Piflar: sistem za učenje in odgovarjanje na vprašanja v naravnem jeziku
Crammer: the system for natural language learning and question answering

Peter Holozan
Amebis, d. o. o., Kamnik
peter.holozan@amebis.si

Povzetek

Odgovarjanje na vprašanja v naravnem jeziku je pomemben del jezikovnih tehnologij, ki za slovenščino še ni bil dovolj razvit. Zato je bil razvit sistem Piflar, ki se s pomočjo vmesnega jezika in modulov strojnega prevajalnika Presis nauči posamične povedi in zna potem odgovarjati na vprašanja o tem stavku. Odgovori so lahko kratki (le samo dejstvo) ali dolgi (v stavku). Sistem je jezikovno neodvisen (če so razviti ustrezni moduli za analizo in generacijo besedila, za zdaj so podprte slovenščina, angleščina in nemščina), tako da se lahko uporablja tudi pri medjezikovnem informacijskem poizvedovanju.

Ključne besede: *odgovarjanje na vprašanja, tvorba besedila, vmesni jezik, medjezikovno informacijsko poizvedovanje, strojno prevajanje*

Abstract

Natural language question answering (QA) is an important part of language technologies, but unfortunately still underdeveloped for Slovenian, thus the system Crammer was developed. Crammer uses the Interlingua and modules of Presis machine translator to learn individual sentences and answer questions about those sentences. The answers can be short (just facts) or long (used in sentence). The system does not depend on the language (if modules for analysis and generation of text, for now Slovenian, English and German are developed) and can be used in cross-language information retrieval (CLIR).

Key words: *question answering (QA), text generation, Interlingua, cross-language information retrieval (CLIR), machine translation*

Uvod

Spraševanje v naravnem jeziku (torej v stavkih in ne le po ključnih besedah) je pomemben del komunikacije med uporabnikom in računalnikom in smiselno je, da se računalniki prilagodijo človeškemu načinu spraševanja in iskanja podatkov in ne obratno.

Na področju odgovarjanja na vprašanja (ang.: question answering oz. QA) je predvsem za angleščino že veliko narejenega (Ledeneva in Sidorov, 2010). Za slovenščino je bil narejen sistem (Čeh et al., 2009), ki pa temelji predvsem na klasifikaciji vprašanj in na izbiranje najbližjega vprašanja v seznamu ročno pripravljenih vprašanj, na zaostanek slovenščine na tem področju pa opozarja tudi (Krek, 2012): »Večjih projektov, ki bi se ukvarjali z

odgovarjanjem na vprašanja in luščenjem informacij, za slovenščino ni, prav tako ne jezikovnih virov, potrebnih za izdelavo tovrstnih aplikacij.«

Zasnova sistema Piflar

Osnovna ideja sistema Piflar je uporabiti vmesni jezik, ki ga sicer uporablja strojni prevajalnik, in izkoristi njegove analizatorje (ki prevajajo naravni jezik v vmesni jezik) in generatorje (ki prevajajo vmesni jezik v naravni jezik). V ta namen je bil uporabljen strojni prevajalnik Presis (presis.amebis.si), ki ima za zdaj narejene analizatorje za slovenščino, angleščino in nemščino (testno tudi francoščino) in generatorja za slovenščino in angleščino. S tem ko uporabimo kot osnovo vmesni jezik, preložimo delo z naravnim jezikom na že napisane module strojnega prevajalnika, kar pomeni, da nam tega ni treba razvijati na novo.

Sistem je zamišljen tako, da se nauči poved in zna potem odgovarjati na vprašanja v zvezi njo. Za primer vzemimo, da je poved:

Jure včeraj ni prišel, ker je Tine videl sliko, ki jo je naslikal Grohar.

Tabela 1 prikazuje vprašanja, na katera želimo odgovoriti iz te povedi:

Tabela 1: Možna vprašanja za vzorčno poved

Kdo včeraj ni prišel?
Kdo ni prišel?
Zakaj Jure ni prišel?
Kaj je videl Tine?
Kdo je videl sliko, ki jo je naslikal Grohar?
Kdo je videl sliko?
Ali je Tine videl sliko?
Kaj je naslikal Grohar?
Ali je Grohar naslikal sliko?
Ali je Grohar naslikal sliko, ki jo je videl Tine.

