

Jasminka Tomljanović¹

Marina Krsnik²

Mile Pavlič³

Stručni rad

UDK 007

SUSTAVI PITANJA I ODGOVORA⁴

SAŽETAK

Sustavi pitanja i odgovora su način pretraživanja informacija koji odgovara na pitanje postavljeno prirodnim jezikom. Sastoje se od triju glavnih komponenata: klasifikacija pitanja, pronalaženja informacija i izvlačenja odgovora. Uvođenjem velikih zajedničkih procjena, kao što je TREC konferencija, stvorene su snažne interesne zajednice i ubrzan napredak u istraživanju sustava pitanja i odgovora. U radu su prikazane domene problema odgovaranja na pitanja. Prikazan je povijesni razvoj sustava pitanja i odgovora od jednostavnijih, koji su bili usko specijalizirani, do današnjih, puno složenijih i kvalitetnijih sustava koji su sposobni dati nam kratke i sažete odgovore na pitanja iz raznih domena. Analizirani su sustavi prema namjeni. Dat je pregled dosadašnjih istraživanja o sustavima pitanja i odgovora. Definirana je opća arhitektura sustava pitanja i odgovora te analizirani aktualni pristupi u svakoj fazi arhitekture. Sustavi pitanja i odgovora još uvijek su u fazi razvoja i eksperimentiranja. Zaključeno je kako još uvijek najveći problem sustava pitanja i odgovora predstavlja odabir najtočnijeg i najsazetijeg odgovora s kojim će korisnik biti zadovoljan.

Ključne riječi: sustavi pitanja i odgovora, TREC konferencija, arhitektura sustava pitanja i odgovora

1. UVOD

Sustavi pitanja i odgovora ili tzv. *question – answering systems* (QAs) pronalaze točne i sažete odgovore na pitanja postavljena govornim jezikom (Hirschman et al., 2001). Izravno se odnose na pretraživanje informacija i obradu prirodnog jezika u okviru izgradnje sustava za automatsko odgovaranje na pitanja postavljena prirodnim jezikom. Za pronalaženje odgovora mogu koristiti strukturirane baze podataka, zbirke dokumenata na lokalnoj razini, interne dokumente poduzeća, mreže novinskih izvješća i internet. Glavni cilj jamstva kvalitete sustava je dati točne odgovore uz pomoć vlastite baze znanja. Istraživanja ovog područja bave se tipovima pitanja, uključujući činjenice, popise, definicije, semantička ograničenja i jezično neovisna pitanja.

¹ Mr. sc., viši predavač, Veleučilište u Rijeci, Vukovarska 58, Rijeka, Hrvatska. E-mail: jasminka.tomljanovic@veleri.hr

² Student, Odjel za informatiku, Sveučilište u Rijeci, Radmile Matejčić 2, Rijeka, Hrvatska.
E-mail: marina.krsnik@inf.uniri.hr

³ Dr. sc., redoviti profesor, Odjel za informatiku, Sveučilište u Rijeci, Radmile Matejčić 2, Rijeka, Hrvatska.
E-mail: mile.pavlic@ris.hr

⁴ Datum primitka rada: 16. 2. 2014.; datum prihvatanja rada: 5. 5. 2014.

Prvi QA sustavi razvijeni u 1960-ima su sustavi sa sučeljima na prirodnom jeziku za inteligentne sustave sa specifičnom domenom. Razvoj interneta obnovio je potrebu za istraživačkim tehnikama čiji je cilj smanjenje preopterećenosti informacijama. Tražilice vraćaju članke s fragmentima teksta svih dokumenata koji možda sadrže odgovor. Razvoj osiguranja kvalitete sustava nastaje kao pokušaj rješavanja preopterećenja s informacijama.

Zadaća ovog rada je analizirati razvoj QA sustava, a cilj je utvrditi njegov najveći problem.

Ostatak rada strukturiran je na sljedeći način. U točki 2. definiran je pojam QA sustava, dimenzije problema njihovog vrednovanja i metodologije konstruiranja odgovora. U točki 3. prikazan je povijesni razvoj QA sustava. Točka 4. daje pregled dosadašnjih istraživanja QA sustava. Prikaz generičke arhitekture QA sustava i analiziranje aktualnih pristupa u svakoj fazi je u točki 5. Rad završava zaključkom u točki 6.

2. POJAM QA SUSTAVA, DIMENZIJE PROBLEMA VREDNOVANJA QA SUSTAVA I METODOLOGIJA

Razvoj informacijskih tehnologija potiče potrebu korisnika za sustavima koji odgovaraju na pitanja. Raste potreba za sustavima koji omogućuju postavljanje pitanja koristeći svakodnevni govor ili tzv. *Natural Language (NL)* i dobivanje odgovora brzo i jezgrovito s najmanjom količinom konteksta. Današnji pretraživači mogu ponuditi liste dokumenata koji sadrže potencijalne odgovore, ali ne dostavljaju odgovore korisniku.

Rješenje tog problema su upravo QA sustavi. Oni su način pretraživanja informacija koji odgovara na pitanje postavljeno govornim jezikom. Pitanja mogu biti jednostavna poput činjenica, definicija i nabiranja (*Koji je najviši vrh svijeta?*), ali isto tako mogu biti složena (*Kolika je učinkovitost antibiotika kod ljudi oboljelih od gnojne angine?*).

Uspjesi u ovom području postignuti su kao dio *Text Retrieval Conference (TREC)*, tj. niza radionica čiji je zadatak prikupiti što više informacija iz različitih područja istraživanja te poduprijeti i potaknuti istraživanja na području QA sustava.

Za odgovor na pitanje sustav prvo mora analizirati problem koristeći kontekst. Pronalazi jedan ili više mogućih odgovora koje internetski izvor nudi i odgovor predstavlja u nekom odgovarajućem obliku. Najčešće je to popratni materijal koji korisniku odgovor čini razumljivim ili ga dodatno objašnjava.

Počevši 1999., TREC je započeo s ocjenjivanjem sustava koji odgovaraju na činjenična pitanja konzultirajući se s dokumentima TREC zbirke. Velik broj sustava uspješno je kombinirao pretraživanje informacija i obradu tehnike prirodnog jezika ili tzv. *Natural Language Processing (NLP)*.

