

SecondEGO – virtualni pomočnik za vsakogar

Miro Romih

Amebis, d. o. o., Kamnik
Bakovnik 3, 1241 Kamnik
miro.romih@amebis.si

Povzetek

Prispevek predstavlja sistem SecondEGO, ki je namenjen izdelavi lastnih virtualnih pomočnikov. Ciljna publika sistema so vsi tisti, ki želijo na svojih spletnih straneh uporabnikom ponuditi pomoč v obliki direktnih odgovorov na njihova vprašanja. Ti odgovori se generirajo tudi ob pomoči jezikovnih tehnologij, možno pa je vključevati tudi naprednejše funkcije.

SecondEGO – virtual agent for everyone

This paper presents the SecondEGO system, which is designed to create your own virtual agents. The target audience of the system are all those who wish to offer assistance in the form of direct answers to user's questions on their websites. These answers are generated with the help of language technologies, but it is also possible to include other advanced functions.

1. Uvod

Običajno imamo na voljo več načinov, kako priti do informacije. Vzemimo na primer, da smo na avtobusni postaji in želimo izvedeti, kdaj odpelje prvi avtobus v nek kraj. Na zidu je obešen vozni red, poleg njega pa je okence za informacije. Nekateri si bodo čas odhoda prebrali iz voznega reda, drugi pa bodo odgovor iskali pri okencu za informacije.

Podobno je na spletu. Določeno informacijo na neki spletni strani lahko poiščemo sami, lahko pa, če obstaja seveda, preko pogovornega okenca o tem povprašamo operaterja, ki na naše vprašanje čaka v ozadju. Če operater (izven delovnega časa) ni dostopen, lahko njegovo vlogo prevzame virtualni pomočnik, ki je sposoben odgovarjati vsaj na najpogostejša vprašanja uporabnikov.

2. Namen članka

Namena članka je predstavitev sistema SecondEGO, ki ga je razvilo podjetje Amebis. Dostopen je na naslovu www.secondego.com, omogoča pa izdelavo lastnih virtualnih pomočnikov na enostaven in hiter način.

Poleg samega (vidnega dela) portala je predstavljeno tudi njegovo ozadje, ki vključuje nekatere zanimive jezikovne tehnologije.

3. Virtualni pomočniki

Virtualni pomočniki, kakor jih poimenujem v tem prispevku, so programi, zasnovani za konverzacijo z uporabniki. Imajo številna imena – virtualni asistenti, navidezni agenti, programirani sogovorniki itd. V angleščini obstaja nekaj sto izrazov, ki opisujejo te programe oz. sisteme za različne namene (npr. »virtual agent«, »virtual assistant«, za klepetanje »chatbot« ali »bot«).

Tudi sistemov oz. platform, ki podpirajo to funkcionalnost, je precej, z njihovo pomočjo pa danes deluje na tisoče bolj ali manj uporabnih virtualnih pomočnikov. Med bolj znanimi starejšimi in še danes delujočimi sta Eliza (Weizenbaum, 1966) in Alice (Wallace, 2003), med sodobnimi pa izstopata Siri (Apple) in Watson (IBM). Vsak od omenjenih predstavlja tudi nov mejnik tehnologije za komunikacijo oz. dialog v naravnem jeziku.

Vsak virtualni pomočnik je običajno namenjen določeni ciljni skupini in odgovarja na vprašanja z bolj ali manj omejenega področja. Največkrat se jih uporablja za podporo uporabnikom (tudi kot neke vrste »pametni« iskalnik), kot marketinško orodje (za večji odziv uporabnikov), za zabavo (klepetanje), ali kot dodatek uporabniškemu vmesniku za lažjo, hitrejšo in naravnejšo komunikacijo.

Kot je bilo že omenjeno, obstaja veliko število sistemov, s katerimi je mogoče graditi virtualne pomočnike. Večina je zelo enostavnih, in omogoča dialog v poljubnem naravnem jeziku. Je pa za slovenski jezik treba v takem sistemu ustvariti bazo vprašanj in odgovorov povsem na novo, saj za zdaj še ni na voljo nobene baze, ki bi jo lahko uporabili kot osnovo. Izgradnja take baze namreč zahteva precej truda, če želimo, da bo osnovno »znanje« kolikor toliko zadovoljivo. Še posebej za pregibni jezik, kot je naš, saj je v teh sistemih vse potrebne besedne oblike za zdaj treba vpisovati ročno.