Prvi prototip odgovarjanja na vprašanja na tak način je bil razvit za projekt Uvid (Holozan, 2010), vendar je imel precejšnje omejitve: omejen je bil le na enostavčne povedi, znanje ni bilo prenosljivo med verzijami baze (ker so bili pomeni zapisani z oznakami, ki se spreminjajo v vsaki novi verziji baze), težave so bile zaradi dvoumnosti vprašanj (ker vprašanja vsebujejo manj informacij kot vhodne povedi, jih je teže razdvoumljati, tako pri vprašanju »Kaj je pekel?« ni tako jasen pomen, kot pri stavkih »Miha je pekel piškote.« in »Pekel je kraj, kamor gredo grešniki po smrti.«).

Ta prototip je bil zdaj nadgrajen v sistem Piflar, glavne dopolnitve pa so bile:

- Analizator je bil nadgrajen in zdaj podpira večstavčne povedi, stavki (odvisniki) lahko nastopajo v vlogi stavčnih členov oz. njihovih delov (prilastkov).
- Pomeni so zdaj namesto s spremenljivimi številčnimi oznakami zapisani z bolj stalnimi imeni pomenov, tako da so podatki prenosljivi med verzijami, dodatna prednost pa je prenosljivost med jeziki, torej to, da se Piflar lahko nauči v enem jeziku in odgovarja v drugem.
- Pri vprašanjih upošteva več analiz in jih po vrsti preizkuša, dokler ne najde odgovora, s tem reši problem dvoumnosti vprašanj (čeprav seveda ne more vedeti, da je to točno tisti pomen, ki ga je imel v mislih uporabnik, ko je zastavil vprašanje, je večja verjetnost, da je pravi pomen tisti, katerega odgovor je naučen v bazi.

Kratek opis Amebisovega vmesnega jezika

Na tem mestu bo Amebisov vmesni jezik opisan le na kratko, precej bolj podroben opis je v Prilogi 6.2 v (Holozan, 2011).

Amebisov stavčni analizator vrne rezultat, zapisan v vmesnem jeziku. Cilj vmesnega jezika je, da strojno berljivo in nedvoumno opiše pomen povedi in dodatno opiše razčlembo stavkov na stavčne člene (stavčno analizo) ter besedno analizo (Holozan, 2011). Vse podatke o besedah, njihovih oblikah, pomenih, povezavah pomenov ipd. dobi analizator iz Amebisove jezikovne baze Ases (Arhar in Holozan, 2009).

Poved »Jure včeraj ni prišel, ker je Tine videl sliko.« tako analizator prevede v takle vmesni jezik:

```
*(-POV:(-STAg-npppdvn-------:(1OSB:(-SFR:(-DSF:(-JED:(-
SAMe:{38d8f9;181c9ef}[0]<26c>))))),(-PDOc:(-PRF:(-
PRSo:{7ec039;34c2a5e}[1]<b40>))),(*PVD:(-GPO:[2])),(0PVD:(-
GGL:{638f3c;2940aca}[3]<2030>)),(-PDOv:(-ODVd:((-STAdvnpppdvt-------:(-
LOCzv:[4]),(-VEZodv:{3a6c3b;1946c26}[5]<49d4>),(*PVD:(-GPO:[6])),(1OSB:(-SFR:(-
DSF:(-JED:(-SAMe:{76913d;310176f}[7]<26c>))))),(0PVD:(-
GGL:{7c3054;cd6a}[8]<42f4>)),(2PR4:(-SFR:(-DSF:(-JED:(-
SAMe:{18c4a;2a09724,1b52605,16066dd,175a68}[9]<50>))))),(-LOCkp:[10])))))))
```

Tabela 2 prikazuje pomene oznak elementov, ki se pojavljajo v zgornjem primeru.