Drugačiji pristup odgovaranju na pitanja omogućili su testovi za ocjenjivanje čitanja s razumijevanjem. Temelji se na sposobnosti sustava da odgovori na pitanje o određenom odlomku. Ti su testovi danas popularni za ocjenjivanje razumijevanja teksta učenika i

studenata, a kao rezultat daju osnovu za uspoređivanje performansi čovjeka i performansi sustava.

Konferencije (Seattle 2000), radionice i ocjene sustava prikazuju sve domene problema odgovaranja na pitanja. Dimenzije ovog problema mogu se promatrati s obzirom na: aplikacije, korisnike, vrste pitanja, vrste odgovora, ocjenjivanje i prezentiranje.

QA sustavi imaju brojne *aplikacije* koje možemo podijeliti s obzirom na izvor koji sadrži odgovore. To mogu biti strukturirane baze podataka, polustrukturirane baze (npr. polja za komentare u bazi) ili slobodan tekst.

QA *aplikacije* mogu se razlikovati i prema pretraživanju. Može biti pretraživanje nepromjenjivog skupa zbirke (koristi TREC za ocjenjivanje), pretraživanje *weba*, pretraživanje zbirke ili knjige (npr. enciklopedije) (Kupiec, 1993) ili pak pretraživanje samostalnog teksta, kao kod ocjenjivanja čitanja s razumijevanjem.

Te sustave možemo klasificirati kao neovisne i specijalizirane, s obzirom na njihovu domenu. Generalno gledajući, zbirke postaju sve veće i raznolikije i za očekivati je da pronalaženje odgovora postaje sve teže u takvim zbirkama. Istražitelji ovog područja smatraju kako je veća vjerojatnost da ćemo pronaći odgovor ako imamo više izvora iz kojih tražimo odgovore (Mann et al., 2001).

Korisnici mogu varirati od onih koji sustav koriste po prvi put do onih koji se sustavom koriste svakim danom. Različite skupine korisnika zahtijevaju različita sučelja, postavljaju različita pitanja i žele različite vrste odgovora. Za one koji se sustavom koriste prvi put važno je objasniti ograničenja sustava, kako bi znali kako interpretirati odgovore koje su dobili. Za stručne, ekspertne korisnike, poželjno je razviti i ažurirati model korisnika koji će korisnicima od velike količine informacija napraviti sažetak te im dostaviti točno one podatke koji ih zanimaju.

Još uvijek stručnjaci ne znaju kako predvidjeti što čini jedno pitanje težim od drugog. To je problem koji je osobito važan za testiranje obrazovne zajednice, gdje ispitivači moraju pripremiti i provjeriti standardizirane testove.

Pitanja možemo klasificirati na temelju tipa odgovora koji zahtijevaju. Postoje činjenična pitanja, subjektivna pitanja i sažetci.

Druga podjela pitanja odnosi se na da/ne pitanja, pitanja s upitnim riječima kao što su gdje, tko, što, kako, kada, koliko, zašto (*Tko je predsjednik Hrvatske? Koliko je težak slon?*), indirektni zahtjevi (*Htio bih posjetiti...*) i naredbe (*Nabroji sve hrvatske predsjednike*). Sve ih tretiramo kao pitanja, iako neka od njih prema formi zapravo to i nisu.

Neki sustavi ovise o pitanjima s upitnim riječima za traženje odgovora. Ako pitanje započinje upitnom riječi *tko*, sustav prepoznaje da odgovor zahtijeva osobu. Zapčinje li upitnom riječi *kada*, prepoznaje kako odgovor mora biti neko vrijeme. Takvi sustavi nerijetko nailaze na problem ako korisnik oblikuje pitanje kao naredbu (*Navedi ime prvog hrvatskog predsjednika*) (Pantel et al., 2012).

Postoje dokazi da su neke vrste pitanja teže od drugih. Npr. pitanja *kako* i *zašto* obično su teža jer zahtijevaju razumijevanje uzročno-posljedičnih veza i uglavnom se nalaze kao dijelovi rečenice ili čak kao samostalne rečenice (Hirschman, 1999).

Ako sustav dobro analizira vrstu odgovora koje očekuje, smanjit će količinu mogućih odgovora (Mann et al., 2001). Na određene vrste pitanja teže je odgovoriti ako vrsta odgovora nije dovoljno dobro definirana. Npr. pitanja koja započinju upitnom riječi *što* teška su jer daju vrlo malo ograničenje na vrstu odgovora (*Što se dogodilo? Što si vidio?*).

Odgovori mogu biti kratki, dugi ili čak popisi ili priče. Mijenjaju se ovisno o namjeni korištenja i namjeni korisnika. Ako korisnik želi objašnjenje odgovor će biti duži. Taj tip pitanja odmah implicira dug odgovor. Sukladno tome, ako korisnik postavi činjenično pitanje koje se odnosi na razumijevanje teksta (npr. *Tko je glavni lik u priči?*) odgovor će biti kratak, uglavnom jedna, dvije riječi ili kratka fraza.

Postoje različite metodologije koje konstruiraju odgovor. Jedna od njih je uz pomoć izlučivanja; rezanje i lijepljenje isječaka iz izvornog dokumenta koji sadrže odgovor. Druga metodologija odvija se putem generiranja. Tu se odgovor kombinira iz više rečenica ili više dokumenata. Međusobna povezanost izdvojenih dijelova odgovora može biti vrlo niska, što onda zahtijeva generiranje kako bi se ti dijelovi spojili u smislenu cjelinu.

Što čini odgovor dobrim? Je li dobar odgovor dug odgovor koji sadrži popratni tekst koji ga objašnjava? Kontekst je koristan ako sustav nudi više mogućih odgovora. Omogućava korisniku odabir najboljeg odgovora koji ne mora biti prvi ponuđen niti najbolje ocijenjen. U drugim slučajevima, kratki odgovori mogu biti puno bolji. Iskustva u *ocjenjivanju* odgovaranja na pitanja koja TREC iznosi pokazuju kako je lakše navesti duže dijelove koji sadrže odgovor u sebi nego samo navesti kratke segmente.