Naprednejši sistemi sicer imajo vgrajeno boljše jezikovno podporo, ampak največkrat le za angleščino, sem in tja še za kak drug svetovni jezik, za slovenščino pa te podpore trenutno ni.

Za izdelavo virtualnega pomočnika, ki bi se znal pogovarjati v slovenskem jeziku, smo tako obsojeni na uporabo enostavnih mehanizmov vnosa znanja, ki zahtevajo veliko ročnega dela, ali pa razviti sistem, ki bi imel vgrajene mehanizme za obvladovanje slovenščine.

V Amebisu smo se odločili za drugo možnost in razvili svoj lastni sistem, v katerega lahko poljubno dodajamo različne module. Enostavne/osnovne, specifične za določenega uporabnika, pa tudi nekatere take, ki so bili razviti za druge sisteme in omogočajo uporabo že razvitih baz znanja za tuje jezike.

4. SecondEGO

Sistem SecondEGO je sestavljen iz jedra in vmesnika/portala, s katerim je sistem mogoče upravljati. Z njim lahko uporabniki enostavno in hitro kreirajo svoje virtualne pomočnike, jih učijo, testirajo in analizirajo njihovo delovanje ter uporabo.



Slika1. Grafična podoba sistema SecondEGO

4.1. Jedro

SecondEGO je vzorčno voden sistem. Deluje na osnovi vzorcev, v katerih so odgovori na vnaprej določena vprašanja ali ključne besede. Sistem postavlja vprašanje primerja z vsemi vzorci in izbere tistega, ki se po ključnih besedah najbolj ujema. Tako v osnovi delujejo praktično vsi sistemi za uporabo virtualnih pomočnikov.

Prva značilnost, ki sistem SecondEGO razlikuje od večine primerljivih rešitev, je modularna zgradba. To pomeni, da so vzorci v posameznih modulih lahko zapisani v več različnih formatih, ki jih sistem podpira. Modulov določenega formata je lahko poljubno veliko, vsak modul pa lahko vsebuje poljubno število vzorcev.

Druga razlika je, da se za izbrane jezike vhodno vprašanje najprej morfološko, sintaktično in semantično analizira. Na osnovi rezultatov te analize je veliko več možnosti za izbiro ustreznega odgovora, seveda pa morajo vzorci omogočati zapis teh informacij. Zato smo razvili poseben skriptni jezik. Gre za podoben jezik, kot je ChatScript (ChatScript Open Source project), le da je njegova sintaksa drugačna, saj smo ga razvili že precej let prej, vanj pa je vgrajena podpora za slovenski in angleški jezik. Z večjim dodatnim vložkom je mogoče vgraditi podporo tudi za druge jezike.

V tem jeziku so napisani vsi vgrajeni moduli znanja in nekaj specifičnih modulov za določene uporabnike (npr. virtualna računovodkinja Zdenka podjetja Pronet Kranj).

Vnos znanja na ta način je veliko hitrejši, omogoča vnos lem, pomenov, uporabo referenčnih datotek, vključitev zunanjih podatkovnih baz, ter poljubnega števila lastno sprogramiranih novih funkcij. Z njihovo pomočjo lahko realiziramo praktično vse, kar uporabniki zahtevajo od virtualnega pomočnika.

Vzorci so lahko zelo enostavni in lovijo točno določen niz besed (slika2) ali določene ključne besede (slika3). Lahko pa namesto besed na osnovi pomenske analize vprašanja in v skriptni jezik vgrajene obsežne baze pomenov, povezanih v mrežo nadpomenov, podpomenov in drugih pomenskih relacij, v vzorcih uporabljamo tudi naprednejše funkcije (slika4).

```
$ (25) dobro odgovarjaš > Hvala.
```

Slika2. Enostavni vzorec

```
$ @KljučneBesede("nič|ničesar", "ne", "veš")
>
[_tip == 0] Pa me poskusite vprašati drugače.
[_tip == 1] To pa ni res.
```

