stavek/vprašanje, pridobi stvar zanimanja ter izvede iskanje po naboru podatkov ter API-jev ter vrne primerne zadetke. Asistent, deluje tako, da iz uporabnikovega stavka, s pomočjo lematizatorja, izlušči samostalnike ter jih pretvori v osnovno obliko. Tako dobljene lematizirane samostalnike nato uporabimo za iskanje po bazi podatkov ter API-jev. Za iskanje po bazi podatkov že imamo na voljo storitev, ki nam sama vrne ustrezne zadetke. Za uporabo te storitve moramo samo priskrbeti iskane besede, za le te uporabimo lematizirane samostalnike. Za iskanje po bazi API-jev nimamo na voljo iskalne storitve, zato moramo iskanje implementirati sami. Vsak API ima svoje oznake (angl. tags), ki služijo kot ključne besede. Princip iskanja po bazi API-jev je ta, da za API lematiziramo samostalnike v oznakah (kot to naredimo za uporabnikov stavek/vprašanje) ter nato primerjamo lematizirane samostalnike v oznakah z lematiziranimi samostalniki v uporabnikovem vprašanju ter si zabeležimo API kot zadetek v primeru ujemanja.

Uporaba asistentov/chatbotov-ov se je v zadnjem času z napredkom procesiranja naravnega jezika zelo povečala. Primer uporabe asistenta za avtomatsko sklicevanje sestankov je opisan v [4]. V tem članku je opisan asistent za sklicevanje sestankov preko elektronske pošte. Asistent iz naslova ter vsebine elektronske pošte, s pomočjo ključnih besed, identificira namen elektronske pošte. Če je namen sklic sestanka, asistent prebere celotno vsebino pošte ter s pomočjo tehnologije iskanja vzorcev izlušči ure ter kraj sestanka. Nato asistent preveri v koledarju če je sestanek mogoč, glede na to pošlje odgovor s podatki o možnosti sestanka. Z nekaj izmenjavami elektronske pošte, je asistent možen potrditi sestanek.

Podoben asistent za klepet, je opisan v [5]. Ta asistent je namenjen za namen servisiranja strank. In sicer, asistent dobi vprašanje/zahtevo od stranke, z uporabo procesiranja naravnega jezika je sposoben izluščiti namen in ključne besede povezane z namenom, s katerimi nato naredi iskanje po bazi znanja ter nato vrne odgovor stranki.

Prav tako je v [6] opisan sistem za tvorjenje asistenta za interakcijo s spletno stranjo. Asistenta uporabnik vodi/sprašuje z uporabo kazalca (na zaslonu) ter z govorom. Sistem je sposoben pretvoriti govor v tekst, ki ga nato pošlje asistentu. Sistem pregleda spletno stran, ko se le ta naloži, ter vstavi bazo znanja iz smiselnih semantičnih struktur (e.g. tabel, grafov, slik). Ko asistent sprejme tekstovni ukaz, pridobi namen ter entitete z uporabo agenta Watson. Z namenom ter entitetami nato pregleda bazo znanja ter ponudi odgovor. V primeru, da v bazi znanja ne odkrije želenih podatkov, asistent prosi uporabnika za bolj podrobne napotke (npr. lahko ga prosi, da z kazalnikom pokaže na stvar zanimanja e.g. stolpec v tabeli).

# Mehanizem delovanja asistenta

Lematizator, uporabljen v asistentu, je narejen posebej za Slovenski ter Srbo-Hrvaški jezik in je prosto dostopen na [7].

## Iskanje po bazi podatkov

Za iskanje po bazi podatkov (od integracijske platforme) se uporabi platformo, ki je dostopna na naslovu <http://wso2.lavbic.net:5000/sl/>. Preko te platforme nato iščemo zadetke za lematizirane samostalnike, npr. če želimo rezultate za samostalnik “Slovenija”, uporabimo zahtevo <http://wso2.lavbic.net:5000/api/3/action/package_search?q=slovenija>, ki nam vrne odgovor v JSON formatu, ki vsebuje podatke o zadetkih iskanja. Za vsak zadetek iskanja (o podatkih), si nato shranimo

* Url naslov, kjer so podatki dostopni.
* Naslov podatkov.
* Opis podatkov.