U stvarnim situacijama kada tražimo određene informacije postoji korisnik koji komunicira sa sustavom u stvarnom vremenu. Korisnik obično započinje općenitim pitanjem i sustav mu izbacuje odgovor direktno ili indirektno, vraćajući mu velik broj dokumenata. Korisnik tada sužava potragu, gdje se bavi nekom vrstom dijaloga sa sustavom. Olakšavanje takve komunikacije vjerojatno bi povećalo uporabljivost i zadovoljstvo korisnika. Ako sučelja imaju mogućnost za obradu govora i dijaloga, sustavi mogu koristiti govor kako bi pristupili određenim informacijama na internetu.

Do danas je malo toga napravljeno na sučeljima za odgovaranje na pitanja. Riječ je o malobrojnim sustavnim procjenama o tome kako najbolje *prezentirati* informaciju korisniku; koliko odgovora i koliko konteksta ponuditi te je li bolje ponuditi kompletan odgovor ili samo kratak odgovor s priloženim sažetkom ili naputkom.

3. POVIJESNI RAZVOJ QA SUSTAVA

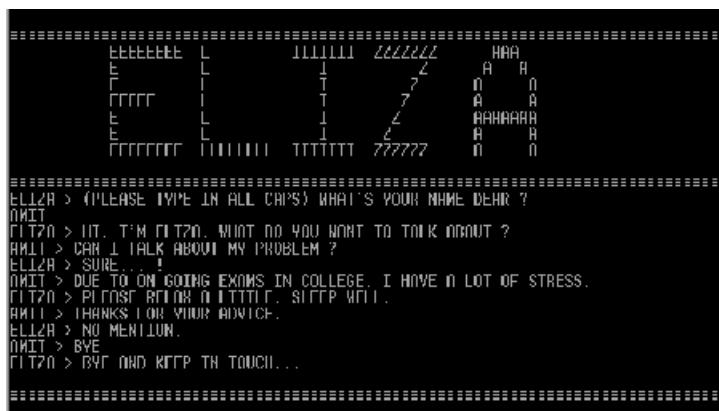
U razdoblju od 1960. do 1965. razvoj QA sustava započinje i snažno se razvija (Simmons, 1965). Sustavi tog doba odgovaraju na razgovorna pitanja, pristupaju spremištu podataka te pokušavaju odgovoriti na pitanja iz tekstualnih izvora (enciklopedija).

Prvi najpoznatiji QA sustav je *Baseball* (Green et al., 1961). Odgovarao je na pitanja o utakmicama američke bejzbolske lige. Analizirao je pitanje koristeći lingvističko znanje u kanonskom obliku (korišteni za generiranje upita). Dobivenim upitom tražio je odgovor u bazi podataka o bejzbolu. S obzirom na način na koji se nosio sa sintaksom i semantikom pitanja imao je ograničenu domenu – odgovarao je samo na pitanja o bejzbolu. Bio je namijenjen prvenstveno kao sučelje strukturiranoj bazi podataka, a ne velikom skupu teksta i bio prvi od niza programa dizajniranih za pristup bazama koje sadržavaju tekst pisan u prirodnom jeziku. Cilj je bio omogućiti korisnicima komunikaciju sa sučeljem na njihovom vlastitom jeziku. Tako bi, umjesto korisnika, sučelje bilo zaduženo za pitanja, strukturu baze i sam prijevod.

Drugi je *Lunar*. Dizajniran je geolozima za postavljanje pitanja o mjesječevim stijenama i sastavu tla (Woods, 1973). Odgovarao je na pitanja tipa *Koja je prosječna koncentracija aluminija u stijenama?*. Uspješno je odgovorio na 90 % pitanja geologa koja su bila unutar njegove domene (Voorhees, Tice, 2000). Ograničen je na usko područje.

Zajednička karakteristika ovih sustava bila je baza podataka ili sustav znanja pisan ručno, a pisali su ih stručnjaci određenog područja. Koristili su tehnike slične tehnikama korištenim kod *Elize* i *Doctora*, prvih *chatterbot*⁵ programa.

Slika 1. *Eliza*, chatterbot program



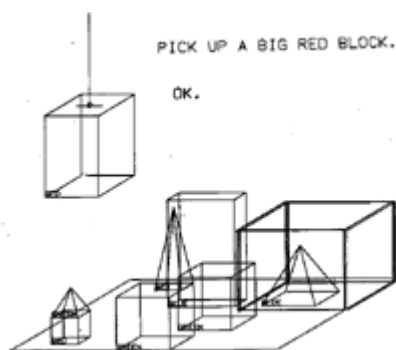
Izvor: https://www.google.hr/search?q=eliza+program&source=lnms&tbm=isch&sa=X&ei=xS7_UvnqCZTW4AS78oDwBA&ved=0CAcQ_AUoAQ&biw=1366&bih=572

Sljedeći je *SHRDLU*⁶ (Winograd, 1972). Stimulirao je rad robota u svijetu igračaka i nudio je mogućnost postavljanja pitanja o stanju svijeta. Imao je ograničenu domenu i vrlo jednostavan svijet objašnjen kroz pravila fizike prenošena u računalni program.

⁵ Program koji razgovara s korisnicima.

⁶ Izvedeno od *etaoin shrldu*; odnosi se na redoslijed 12 najčešće korištenih slova engleskog jezika koja su se nalazila u prva dva stupca pisaćih strojeva.

Slika 2. Demonstracija programa SHRDLU



Izvor: <http://hci.stanford.edu/~winograd/shrdlu/>

Razvoj specijaliziranih baza znanja potiče QA sustave na razvoj sučelja ekspertnih sustava. Vrlo su slični današnjim modernim QA sustavima, s razlikom u arhitekturi. Oslanjaju se na stručno izgrađene i organizirane baze znanja, a moderni QA sustavi na statističke obrade velikih, nestrukturiranih tekstova prirodnog jezika.