Tabela 2: Oznake elementov vmesnega jezika

oznaka	pomen
POV	povedek
STA	stavek
OSB	osebek
SFR	samostalniška fraza
DSF	del samostalniške fraze
JED	jedro samostalniške fraze
SAM	samostalnik
PDO	prislovno določilo
PRF	prislovna fraza
PRS	prislov
PVD	povedek
GPO	pomožni glagol
GGL	glavni glagol
ODV	odvisnik
LOC	ločilo
VEZ	veznik
PR4	predmet v tožilniku

Vsak element se začne z oklepajem, ki mu sledi en znak. Ta je lahko številka, v tem primeru to pove, da gre za oznako dela v glagolski predlogi ali pa za samo glagolsko predlogo pri številki 0. Zvezdica pomeni, da gre za element, ki se ga da izpustiti (tipično pomožni glagoli), običajni elementi pa imajo -.

Temu potem sledu tričrkovna oznaka elementa, sledijo pa lahko parametri. Element SAM (samostalnik) ima tako npr. zadaj podatke o številu (e – ednina, d – dvojina, m – množina).

Sledi dvopičje, potem je pa med zavitimi oklepaji naprej oznaka leme, za podpičjem pa so naštete oznake možnih pomenov. Sledi zaporedna številka besede v originalnem besedilu med oglatimi oklepaji in na koncu še koda oblikoskladenjske oznake med lomljenimi oklepaji. Če pa je element sestavljen iz drugih podelementov, pa je namesto zavitih oklepajev seznam podelementov, ločenih z vejicami.

Primer zapisa znanja iz povedi v sistemu Piflar

Zapis znanja je sestavljen iz dveh konceptov: prvi je *dejstvo*, to je pravzaprav sam stavek, iz katerega smo se nekaj naučili, lahko so dodani še dodatni podatki (npr. spletna stran, kjer smo dobili ta stavek).

Drugi koncept so *delci*. Delci so v grobem stavčni členi stavka, razdeljeni so na povedek (ta vsebujemo pomen glagola oz. glagolske predloge (glagolska predloga vsebuje še informacije o glagolski vezljivosti, več o glagolskih predlogah v (Holozan, 2011, str. 47–50)) in dodatne podatke o glagolskem naklonu, času, dovršnosti in trdilnost), elementi glagolske predloge (osebek, predmeti, lahko tudi predložni deli in prislovna določila, ti deli ustrezajo elementom vmesnega jezika, ki so označeni s številkami), prislovna določila (po vrstah, npr. časovna, načinovna, krajevna) in členki (vezani na povedek, drugi so zajeti že v drugih delih). Pri modalnih glagolih je povedek sestavljen in obeh glagolov, elementi glagolske predloge pa imajo dodaten podatek, na kateri glagol se nanašajo. (Holozan, 2010)

Ker vmesni jezik vsebuje nekatere podatki, ki so za iskanje odveč oz. celo moteči (npr. podatki zaporedni števili besede v originalnem besedilu, oblikoskladenjske oznake in tudi same oznake lem, saj se pri iskanju zanašamo le na pomene, s čimer dosežemo, da najdemo tudi sopomenke), vmesni jezik najprej nekoliko predelamo (primer je za poved »Jure včeraj ni prišel, ker je Tine videl sliko.«) (imena pomenov so napisana ležeče za boljšo čitljivost):

*(-POV:(-STAg-npppdvn-------:(1OSB:(-SFR:(-DSF:(-JED:(-SAMe:{;*Jure (osebno ime (m)){0:1:0}*}[]<>))))),(-PDOc:(-PRF:(-PRSo:{;*včeraj{0:0:0}*}[]< >))),(*PVD:(-GPO:[])),(0PVD:(-GGL:{;*[priti] {NAM} {PDA}{0:0:0}*}[]<>)),(-PDOv:(-ODVd:((-STAdvnpppdvt-------:(-LOCzv:[4]),(-VEZodv:{;*ker{0:0:0}*}[]<>),(*PVD:(-GPO:[])),(1OSB:(-SFR:(-DSF:(-JED:(-SAMe:{;*Tine (osebno ime (m)){0:1:0}*}[]<>))))),(0PVD:(-GGL:{;*videti {PR4} {PDL}{0:0:0}*}[]<>)),(2PR4:(-SFR:(-DSF:(-JED:(-SAMe:{;*slika (splošno){0:0:0}*}[]<>))))),(-LOCkp:[])))))))