API za iskanje po bazi podatkov <http://wso2.lavbic.net:5000/api/3/action/package_search?q=>, išče po oznakah podatkov ter po opisu podatkov, tako vrne vse zadetke, ki ustrezajo posredovanemu konceptu/besedi. Npr. če nas zanima ozračje lahko pošljemo zahtevek <http://wso2.lavbic.net:5000/api/3/action/package_search?q=ozra%C4%8Dje> ter dobimo JSON odgovor, del le tega je predstavljen v Slika 1. Najbolj pomemben del JSON odgovora je »resources« (slo. viri), ki je v delu »results« (slo. rezultati). Pod viri imamo ime podatkov (»name«), opis podatkov (»description«) ter spletni naslov podatkov (»url«).



Slika 1:Primer dela API odgovora za iskanje po podatkih

## Iskanje po bazi API-jev

Za iskanje po bazi podatkov o API-jih nimamo vgrajenega iskalnika v iskalni platformi (<http://wso2.lavbic.net:5000/sl/>). Zato pridobimo nabor API-jev z zahtevo <https://wso2.lavbic.net:9443/api/am/store/v0.11/apis>, ki nam vrne odgovor v JSON obliki (poglej sliko Slika 3 za primer odgovora). Ta odgovor vsebuje identifikacijski podatek (ID) za vsak API, z le tem podatkom pridobimo celotne podatke o določenim API-jem; npr. če želimo pridobiti celotne podatke o API-ju z ID-jem **218466a1-ac3b-464d-ba21-f2605f8391e8**, pošljemo sledečo zahtevo <https://wso2.lavbic.net:9443/api/am/store/v0.11/apis/218466a1-ac3b-464d-ba21-f2605f8391e8>. Ta zahteva nam vrne odgovor v JSON format, v katerem so vsi podatki o izbranem API-ju. Za potrebe našega asistenta, si shranimo sledeče podatke:

* Ime API-ja.
* Opis API-ja.
* Oznake API-ja.
* Lematizirane oznake API-ja.
* Url naslov, kjer izbran API na voljo.

Za lematizacjio API oznak uporabimo isti lematizator, kot za lematizacijo uporabnikovega vprašanja/stavka. Prav tako iz vsake oznake (za izbran API), z lematizatorjem, izluščimo samostalnike ter jih pretvorimo v korensko/osnovno obliko. Tako obdelane oznake imenujemo lematizirane oznaka (kot v zgornjem seznamu).

Za primer API oznak izberemo API z ID-jem **0815aeb4-b191-48ca-aad5-244f12be701c** in imenom **24alife**.

Primer JSON odgovora za ta API je na sliki Slika 2. Pomembni deli JSON odgovora so: ime (»name«), opis (»description«), oznake (»tags«) ter spletni naslovi (»https« in »http«).



Slika 2:Primer zahtevka za podatke o določenem API-ju



Slika 3: Primer odgovora na zahtevek za podatke o vseh API-jih

 Ta API ima oznake (angl. tags) »**eZdravje«**, »**24alife**« in »**aktivnost**«. Te oznake služijo kot ključne besede, ki opišejo API. Kot rečeno, se oznake s pomočjo lematizatorja pretvorijo v osnovno obliko. V tem primeru so vse oznake že v osnovni obliki, tako da so lematizirane oznake »**eZdravje«**, »**24alife**« in »**aktivnost**«. V drugem primeru si ogledamo API z ID-jem **5892db13-f93e-4ff6-a812-0da0a941e7a6,** imenom»**CakalneVrste**« ter oznakami »**posegi«**, »**ezdravje**«, »**zdravnik**«, »**čakalne**«, »**zdravstvene ustanove**«. V oznakah obdržimo le samostalnike v osnovni obliki. V našem primeru je oznaka »**čakalne**« pridevnik, zato se te oznake ne upošteva. Prav tako se oznaka »**zdravstvene ustanove**« (v tej oznaki je samostalnik samo »ustanove«) s pomočjo lematizatorja pretvori v »**ustanova**«. Tako nam lematizator za ta primer vrne sledeče lematizirane oznake: »**poseg«**, »**ezdravje**«, »**zdravnik**«, »**ustanova**«.