Razvoj računalne lingvistike⁷ donosi ambiciozne projekte na području tekstualnog razumijevanja i odgovaranja na pitanja, npr. *Unix Consultant*; spec. baza pod. *Unix OS* (Wilensky et al., 1984). Drugi je *LILOG*, sustav razumije tekst, radi s domenom turističkih informacija o njemačkom gradu. Nikad nisu dospjeli dalje od faze demonstracije, ali su pridonijeli razvoju teorija računalne lingvistike i zaključivanja.

3.1 Dijalog interaktivnih savjetodavnih sustava

Alan Turing predložio je testiranje inteligencije stroja kroz test razumijevanja razgovora (Turing, 1950). Test je zamislio s ispitivačem koji postavlja pitanja subjektu koji ne vidi, može biti osoba ili stroj. Ispitivač po odgovoru određuje je li odgovorio stroj ili osoba.

SHRDLU i *GUS* (Bobrow, 1977) razvijeni su za istraživanje razumijevanja problema u modeliranju ljudskog dijaloga. *SHRDLU* je stimulirao rad robota za pomicanje objekta, a *GUS* stimulirao savjetnika za putovanja. Jednostavni dijalozi bili su izazov i problemi rješavanja razvoja interaktivnog savjetodavnog sustava.

Rani QA interaktivni sustavi koriste strukturiranu bazu znanja koja im nije bila potrebna. Umjesto toga, bolje je koristiti zbirke tekstova, ali je važno naglasiti kako je odgovor u stvarnom vremenu neophodan za takve sustave. Npr. MIT-ev *Jupiter* sustav pruža telefonskim dijaloškim sučeljem informacije o vremenu. Sakuplja informacije o vremenu s različitih stranica na internetu. Ističe pragmatične probleme i probleme korisnika vidljive u izražavanju složenih zahtjeva i dobivanju točnih odgovora iz kompleksnog skupa podataka.

⁷ Znanstvena disciplina koja se bavi razvojem kompjutorskih programa koji omogućavaju računalnu obradu prirodnog jezika.

3.2 Odgovaranje na pitanja i razumijevanje teksta

Najjednostavnija provjera razumijevanja teksta je postavljanje pitanja o tekstu. Odgovori li, tekst je shvatio, ne odgovori li, nije ga shvatio. Tehnika se već dugo koristi kod određivanja razine razumijevanja u učenju stranih jezika i otpočetak je prihvaćena kao prikladan način za testiranje sposobnosti sustava u razumijevanju prirodnog jezika.

U sustavu *QUALM* (Lehnert, 1977) ključna ideja je odmaknuti se od pristupa koji odgovaranje na pitanja shvaća kao ulaz u skroz odvojenu bazu ili proces vraćanja podataka, već kao proces koji odgovara i razumije pitanja. Razumijevanje i odgovaranje se oslanja na kontekst priče i pragmatični odabir prikladnog odgovora. Pitanje i tekst rastavljeni su s obzirom na konceptualnu ovisnost reprezentacije. Odgovaranje na pitanja nije samo proces usklađivanja tih reprezentacija. Interpretacija pitanja mora pripadati određenoj konceptualnoj kategoriji kao što su potvrda, zahtjev, uzročni prethodnik itd.

Takva klasifikacija pitanja nužna je za izbjegavanje odgovaranja na pitanja tipa *Znaš li koliko je sati?* s *Da*, ili *Kako je Marina napisala ispit?* s *Olovkom*. Daljnji zaključak treba biti donesen na temelju konteksta. Odgovaranja na pitanje, očekivanja koja priče mogu izazvati dok su pričane treba preoblikovati. Ako pročitamo *Ivan je naručio hamburger*, pretpostavlja se da ga je i pojeo. Ako se priča nastavi i dobijemo informaciju da je hamburger zagorio, pretpostavka više ne vrijedi. No ako priča ne sadrži činjenicu da hamburger nije pojeđen i postavimo pitanje *Zašto Ivan nije pojeo hamburger?* nedavanje odgovora nije dovoljno dobro i nije nešto što bi čovjek napravio. Jedini ispravan odgovor bio bi *Zato što je hamburger zagorio*. Kako bismo došli do tog odgovora potrebno je preoblikovati očekivanja odgovora i odrediti što je u tekstu utjecalo na promjenu. Najvažnije je razumijevanje kao dinamički proces koji uključuje povezivanje svjetskog znanja i informacija doslovno preuzetih iz teksta.

Slika 3. WolframAlpha QA sustav



Izvor: <http://www.wolframalpha.com/>

Interes za ovu domenu smanjen je jer nije postojao opće prihvaćen način za ocjenjivanje sustava i samo ocjenjivanje poprimilo puno veću ulogu u metodologiji prirodnog jezika tijekom 90-ih godina. Testovi razumijevanja teksta nude rješenje tog problema; rješenje koje ima dodatnu prednost jer je dostupno za ljude i ne zahtijeva poseban trud za proizvodnju.

Istraživanja razumijevanja teksta pokazuju da odgovor na pitanje treba biti izveden iz nestrukturiranih tekstova, kao kod QA sustava. Pitanja mogu biti postavljena slijedno, gdje jedno pitanje proizlazi iz drugog (*Tko je bio američki predsjednik 1958? A 1960? ... Kojoj stranci je pripadao?*). Za razliku od otvorenog teksta kod odgovaranja na pitanja, tekst koji sadrži odgovor unaprijed je poznat. Velik broj pitanja o jednom tekstu zahtijeva detaljniju obradu tog teksta i rješavanje problema neusklađenosti koje donose slični ali nebitni tekstovi. Testovi koji ocjenjuju razumijevanje uzrokuju manju redundanciju, što dodatno otežava traženje lokacije odgovora.

3.3 Dohvat informacija, ekstrakcija informacija i odgovaranje na pitanje

Dohvaćanje informacija ili *Information Retrieval* (IR), je pronalaženje bitnih dokumenata kao odgovor na korisnički upit. Od sredine 1950-ih IR je područje aktivnog istraživanja (Spärck Jones Willett, 1997). Povezano je s odgovaranjem na pitanja jer korisnik kreira upit za pronalaženje odgovora. Sličnost tu uglavnom i završava. IR sustavi vraćaju dokumente, ne odgovore, na korisnicima je da sami izdvoje odgovore iz dokumenata. Upiti koje korisnici postavljaju IR sustavima ne moraju biti sintaktički oblikovani kao upitne rečenice, štoviše, to im može štetiti. Neprimjetne sintaktičke razlike, kao kod pitanja *Tko je ubio medvjeda?* i *Koga je ubio medvjed?* potpuno se gube u IR sustavima.