Slika3. Lovljenje ključnih besed

```
$ @_JeBeseda("kaj je|kva je|kuga je|ka
je|kaj so|kva so|kuga so|ka so|kdo je|kdo
so|kaj veš o|kva veš o|kuga veš o|ka veš
o|poznaš|ali poznaš|a poznaš|al poznaš")
@OsnovniNadpomen([[zdravnik{0:0:0}]]
>
To je vrsta zdravnika. Podobno kot
@IzberiPodpomen([[zdravnik{0:0:0}]], #1)[So--
i---e].
```

Slika4. Uporaba nadpomenov in podpomenov

Tako se npr. z uporabo lem in pomenov v vzorcih lahko znebimo ročnega vnašanja besedni oblik, če to želimo, na drugi strani pa lahko uporabnik z uporabo poenostavljenih vzorcev enostavno in povsem zadovoljivo reši večino problemov, ki jih mora rešiti njegov virtualni pomočnik (slika5).



Slika5. Poenostavljena predstavitev vzorcev

Ena od zanimivih jezikovnotehnoloških funkcij, ki je vgrajena v sistem SecondEGO, je tudi obravnava oz. upoštevanje tipkarskih napak. Mnogokrat se namreč zgodi, da se ljudje pri pisanju vprašanja zatipkajo, potem pa se začudeno sprašujejo, kako to, da virtualni pomočnik ne zna odgovoriti na tako enostavno vprašanje. Svoje tipkarske napake velika večina seveda ne opazi, še kako pa ta zmoti funkcijo za primerjavo ključnih besed, ki zaradi zatipkane besede ne najde ustreznega vzorca. V sistemu SecondEGO smo primerjavo ključnih besed nadgradili tako, da se pri primerjavi vsake besede do neke mere upoštevajo tudi vse morebitne tipkarske napake. Zaradi časovne optimizacije seveda šele potem, ko v prvem prehodu ne najdemo točnih zadetkov. Taka funkcionalnost se je pokazala za zelo uporabno.

Ker je sistem zasnovan tako, da lahko vključujemo različne module, pripravljamo še dva, ki bosta pripomogla k dodatni uporabnosti sistema in pripomogla k dodatni konkurenčnosti v primerjavi z drugimi.

Prvi modul bo lahko vključeval zapis znanja v formatu AIML (Wallace, 2003, 2005; AIML), ki je standard na tem področju. Za različne jezike je na voljo kar nekaj že narejenih in celo prosto dostopnih baz znanja v tem formatu, ki jih bo mogoče enostavno vključiti. Za slovenščino po nam znanih podatkih s tem sicer ne bomo veliko pridobili, za večje svetovne jezike pa se bo baza osnovnega znanja na ta način hitro povečala.

Drugi modul, ki ga pripravljamo, bo omogočal direktno vključitev naše tehnologije Piflar na zelo enostaven način. Uporabnik bo vpisal le dejstva v obliki stavka ali odstavka, sistem pa bo na osnovi njihove analize sam odgovarjal na vprašanja v zvezi z njimi. Če bomo npr. vpisali stavek »Besana je avtomatska lektorica, ki odkriva slovnične napake v slovenskih besedilih.«, bo sistem znal odgovoriti na vprašanja »Kaj je Besana?«.

»Kaj dela/odkriva Besana?« ipd. Enako velja tudi za angleški jezik. Take tehnologije za zdaj nima še noben primerljiv sistem v svetu. Testno je ta tehnologija že vključena kot del iskanja po spletnih (pod)stranah, kjer pa v praksi še ne pride do polnega izraza.