Podatke o API-jih zaradi časovne zahtevnosti ni smiselno shranjevati vsakič, ko uporabnik vnese vprašanje/stavek, v ta namen uporabimo “caching”, kar pomeni, da si vsakič, ko je spletna stran z uporabniškim vmesnikom ( [3]) naložena, shranimo podatke o API-jih v podatkovno bazo, iz katere nato beremo API podatke.

Iskanje po API podatkih (ki jih shranimo ob naložitvi uporabniškega vmesnika) poteka po sledečem postopku. Vsak lematiziran samostalnik (iz uporabnikovega vprašanja/stavka) primerjamo z lematiziranimi API oznakami (za vsak shranjen API), primerjava je narejena eksaktno (obe besedi se morata ujemati), za vsako ujemanje si shranimo ujemajoč API. Na tak način naredimo iskanje po API podatkih.

Z postopkoma za iskanje po statističnih podatkih ter API-jih pridobimo podatke, ki jih nato v določeni meri prikažemo uporabniku. Za statistične podatke prikažemo največ štiri zadetke (za vse samostalnike v uporabnikovem vprašanju/stavku skupaj), prav tako za API-je prikažemo največ štiri zadetke (za vse samostalnike v uporabnikovem vprašanju/stavku skupaj). Štirje zadetki, ki so prikazani so prvi zadetki, ki so bili najdeni z iskanjem po bazi podatkov ter API-jev. Npr. če so v uporabniškem stavku trije samostalniki, potem je vrstni red samostalnikov za iskanje po bazi podatkov ter API-jev kar vrstni red samostalnikov v stavku; npr. za stavek »Zanima me zdravje, slovenija in okolje«, je vrstni red iskanja po bati podatkov (ter API-jev) »zdravje«, »slovenija«, »okolje«. Za vsak samostalnik pridobimo določeno število zadetkov, na koncu pa prikažemo (največ) prvi štiri zadetke, ki so bili najdeni (za vse samostalnike skupaj). Tak mehanizem velja za iskanje po bazi podatkov ter bazi API-jev. Npr. da za primer »Zanima me zdravje, slovenija in okolje« predpostavimo, da ima (prvi) samostalnik »zdravje« 5 zadetkov, potem prikažemo prve 4 zadetke, ki so bili najdeni za »zdravje«, če pa npr. ima »zdravje« 2 zadetka ter »slovenija« 3 zadetke, potem prikažemo 2 zadetka za »zdravje« ter prva 2 zadetka za »slovenija« (le te dva zadetka sta prva zadetka, ki sta bila najdena v bazi podatkov oz. bazi API-jev). Vrstni red prikazanih zadetkov za posamezen samostalnik, je enak vrstnemu redu zadetkov v JSON odgovoru (za iskanje po bazi samostalnikov) oz. vrstnemu redu API-jev v bazi za iskanje po bazi API-jev.

## Stari EkoSmart asistent

V primeru, da asistent ne najde nobenih zadetkov za statistične podatke in API-je, asistent posreduje uporabnikov stavek/vprašanje staremu EkoSmart asistentu. Poizvedba po starem EkoSmart asistentu je narejena s sledečo zahtevo <http://www.projekt-asistent.si/ekosmart/ask?question=>. Npr. če uporabnik vpraša/napiše “ekosmart” (in asistent ne najde nobenih zadetkov za statistične podatke ter API-je), dobimo odgovor starega EkoSmart asistenta s sledečim zahtevkom <http://www.projekt-asistent.si/ekosmart/ask?question=ekosmart>. Uporaba starega EkoSmart asistenta je uporabna za t.i. “small talk”, ki ga le ta asistent razume.