IR sustav nužan je za odgovaranje na pitanja iz dvaju razloga. Prvo, IR tehnike proširene su tako da ne vraćaju samo relevantne dokumente, nego i nebitne odlomke koji se nalaze u dokumentu. Veličina tih odlomaka može biti redovito smanjena, bar u teoriji, tako da u ograničenom broju slučajeva, ono što je izdvojeno je učinkovito, samo čist odgovor na pitanje. Drugo, IR zajednica tijekom godina razvila je iznimno temeljitu metodologiju za ocjenjivanje. Najpoznatija je godišnja TREC konferencija, čija je metodologija i zajednica zaslužna za nedavni razvoj ocjenjivanja QA-a.

Drugi dio istraživanja TREC-a odnosi se na ekstrakciju informacija ili *Information Extraction* (IE), u početku nazvano razumijevanje poruke. Može se definirati kao aktivnost popunjavanja unaprijed definiranih predložaka iz teksta prirodnog jezika, gdje su predlošci dizajnirani za sakupljanje informacija o ključnim ulogama u tipičnim događajima (Gaizauskas and Wilks, 1998). Uzmimo za primjer predložak koji sakuplja informacije o događajima neke korporacije. Predložak će imati praznine za kupnju tvrtke, stečenu tvrtku, datum kupnje, isplaćen iznos... Upravljanje IR sustavom za ispunjavanje predloška s velikim količinama teksta rezultirat će strukturiranom bazom informacija o korporativnim preuzimanjima. Ta baza može poslije biti korištena i za druge svrhe, kao što su upiti, pronalaženje znanja (*data mining*) i sažimanje. Predloške za IE možemo shvatiti kao izražavanje pitanja, a ispunjen predložak dio koji sadrži odgovor. IR je ograničen oblik odgovaranja na pitanja u kojem su pitanja (predlošci) statički, a baza iz koje se dobivaju odgovori proizvoljno velika zbirka tekstova. IR zajednica konstruirala je svoju vježbu za ocjenjivanje *Message Understanding Conferences* ili MUC koja je trajala od 1987. do 1998. Završetak MUC vježbe, razvoj tehnologije za razumijevanje jezika kroz vježbe ocjenjivanja, omogućio je TREC-u uvjete za ocjenjivanje odgovaranja na pitanja.

3.4 Logika pitanja i odgovora

Objašnjenje pojma „erotetička logika“ nužno je za logiku pitanja (Hirschman, Gaizauskas, 2001). Erotetička logika je dio logike koji analizira pitanja kao logičke entitete koji se razlikuju od izjava. Erotetičku logiku često nazivaju logika pitanja. Zanimanje logičara bilo je usmjereno na dobar formalni zapis i skup razlika pojmova za istraživanje pitanja i odgovora.

Logika pitanja započinje pretpostavkom da je korišteni jezik jasno definiran i određen. Definirani su ključni pojmovi osnovnih pitanja (kondicionalna pitanja *ako* ili obična pitanja tipa *što, koji, koje...*) i direktni odgovori pokazuju kako službene definicije sadrže osnovne intuicije o osnovnim pitanjima i odgovorima na njih (Harrah, 1984; Belnap, Steel, 1976).

Nastavili su s istraživanjem složenijih oblika pitanja i odgovora, pretpostavki, učinkovitosti i cjelovitosti. Rezultat je skup potencijalno korisnog analitičkog alata za automatizirano odgovaranje na pitanja.

Logičare ne zanima izvedba odgovora na pitanja u praksi. Započinju s radom o računalnim odgovorima na logičke upite naspram logičkih baza podataka (Green, 1969). Govori se o korištenju rezolucijske teorije dokaza kako bi dobili instancu koja se nalazi u konstruiranju dokaza o postojećoj kvantificiranoj formuli što nas vodi do logičkog programiranja i deduktivne baze podataka⁸.

4. PREGLED DOSADAŠNJIH ISTRAŽIVANJA

Najvažnija područja primjene QA su crpljenje informacija iz cijelog *weba, online* baza podataka i upiti na pojedinim *web*-stranicama. Trenutni QA sustavi koriste tekstualne dokumente kao temeljni izvor znanja i kombiniranjem različitih tehnika NLP-a traži se odgovor na pitanje korisnika (Allam et al., 2012). U cilju pružanja točnih odgovora korisnicima, QA sustavi moraju leksičko-sintaktičku analizu primijeniti na semantičku analizu i obradu tekstova i resursa znanja. QA sustavi opremljeni su sposobnošću zaključivanja i mogu izvući više adekvatnih odgovora za predstavljanje znanja i sustav zaključuje slično opisu logike i ontologije. Ontologija je osnovica QA sustava (Lopez et al., 2011), studija o prikazu uporabljivosti NL sučelja i NL upita preko ontologije na kojoj se temelji baza znanja za krajnjeg korisnika (Kaufmann et al., 2010). Autori su predstavili četiri sučelja koja omogućuju različite jezike za pretraživanje i predstavljaju komparativnu studiju o njihovoj uporabi. Zaključili su kako korisnici imaju jasnu prednost kod upita izraženih u NL-u i mali je skup izraza sastavljenih s nekim ključnim riječima ili neke formalne strukture.

Značajan broj sustava prikazanih u TREC ocjenjivanju u mogućnosti su uspješno kombinirati IR i tehnike NLP-a. QA sustavi mogu se pregledati kroz tri glavna pristupa: sustavi temeljeni na NLP-u, IR sustavi i sustavi temeljeni na definiranom predlošku pitanja-odgovor (Andrenucci et al., 2005). Autori su istakli njihove glavne razlike i primjenu konteksta koji je primjereniji za svaki sustav.