4.2. Vmesnik

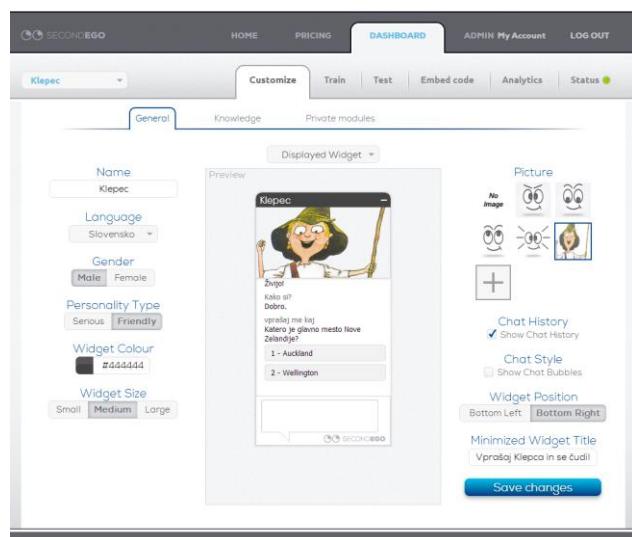
Ker problemi, ki jih sistem SecondEGO oz. z njim izdelani virtualni pomočniki rešujejo, niso omejeni le na slovenski prostor, je naš cilj tudi prodor na svetovni trg, zato je vmesnik portala v angleškem jeziku.

Uporaba portala je brezplačna, potrebno se je le registrirati. Plačljivost pride v poštev šele, ko dejansko število odgovorov nekega virtualnega pomočnika preseže določeno vrednost.

Po registraciji/prijavi ima uporabnik na voljo osnovne funkcionalnosti za upravljanje s pomočnikom.

4.2.1. Osnovne nastavitve

Ime virtualnega pomočnika je edini podatek, ki ga je potrebno vpisati. Vse ostale nastavitve si lahko uporabnik prilagodi glede na svoje cilje. Med te nastavitve sodijo jezik, spol, tip (uraden ali prijateljski), barva in velikost okenca, slika, prikaz zgodovine pogovora, oblika prikaza pogovora itd.



Slika6. Osnovne nastavitve virtualnega pomočnika

4.2.2. Osnovno znanje

Ko naredimo novega virtualnega pomočnika, ta seveda ne ve nič o specifičnih stvareh, na katere naj bi odgovarjal. Lahko pa mu za začetek dodamo nekaj splošnega znanja o stvareh, po katerih sogovorniki običajno povprašujejo. To znanje je trenutno zbrano v petih že vgrajenih modulih:

- »Osebnost« - modul vsebuje odgovore na najpogostejša vprašanja o osebnosti virtualnega pomočnika (»Kako ti je ime?«, »Ali rad bereš?«, »Ti si pameten.«, »Kaj najraje ješ?«, »Ali imaš kaj otrok?«);
- »Splošno znanje« - modul z vgrajenim splošnim znanjem (»Povej mi kakšno šalo.«, »Koliko je 3 + 2?«, »Sklanjaj samostalni miza.«, »Kdaj bo pust?«, »Koliko je ura?«);

- »Wikipedija« - informacije iz Wikipedije (»Kaj je voda?«, »Kdo je Pele?«);
- »Klepetanje« - pogovorne fraze (»Dober dan.«, »Dobro odgovarjaš.«, »Lažeš!«, »Povej kaj pametnega.«, »Ne spreminjaj teme.«);
- »Mašila« - splošni odgovori na neznana vprašanja. S pomočjo tega modula sogovornik sicer ne dobi konkretnega odgovora, vendar je odgovor tako oblikovan, da je vsaj delno v povezavi z vprašanjem. Na vprašanje, ki se začne s »Kako ...?« npr. odgovori »Kako? Ne bi vedel.« in podobno. Tak odgovor je vseeno boljši, kot da bi vedno in na vsa neznana vprašanja odgovarjal enako, npr. »Tega še ne vem.«.

Moduli so jezikovno odvisni in vsebujejo na tisoče vprašanj in odgovorov, ki jih v Amebisu redno nadgrajujemo. Module lahko uporabnik poljubno vklopja in izklopja, odvisno od potreb in namena virtualnega pomočnika.

4.2.3. Učenje

Za vnos splošnih odgovorov, ki jih v že vgrajenih modulih ni, ali za vnos specifičnega znanja o podjetju, zaposlenih, izdelkih in storitvah, mora seveda poskrbeti uporabnik sam. V ta namen ima na voljo vrsto možnosti, v razvoju pa so tudi že nekatere nove.

Prva in najenostavnejša možnost učenja je vpis pozdravnega sporočila, s katerim virtualni pomočnik pozdravi ali nagovori sogovornika. Teh je lahko tudi več in se lahko poljubno(krat) spreminjajo. Enako velja za vnos odgovora oz. odgovorov na neznano vprašanje.