V naslednjem koraku razbijemo to analizo na štiri delce

- ppdn+[priti] {NAM} {PDA}{0:0:0}
- (-SFR:(-DSF:(-JED:(-SAMe:{Jure (osebno ime (m)){0:1:0}}[]<>))))
- (-PDOc:(-PRF:(-PRSo:{včeraj{0:0:0}}[]<>)))
- (-PDOv:(-ODVd:(-STAdvnpppdvt-------:(-LOCzv:[]),(-VEZodv:{ker{0:0:0}}[]<>),(*PVD:(-GPO:[])),(BOSB:(-SFR:(-DSF:(-JED:(-SAMe:{Tine (osebno ime (m)){0:1:0}}[]<>))))),(APVD:(-GGL:{videti {PR4} {PDL}{0:0:0}}[]<>)),(CPR4:(-SFR:(-DSF:(-JED:(-SAMe:{slika (splošno){0:0:0}}[]<>)))))))

Vsi ti delci se potem povežejo na ustrezno dejstvo (prek vmesne tabele dejstvo_delec), po potrebi se dodajo še poenostavljeni delci (če je v osebku npr. zveza pridevnika in samostalnika, se kot delec doda tudi samo samostalnik ipd.).

Pri spraševanju vprašanja spet enako prevedemo v vmesni jezik in razbijemo na delce. Za vprašanje »Zakaj Jure ni prišel?« dobimo tako naslednja delca:

- ppdn+[priti] {NAM} {PDA}{0:0:0}
- (-SFR:(-DSF:(-JED:(-SAMe:{Jure (osebno ime (m)){0:1:0}}[]<>))))

Dodatno pa dobimo še podatek, da je vprašalnica v elementu »(-PDOv:(-PRF:(-VPRro:{;*zakaj (v){0:0:0}*}[]<>)))«, iz česar razberemo, da iščemo prislovno določilo vzroka. V bazi nato poiščemo, ali vsebuje kakšno dejstvo, ki vsebuje oba delca iz vprašanja in ima še podatek o prislovnem določilu vzroka, in kot rezultat tako dobimo delec »(-PDOv:(-ODVd:(-STAdvnpppdvt-------:(-LOCzv:[]),(-VEZodv:{ker{0:0:0}}[]<>),(*PVD:(-GPO:[])),(BOSB:(-SFR:(-DSF:(-JED:(-SAMe:{Tine (osebno ime (m)){0:1:0}}[]<>))))),(APVD:(-GGL:{videti {PR4} {PDL}{0:0:0}}[]<>)),(CPR4:(-SFR:(-DSF:(-JED:(-SAMe:{slika (splošno){0:0:0}}[]<>)))))))«, kar zna generator prevesti v »ker je Tine videl sliko« (ali pa v »because Tine saw picture«, če uporabimo angleški generator).

Generiranje odgovorov

Piflar lahko vrne tri odgovore: originalno poved, iz katere se je naučil odgovor na zadano vprašanje, kratki odgovor, kjer odgovori le z besedo oz. frazo, in dolgi odgovor, kjer odgovori v stavku. Dodatno pa si lahko zapomni še spletno stran, kjer je bila originalna poved (po potrebi pa bi si lahko tudi druge podatke).

Dolgi odgovor, kjer je dejstvo zapisano v stavku, je najbolj naraven način odgovora, na tak način običajno odgovarjamo tudi ljudje. Pri tem je zelo pomembno členjenje po aktualnosti, ki vpliva na besedni red v stavku (v slovenščini je novo dejstvo tako tipično na koncu odgovora). Tako zgrajeni odgovori nam zvenijo bolj naravno in jih zato laže preberemo.

Za prevajalnik Presis sta razvita generatorja za slovenščino in angleščino, zato lahko Piflar vrača dolge odgovore le v teh dveh jezikih. Za nemščino je narejena poenostavljena verzija generatorja, tako da lahko za nemščino vrača le kratke odgovore, bi pa bilo po potrebi možno razviti tudi popoln generator za nemščino.

Primeri rezultatov

Piflar je bil med drugim preizkušen na slovenski Wikipediji. Tabela 3 prikazuje štiri povedi, iz katerih se je Piflarju uspelo naučiti, vprašanja, ki so bila vprašana o teh povedih, in kratke in dolge odgovore, ki jih je vrnil Piflar.