Suradnički QA sustav je automatizirani sustav u kojem je polazište unos upita, pokušava uspostaviti kontrolirani dijalog sa svojim korisnikom, odnosno, sustav automatski surađuje s korisnicima kod pronalazjenja podataka koji se traže. Ovi sustavi pružaju korisnicima dodatne informacije, srednje odgovore, kvalificirane odgovore ili alternativne upite. Suradničko ponašanje uključuje povezane

⁸ Relacijska baza podataka proširena deduktivnom komponentom.

informacije relevantne za upit. Generaliziranjem upita dobivaju se relevantne informacije. Sustavi za suradničko odgovaranje koriste automatsko identificiranje novih upita koji se odnose na izvorni upit (Gaasterland, 1997). Prilagodбом tehnike strojnog učenja definira se izvlačenje informacija za odgovore u QA sustavu (Jousse et al., 2005). Autori su identificirali specifičnosti sustava i testirali i usporedili tri algoritma, pod pretpostavkom apstrakcije NL tekstova. Semantički zastupljen formalizam posvećen suradničkim QA sustavima temelji se na konceptualnim i leksičkim strukturama i predstavlja homogene *web*-tekstove, NL pitanja i srodne odgovore (Benamara, 2008). Autor predstavlja i analizira neke od preduvjeta za izgradnju suradničkih odgovora, ovisno o resursima, znanju i procesima. U cilju jačanja suradničkih QA sustava, prikazan je skup tehnika za poboljšanje tih sustava i rasprava o potencijalnom utjecaju njihovog korištenja (Meguinness, 2004).

Suradnički odgovor [10,13] na NL pitanje je jedan neizravan odgovor koji je korisniku tada korisniji nego izravan i doslovan odgovor (Corella, Lewison, 2009). Suradnički odgovor može objasniti neuspjeh računanja rezultata i/ili predložiti pitanja za nastavak istraživanja. Suradnički odgovor može dostaviti dodatne informacije koje nije izričito zahtijevao korisnik i uklapaju se u kontekstu osiguranja kvalitete sustava te su motivirani željom za približavanje sustava za dijalog s korisnikom. Suradnička obrada odgovora humanizira sustav i omogućuje korištenje prilagođenog rječnika .

Autori de Sena i Furtado predlažu model QA sustava koji pokušava uspostaviti kontrolirani dijalog s korisnikom izvan korisničkog pitanja. U dijalogu, sustav ima svoj glavni cilj identificirati korisničko pitanje i predložiti novo pitanje u svezi s korisničkim početnim pitanjem. Kontrolor dijaloga temelji se na konceptu strukture u bazi znanja, u ograničenjima domena i klimatizaciji posebnih pravilima. Prototip sustava koji vraća suradničke odgovore korigira pojmove koji nedostaju, namjerava zadovoljiti potrebe korisnika te koristi semantičke informacije baze podataka u svrhu dobivanja koherentnih i informativnih odgovora (Gaasterland, 1992 c). Glavne karakteristike leksičkih strategija govore kako odgovoriti na pitanja (Benamara et al., 2004 c). Autor predstavlja strategiju reproduciranja u izgradnji sustava osiguranja kvalitete posebno u inteligentnim suradničkim QA sustavima. Metoda traženja susjednih odgovora početnom pitanju korisnika može se koristiti za obradu odgovora koji mogu zadovoljiti korisničke potrebe i zahtjeve (Gaasterland, 1992 b).

Korištenje naprednog zaključivanja QA sustavima postavlja nove izazove za istraživače jer odgovori nisu samo izvađeni izravno iz teksta ili iz strukturirane baze podataka, nego su izgrađeni od nekoliko oblika zaključivanja s ciljem stvaranja i opravdanja odgovora. Integrirani mehanizmi koji zastupaju znanje i razmišljanje omogućiti će sustave koji predviđaju odgovor na pitanja koje se ne može naći u bazi znanja. Ti sustavi trebaju identificirati i objasniti lažne pretpostavke i druge vrste sukoba koji se mogu naći u pitanju (Melo et al., 2013).

Suradnički QA sustavi prikazuju se koristeći baze podataka kao izvor znanja (Minker, 1998). Benamara predlaže logiku izrade predložaka za generiranje namjernih i preciznih odgovora u suradničkim QA sustavima (Benamara, 2004 b). Sejid QA sustav i WEBCOOP integriraju predstavljanje znanja i tehnike zaključivanja kako bi potaknuli suradnju odgovora na pitanja koje postavlja NL na *webu* (Benamara, 2004 a). PowerAqua je multi-ontološki QA sustav (Lopez, Motta, 2006).

5.1 Analiza pitanja

Ulaz u ovoj fazi je pitanje prirodnog jezika. Prvo, moguća su ograničenja unosa i prirodnog jezika. Od korisnika se traži da koristi podskup prirodnog jezika, „kontrolirani jezik“ koji ima ograničen vokabular i sintaksu (što je slučaj kod većine sučelja baza prirodnog jezika). Korisnik može biti prisiljen koristiti model sučelja koje sustavu značajno pojednostavljuje interpretaciju pitanja. Drugo, izravan unos pitanja može biti u obliku konteksta, u slučaju da sustav podržava kontinuirani dijalog. U pitanju može biti elipsa⁹ ili anafora¹⁰, što zahtijeva interpretaciju konteksta dijaloga. Drugi implicitni unos može biti znanje sustava o korisniku i njegovim ciljevima.

Izlaz ove faze je jedna ili više reprezentacija pitanja koje će se koristiti u daljnjim fazama. Mehanizam koji bira potencijalne dokumente u sljedećim fazama govori o IR sustavu i tada jedna reprezentacija pitanja može biti osnova, tj. usmjerivač za unos u tražilicu. Većina sustava koristi detaljniju analizu pitanja koja uključuje:

1. identifikaciju semantičkog tipa entiteta koje traži pitanje (datum, osoba, tvrtka...)
2. utvrđivanje dodatnih ograničenja na odgovor entiteta, npr:
 - a) identifikaciju ključne riječi u pitanju koja se koristi za odgovarajuću potencijalnu rečenicu koja sadrži odgovor ili
 - b) identifikaciju odnosa, sintaktičkih ili semantičkih, koji postoje između entiteta koji su kandidat za odgovor i svih ostalih entiteta ili događaja spomenutih u pitanju.