Za hiter in enostaven vnos podatkov o nekaterih tipskih zadevah smo dodali »predloge« (»Templates«). Za podjetja, ki bodo pričakovano najštevilnejši uporabniki sistema SecondEGO, je mogoče vpisati osnovne podatke o podjetju, njegovih zaposlenih ter izdelkih in storitvah. Prednost predlog je v tem, da je potrebno vpisati le odgovore, pričakovana in možna vprašanja pa so že vgrajena. Glede na potrebe je mogoče izdelati poljubne predloge za poljubne jezike. Predloga »Podjetje« je za zdaj narejena za angleščino in slovenščino.

Za povsem specifične odgovore na specifična vprašanja je mogoče uporabiti t. i. vzorce. Vzorec »ujame« določena vprašanja ali ključne besede in ustrezno odgovori. Del odgovora je lahko tudi prikaz ali zamenjava določene spletne strani, kar dodatno obogati odgovor. Če je vzorec, ki »ujamejo« vprašanje več, se uporabi odgovor tistega, ki se najbolj ujema s postavljenim vprašanjem. S tem enostavnim mehanizmom lahko vnesemo odgovore na poljubno število vprašanj, pri čemer je vnos hiter in enostaven. Z ustrezno uporabo in povezavo vzorcev je mogoče ustvariti tudi dvosmerno komunikacijo s sogovornikom, kar je lahko zelo uporabno za izvajanje različnih anket ali vodenja sogovornika skozi določen postopek.

Za večjo odzivnost uporabnikov je možna uporaba »sprožilcev« (»Triggers«). Ker uporabniki na spletni strani dostikrat ne opazijo »skritega« virtualnega pomočnika, sprožilci poskrbijo, da se ta sam odpre ob določenih situacijah – po preteku določenega časa, po obisku določenega števila spletnih (pod)strani ali ob obisku točno določene (pod)strani. Sprožilcev je seveda lahko tudi več hkrati, uporabljajo pa se predvsem v marketinške namene (npr. opozarjanje na »akcije«).

Mnogo uporabnikov ima večino informacij, namenjenih svojim (potencialnim) strankam, že na svojih spletnih straneh. Problem nastane, če so te zelo obsežne, in jih stranke ne uspejo (dovolj hitro) najti. V ta namen smo dodali možnost pametnega iskanja po besedilu na spletnih (pod)straneh. Virtualni pomočnik lahko z njegovo pomočjo namesto s konkretnim odgovorom odgovori z naborom povezav na spletne strani, na katerih se najbolj verjetno nahaja odgovor. Na ta način uporabnik najhitreje – brez dodatnega učenja – stranki pomaga do iskanih informacij.

Zahtevnejšim in naprednejšim uporabnikom lahko po dogovoru omogočimo tudi vnos znanja s pomočjo skriptnega jezika, kar je nujno v primeru povezave virtualnega pomočnika z zunanjimi bazami podatkov.

4.2.4. Testiranje

V sistem je vključena možnost testiranja novo vgrajenega znanja, tako da lahko uporabnik vse potencialne napake odpravi še pred objavo.

4.2.5. Analitika

Za uspešno delovanje virtualnega pomočnika je zelo pomembno, da uporabnik sistema sproti preverja, kaj ga ljudje sprašujejo in kako na vprašanja odgovarja. Še posebno na začetku, v prvih tednih in mesecih, da uporabnik pomočnika čim prej nauči odgovorov na najpogostejša vprašanja.

Analitika mu omogoča, da spremlja trenutno, dnevno in mesečno aktivnost (število uporabnikov, število pogovorov, število odgovorov), najpogosteje zastavljena vprašanja v določenem časovnem obdobju, najpogostejša neznanana vprašanja in odgovore, uporabo sprožilcev in uporabo vključenih modulov. V razvoju je tudi možnost vpogleda v posamezne pogovore in druge koristne informacije.

Na osnovi statistike vseh vprašanj lahko uporabnik dobi koristne informacije, kaj njegove stranke dejansko zanima. Teh informacij na noben drug način ne more pridobiti, so pa te informacije izredno koristne za izboljšanje ponudbe ali spletnih strani.