Tabela 3: Primeri odgovorov Piflarja

Vhodna poved	Aminokislinski ostanek je, kar ostane od aminokisline, ko se odcepi molekula vode pri nastanku peptidne vezi.	
vprašanje	kratki odgovor	dolgi odgovor
Ali se molekula vode odcepi pri nastanku peptidne vezi?	*da*	*Da.*
Kaj se odcepi pri nastanku peptidne vezi?	*molekula vode*	*Pri nastanku peptidne vezi se odcepi molekula vode.*
Kje se odcepi molekula vode?	*pri nastanku peptidne vezi*	*Molekula vode se odcepi pri nastanku peptidne vezi.*
Vhodna poved	Aminokisline so osnovni gradniki beljakovin.	
vprašanje	kratki odgovor	dolgi odgovor
Ali so aminokisline osnovni gradniki beljakovin?	*da*	*Da.*
Kaj so osnovni gradniki beljakovin?	*aminokisline*	*Osnovni gradniki beljakovin so aminokisline.*

Kaj so aminokisline?	*osnovni gradniki beljakovin*	*Aminokisline so osnovni gradniki beljakovin.*
Vhodna poved	Posttranslacijska modifikacija je često potrebna, da beljakovina, na primer encim, postane aktivna.	
vprašanje	kratki odgovor	dolgi odgovor
Čemu je potrebna posttranslacijska modifikacija?	*da postane beljakovina, na primer encim, aktivna*	*Posttranslacijska modifikacija je potrebna, da postane beljakovina, na primer encim, aktivna.*
Vhodna poved	Žled nastane, ko dežuje ali rosi pri temperaturah pod lediščem oziroma ko padavine v tekoči obliki padajo na podhlajeno podlago.	
vprašanje	kratki odgovor	dolgi odgovor
Kdaj pride do žleda?	*ko dežuje ali ko prši pri temperaturah pod lediščem*	*Do žleda pride, ko dežuje ali ko prši pri temperaturah pod lediščem.*

Pri dolgih odgovorih je lepo vidno spreminjanje besednega reda zaradi členjenja po aktualnosti. Pri zadnjem vprašanju se vidi tudi uporaba sopomenk, ko je v vhodni povedi uporabljen glagol »nastati«, v vprašanju pa »priti do«.

Zaključek

Kljub nekaterim omejitvam lahko sistem Piflar učinkovito odgovarja na vprašanja in lahko učinkovito nadomesti ročno pripravo odgovorov na množico potencialnih vprašanj. Je pa uspešnost njegovega odgovarjanja v veliki meri odvisna od besedil, iz katerih se uči – iz primerno napisanih vhodnih besedil se lahko nauči zelo veliko.

Sistem Piflar je za zdaj v uporabi v sistemu za virtualne asistente SecondEgo (www.secondego.com), kjer je uporabljen za odgovarjanje na splošna vprašanja in kot dopolnitev iskanja po spletnih straneh.

Trenutne omejitve in načrti za prihodnost

Sistem Piflar ima v tem trenutku še določene omejitve:

- Analizator še ne zna analizirati vseh povedi, pri korpusu Gigafida npr. uspešno analizira 51,4 % povedi. Vzrok je med drugim to, da analizator še ne podpira nekaterih stavčnih struktur, ki jih bo treba dograditi v analizator (npr. vrinjeni deli med oklepaji, ki za zdaj zmotijo analizo zunanjega stavka, naštevanje predložnih zvez, glagolniška vezljivost (»prodaja kupcem«) ipd.).
- Analizatorju delajo težave tudi neznane besede, Piflarju pa dodatno še besede, ki nimajo pripisanih pomenov. Vendar je že zdaj v korpusu Gigafida prepoznano 97,4 % besed, od teh pa jih le 2,8 % nima pripisanih pomenov (ta številka je relativno majhna, ker pomeni večinoma manjkajo pri redkih besedah). Če se omeji domena, na kateri bi radi uporabili Piflarja, je manjkajoče besede in pomene mogoče hitro dodati v bazo Ases, ki jo potem uporablja analizator. Pri manjkajočih pomenih pa bi si lahko pomagali tudi tako, da bi v primeru, ko beseda (lema) nima pripisanih pomenov, namesto pomena uporabili kar direktno lemo, izgubimo pa v tem primeru večjezičnost Piflarja.
- Težave pri razdvoumljanju, analizator kdaj narobe razume besedilo. Možna rešitev je omejitev domene delovanja, pri čemer lahko v Asesu dopolnimo povezave pomenov in s tem izboljšamo razdvoumljanje, druga rešitev pa je dovolj poenostavljeno in nedvoumno napisano vhodno besedilo, iz katerega se Piflar uči.