# Shema

Na sliki Slika 4 je shema mehanizma delovanja asistenta.



Slika 4: Shema delovanja EkoSmart asistenta za iskanje po integracijski platformi

Na sliki Slika 5 je ponazorjen mehanizem delovanja iskanja po bazi podatkov. Vhodni podatek v mehanizem so uporabnikovi lematizirani samostalniki (v osnovni obliki) izhodni podatek pa so maksimalno 4 zadetki.



Slika 5: Shema delovanja algoritma za iskanje po statističnih podatkih

Na sliki Slika 6 je opisan mehanizem za iskanje po bazi API-jev. Kot vhodna podatka sta uporabnikovi lematizirani samostalniki (v osnovni obliki) ter shranjeni API podatki iz baze, izhodni podatek pa so maksimalno 4 API zadetki.



Slika 6: Shema delovanja algoritma za iskanje po API-jih

# Rezultati in primeri.

Prototipno rešitev smo namestili na spletno mesto [3]. Začetna inicializacija traja malce dalje (10-20 sekund) zaradi »prefetch« podatkov in njihovo pred procesiranje (kot je identifikacija samostalnikov ter njihova pretvorba v osnovno obliko). Nadaljnje poizvedbe se izvedejo brez časovnih zamikov.

Kot primer navedemo v uporabniškem vmesniku vprašamo “zanima me slovenija”. Agent nam izpiše 4 zadetke za statistične podatke ter 1 zadetek za API-je.



Slika 7:Primer uporabe

Če npr. vprašamo “zanimajo me občine”, dobimo kot odgovor 1 zadetek za API-je (izbirčnež).



Slika 8: Primer uporabe

Če pa v uporabniški vmesnik vnesemo “Živjo” (v tem primeru nimamo nobenih zadetkov za statistične podatke ter API-je, zato se asistent obrne na starega EkoSmart asistenta), nas agent pozdravi ter se predstavi.



Slika 9:Primer uporabe

# Zaključek

V tem prispevku smo opisali delovanje EkoSmart asistenta za iskanje po integracijski platformi. Asistent uporablja robusten mehanizem za iskanje po bazi podatkov ter API-jev. In sicer asistent uporablja lematizator za identifikacijo samostalnikov in njihovo pretvorbo v osnovno obliko. Kljub malo daljši začetni inicializaciji poteka nadaljnja z asistentom nemoteno brez časovnih zamikov. Asistent je prestal različne preizkušnje uporabnikov ter se izkazal za robustna rešitev problema iskanja po integracijski platformi.

# Literatura

|  |  |
| --- | --- |
| [1]  | [Elektronski]. Available: https://wso2.com/. |
| [2]  | [Elektronski]. Available: https://ckan.org/. |
| [3]  | IJS, „EkoSmart asistnet za iskanje po integracijski bazi,“ [Elektronski]. Available: https://ekosmart.docker-e9.ijs.si/. |
| [4]  | N. Mhatre, K. Motani, M. Shah in S. Mali, „Donna Interactive Chat-bot acting as a Personal Assistant,“ *International Journal of Computer Applications,* pp. 6-11, 2016.  |
| [5]  | „Shallow Parsing Natural Language Processing Implementation for Intelligent Automatic Customer Service System,“ v *ICACSIS*, Jakarta, 2014.  |
| [6]  | M. Peveler, J. O. Kephart in H. Su, „Reagent: Converting Ordinary Webpages into Interactive Software Agents,“ v *Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, 2019.  |
| [7]  | IJS, „Lematizator za slovenski in srbo-hrvaški jezik,“ [Elektronski]. Available: https://repo.ijs.si/DIS-AGENTS/reldi-tagger. |