Posebno koristna je za uporabnike statistika najpogostejših vprašanj, na katera virtualni pomočnik ne zna odgovoriti, saj so osnova za njegovo učenje.

4.3. Vgradnja na spletno stran

Za uporabo virtualnega pomočnika SecondEGO uporabnik ne potrebuje ne strojne opreme, ne dodatne programske opreme, pač pa le nekaj minut dela. Edina stvar, ki jo mora narediti, je, da v krovno spletno stran doda nekaj vrstic JavaScript kode, ki jo zgenerira sistem SecondEGO.

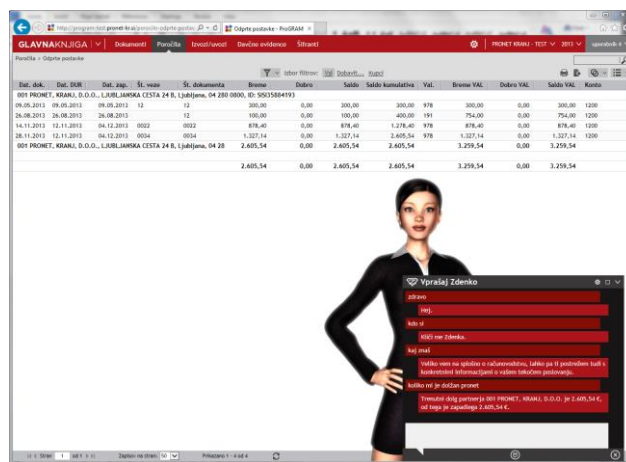
4.4. Vgradnja v druge aplikacije

Virtualnega pomočnika oz. funkcionalnost odgovarjanja na vprašanja v naravnem jeziku je mogoče s pomočjo posebnega vtičnika vgraditi tudi v druge aplikacije. S podjetjem Pronet Kranj smo npr. že izdelali poseben vtičnik za komunikacijo z računovodskimi programi, po naročilu pa je mogoče razviti tudi vtičnike za poljubna področja.

4.5. Delujoči pomočniki

Skupaj s sistemom SecondEGO smo razvijali tudi novo verzijo virtualnega pomočnika Klepca, ki deluje na spletnih straneh podjetja Amebis (www.amebis.si). Namenjen je klepetanju, pa tudi posredovanju informacij o podjetju in njegovih izdelkih.

Poleg Klepca deluje vedno več novih virtualnih pomočnikov SecondEGO. Med bolj uporabljanimi in zanimivimi je virtualna računovodkinja Zdenka (<http://program-test.pronet-kr.si/>). Vgrajena je neposredno v računovodsko spletno aplikacijo ProGRAM in odgovarja uporabnikom na vprašanja glede konkretnega poslovanja, uporabe programa in splošnih računovodskih zadev. Pri svojem delovanju uporablja podatke iz računovodske baze in zunanje baze besedil s področja računovodstva. Tako je uporabniku – kot v primeru voznega reda iz uvoda – prepuščen odločitev, ali do podatkov pride sam, ali za pomoč zaprosi Zdenko.



Slika 7. Virtualna računovodkinja Zdenka

5. Zaključek

Nekaj mesecev delujoč portal SecondEGO že vključuje veliko mehanizmov, s pomočjo katerih je mogoče izdelovati virtualne pomočnike, vendar samo vgrajene možnosti niso dovolj za izdelavo kakovostnih pomočnikov. Za to je s strani uporabnika potrebno vložiti dodatno delo, in več kot ga vloži, boljši je rezultat. Seveda pa lahko z vgradnjo izboljšanih mehanizmov pri tem pomaga tudi Amebis. Zato bomo z razvojem in izboljševanjem sistema glede na želje in zahteve uporabnikov nadaljevali tudi v prihodnje.

6. Literatura

- AIML - www.alicebot.org/aiml.html
- ChatScript Open Source project – <http://sourceforge.net/projects/chatscript/>
- Wallace, R. (2003). The Elements of AIML Style. A.L.I.C.E. Artificial Intelligence Foundation, Inc.
- Wallace, R. (2005) A.L.I.C.E - Artificial Intelligence Foundation <http://www.alicebot.org>
- Weizenbaum, Joseph. "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine," Communications of the Association for Computing Machinery 9 (1966): 36-45.