- Razreševanje sklicev osebnih in kazalnih zaimkov, za zdaj se Piflar nauči dejstvo z zaimkom, kar pa največkrat ni pravilno. Razreševanje sklicev je v splošnem sicer težak problem na področju računalniške obdelave naravnega jezika, na srečo pa za velik del osebnih zaimkov velja, da jih je precej preprosto razrešiti (npr. osebni zaimki v prilastkovih odvisnikih: »*Slika,* ki sem **jo** videl včeraj ...«). Druga rešitev je vhodno besedilo, napisano brez uporabe osebnih in kazalnih zaimkov.
- Težave zaradi različnih slovničnih struktur, ki imajo sicer isti pomen (»včeraj poslano pismo«, »pismo, poslano včeraj«, »pismo, ki je bilo poslano včeraj«), to pride še posebej do izraza pri uporabi v različnih jezikih. Rešitev je, da se za iskanje doda normalizacija vmesnega jezika, da bodo deli z istim pomenom zapisani na enak način ne glede na to, kako so bili zapisani v vhodnem besedilu oz. v vprašanju.
- Sklepanje iz več dejstev. Za zdaj Piflar odgovarjale na nivoju posamične povedi. Vendar bi se to dalo po potrebi dopolniti s pravili sklepanja, s katerimi bi lahko povezoval dejstva iz različnih povedi (npr. vhodni povedi sta »Janez bo prišel ob sončnem vzhodu.« in »Sonce vzide ob sončnem vzhodu.«, vprašanje pa je »Kdo bo prišel, ko bo vzšlo sonce?«).

Zgornje težave kažejo na to, da je pri razvoju Piflarja še veliko odprtih poti za izboljšave, kljub temu pa je sistem Piflar uporaben že na sedanji stopnji razvoja.

Literatura

[1] Arhar, Š., Holozan, P. (2009): »ASES – leksikalna podatkovna zbirka za razvoj slovenskih jezikovnih tehnologij«. V Mikolič V. (ur.): Jezikovni korpusi v medkulturni komunikaciji. Koper: Založba Annales.

[2] Čeh, I., Zorman, M., Ojsteršek, M. (2009): »Klasifikacija vprašanj, strojno učenje in sistemi za odgovarjanje na vprašanja zastavljena v naravnem jeziku«. V zborniku Osemnajste mednarodne elektrotehniške in računalniške konference - ERK 2009 (zv. B, str. 121–124). Portorož.

[3] Holozan, P. (2010): »Prenova sistema dialoga Kolos za projekt UVID«. V zborniku Sedme konference JEZIKOVNE TEHNOLOGIJE. Ljubljana: Institut »Jožef Stefan«.

[4] Holozan, P. (2011): »Samodejno izdelovanje besedilnih logičnih nalog v slovenščini« (str. 51–67). Magistrsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko.

[5] Krek, S. (2012): »Slovenski jezik v digitalni dobi = The Slovene language in the digital age« (str. 27). Ur. Hans Rehm in Georg Uszkoreit. Heidelberg: Springer.

[6] Ledeneva, Y., Sidorov, G. (2010): »Recent Advances in Computational Linguistics«. Informatica, 34:3–18. Ljubljana.

Kratka predstavitev avtorja

Mag. Peter Holozan je razvijalec v podjetju Amebis, d. o. o., Kamnik in raziskovalec v Amebisovem razvojnem centru. Magistriral je na Fakulteti za računalništvo in informatiko v Ljubljani in je doktorski študent na Filozofski fakulteti v Ljubljani (slovenistika). Ukvarja se predvsem z jezikovnimi tehnologijami za slovenščino, med drugim s črkovalniki, slovničnim pregledovalnikom, strojnim prevajanjem, oblikoskladenjskim označevanjem, korpusi (Fida, FidaPLUS), slovarji (ASP32).