Prvi korak odnosi se na traženje ključne upitne riječi (*kada* zahtijeva datum ili vrijeme, *gdje* lokaciju, *tko* osobu...). Samo to nije dovoljno, budući da razne upitne riječi, kao što su *koje* i *što* ne zahtijevaju određen tip informacije. Pitanja tipa „*Koja tvrtka..?*“ ili „*Koja građevina..?*“ vrlo su jednostavna dok su puno složenija pitanja koja uključuju sintaktički složenije konstrukcije kao što su „*Tko su bili Beatlesi?*“, „*Koja je njihova prva pjesma?*“ ili „*Koliko prvih nagrada iz informatike je dodjeljeno na Cambridgeu prošle godine?*“

Različiti sustavi imaju izgrađenu hijerarhiju tipa pitanja baziranu na tipu odgovora i pokušavaju svrstati postavljeno pitanje u odgovarajuću kategoriju hijerarhije. Ručno se gradi hijerarhija tipa pitanja od cca 25 tipova iz analize TREC podataka (Moldovan et al. 2000). Baziraju svoju hijerarhiju tipa pitanja na temelju MUC imenovanih klasa entiteta i koriste plitki parser¹¹ za identifikaciju tipa pitanja (Srihari, Li, 2000). Nazvan je upitna točka. Konstruiraju IR tipologiju sastavljenu od 47 kategorija baziranih na analizi 17 000 stvarnih pitanja, proširivši analizu na semantičke tipove doslovno zatražene klasificirajući pitanja kao što su „*Tko je otkrio Ameriku?*“ na osobu i pitanja tipa „*Tko je bio Christopher Columbus?*“ na *zašto*-poznat (Hovy et al., 2001). Na 9. TREC konferenciji opisana se ručno napisana vrhunski rangirana hijerarhija tipa odgovora koja povezuje dijelove WordNeta¹² kako bi proširila skup mogućih tipova odgovora dostupnih svom sustavu (Harabagu et al., 2001).

⁹ Figura koja nastaje kad se iz rečenične cjeline izostavljaju pojedine riječi.

¹⁰ Glasovna figura kojom se ponavljaju riječi na početku dvaju ili više stihova.

¹¹ Raščlanjivanje ili sintaktička analiza je proces analiziranja niza simbola, bilo u prirodnom jeziku ili računalnom

¹² Velika leksička baza engleskog jezika.

Identifikaciju tipa entiteta slijedi identifikacija dodatnih ograničenja s kojima se susreću entiteti koji odgovaraju opisu određenog tipa. Ovaj postupak vrlo je jednostavan, poput izlučivanja ključnih riječi od ostatka pitanja koje se usklađuju s potencijalnim rečenicama koje sadrže odgovore. Skupina ključnih riječi kasnije može biti proširena, koristeći sinonime i morfološke varijante (Srihari, Li, 2000) ili tehnikama potpunog proširenja upita, primjerice izdavanje upita temeljenog na ključnoj riječi iz enciklopedije i korištenje najbolje rangiranih ponuđenih odlomaka za proširenje skupa ključnih riječi. Postupak identifikacije ograničenja može uključivati parsiranje pitanja s gramatikama različite sofisticiranosti. Harabagiu koristi statički parser široke primjene čiji je cilj proizvesti potpune parsere. Sastavni dio analize koje pitanje proizvodi pretvara se u semantičku reprezentaciju koja bilježi zavisnosti između pojmova u pitanju. Koristi se otporan djelomični parser koji može odrediti gramatičke odnose u pitanju. Kada se ti odnosi povežu s identificiranim i traženim entitetom, prosljeđuju se kao ograničenja na koja treba računati tijekom ekstrakcije odgovora (Scott, Gaizauskas, 2001).

5.2 Predobrada zbirke dokumenata

Ako se na pitanja odgovara u stvarnom vremenu, predobrada teksta je nužna. Većina se TREC IR sustava oslanja na mehanizme koji indeksiraju¹³ dokumente i nema predobrade. Odabir potencijalnog dokumenta oslonjenog na konvencionalnu tražilicu za prvi odabir dokumenata i prijevremeno spremanje analize svih tekstova čini 4. fazu ovog modela nepotrebnom.

Sustav može izgraditi cijele zbirke tekstova logičkog značenja prije izlučivanja odgovora. Primjer je *ExtrAns*¹⁴ sustav (Molla et al., 1998). *ExtrAns* izvodi logičku reprezentaciju zbirke dokumenata unaprijed za svaki upit. Sustav koji dozvoljava plitku lingvističku obradu zbirke dokumenata unaprijed, jest sustav koji dohvaća informacije (Milward, Thomas, 2000). On prepoznaje imena entiteta, dijeli velike dokumente na dijelove a potom pohranjuje rezultate kao indeksirana ograničenja koja se mogu podudarati sa stvarnim vremenom korisnikove interakcije sa sustavom. Prager unaprijed obrađuje zbirku dokumenata i obilježava pojmove s jednom od 50 semantičkih oznaka koje su indeksirane tijekom procesa indeksiranja dokumenta (Prager, 2001). Katz izdvaja trostruke odnose izraza (oblik, subjekt, objekt) iz sintaktičke analize rečenice prirodnog jezika na *web*-stranicama. Iz toga gradi proširenu bazu kako bi potpomogao daljnjem odgovaranju na pitanja (Katz, 1997).

5.3 Odabir potencijalnih dokumenata

Većina postojećih TREC QA sustava koristi neki oblik uobičajene IR tražilice za odabir početnog skupa potencijalnih dokumenata s odgovorom. Prvo, treba odlučiti kakvu tražilicu koristiti, *bool*¹⁵ ili rangirane odgovore. Unatoč tome što tražilice koje koriste rangirane odgovore daju bolje rezultate kod standardnog IR ocjenjivanja, neki TREC sudionici tvrde kako su *bool* tražilice prikladnije kod QA sustava (Moldovan et al., 2000). Ako se koristi tražilica rangiranih odgovora, treba odlučiti kako će se koristiti vraćeni dokumenti, tj. koliko ih uzeti. Kod *bool* tražilica potrebno je riješiti problem ograničavajućeg

¹³ Postupak u kojem informacijski sustav pregledava dokumente i katalogizira ih kako bi se omogućilo njihovo pretraživanje.

¹⁴ An Answer Extraction System, sustav koji dohvaća odgovore.

¹⁵ Pretraživanje u kojem se dodjeljuju vrijednosti (0 i 1 ili *true* i *false*) koje se onda uspoređuju.

broja vraćenih dokumenata. Drugo, tražilica može dopustiti preuzimanje odlomaka pa je potrebno postaviti različite parametre kao što je veličina paragrafa. U skladu s preuzimanjem, dijelovi teksta koji imaju istu temu mogu se koristiti za identifikaciju sukladnih dijelova teksta koji su kraći nego cijeli dokument i prema tome ih rangirati.

5.4 Analiza dokumenata

Odabrani potencijalni dokumenti ili paragrafi koji sadržavaju odgovor na pitanje mogu se podvrgnuti dodatnoj analizi. Ako je sustav već u potpunosti unaprijed obradio sve dokumente, ova faza nije potrebna. Najčešće, sustavi analiziraju odabrane dokumente koristeći identifikator imena entiteta koji prepoznaje i klasificira stringove od više riječi kao imena tvrtki, osoba, lokacija... Klase imena koja su identificirana trebale bi biti navedene u MUCu¹⁶. U mnogim slučajevima klase se proširuju kako bi uključile razne dodatne klase, kao što su proizvodi, adrese i mjere ili se predefiniiraju i tako uključe podklase poput gradova, pokrajina i zemalja.

Tipično za ovu fazu je dijeljenje rečenica, označavanje dijelova govora i parsiranje (identificiranje skupine imenica, glagola...).

Opisan je QA sustav koji koristi plitku sintaktičku analizu za identificiranje pojmova s više riječi i njihove varijante u odabranom dokumentu (Ferret et al., 2001). Sustav koristi ponovno indeksiranje i ponovno rangiranje dokumenata prije usklađivanja s reprezentacijom pitanja. Neki sustavi idu dalje i obavljaju detaljniju sintaktičku analizu iza koje slijedi neka vrsta pretvaranja izvedene sintaktičke strukture u skup relacijskih ograničenja. Ograničenja su izražena logičkim izrazima ili relacijskim oznakama između odabranog pojma i izvorne rečenice.

Analiza pitanja najčešće koristi statistički parser široke primjene za prikaz ovisnosti rečenica u potencijalnom dokumentu koji sadrži odgovor. Nakon toga prikazuje se ovisnost logičkom reprezentacijom, kao što je učinjeno i s pitanjem.

5.5 Ekstrakcija odgovora

Usklađuje se prezentacija pitanja i potencijalnog teksta koji sadrži odgovor. Nastaje skup potencijalnih odgovora, rangiranih prema vjerojatnosti točnosti.

Sustavi koji analiziraju pitanje i očekivan tip odgovora, kao i neki skupovi dodatnih ograničenja, analizirat će potencijalni dokument ili dijelove tog dokumenta. Proces usklađivanja može zahtijevati da jedinica teksta iz potencijalnog dokumenta (možda rečenica, ako je odrađena podjela) sadrži string čiji se semantički tip poklapa s očekivanim odgovorom. Usklađivanje može biti tipa podvrgavanja (tumačeno kao hiponomi¹⁷ u leksičkom sustavu kao što je WordNet) i ne treba biti ograničeno na identitet.

Nakon što je pronađena jedinica teksta koja sadrži očekivani tip odgovora, mogu se primijeniti na nju ostala ograničenja. Možemo ih shvatiti kao potpuna, te ako nisu zadovoljena kod određenog

¹⁶ Niz radionica osnovanih da bi potaknule razvoj novih i boljih metoda izlučivanja informacija; http://en.wikipedia.org/wiki/Message_Understanding_Conference

¹⁷ Riječ ili fraza čiji je sadržaj uključen u sadržaj nekog drugog pojma.

kandidata možemo ga odmah isključiti; ili kao povlastice koje se koriste za dodjeljivanje bodova kandidatu kod rangiranja odgovora. Značajne razlike između sustava postoje u smislu tipova ograničenja koja se koriste u ovoj fazi.

Nakon što je, primjerice, pronađen izraz točnog tipa odgovora u potencijalnom odlomku koji sadrži odgovor, taj se izraz izdvaja i pregledavaju se razne kvantitativne značajke kako bi se izračunao ukupan zbroj kojim će se taj izraz svrstati na ljestvicu ponuđenih odgovora (Harabagiu et al., 2000).

Za svaki se odlomak koji može sadržavati odgovor, a sadrži izraz koji je jednak tipu odgovora koji tražimo, izvodi ocjena. U konačnici te se ocjene računaju i uspoređuju kako bi se dobila ukupna rang ljestvica za sve potencijalne odgovore. Neki stručnjaci (Harabagiu et al., 2001) proširuju ovaj pristup koristeći algoritam strojnog učenja kako bi optimizirali linearno bodovanje koje kombinira značajke karakteristične za odgovor.

Može se izmijeniti redosljed ovog općenitog postupka. Prvo se primijene ograničenja pitanja, zatim očekivani tip odgovora za rangiranje rečenica koji se nalaze u potencijalnim dijelovima teksta koji sadrže odgovor. Očekivani tip odgovora je filter koji izdvaja odgovarajući dio iz odabranih rečenica. Za rangiranje rečenica koriste se značajke tipa: broj jedinstvenih ključnih riječi koje se nalaze u rečenici, redosljed ključnih riječi u rečenici u usporedbi s redosljedom u pitanju te podudaraju li se ključni glagoli (Srihari, Li, 2000).

Sustavi koji izdvajaju veće dokumente i prezentacije pitanja, npr. logičke oblike ili tekst obilježen sa semantičkim ili gramatičkim informacijama, mogu koristiti dodatna ograničenja kako bi ograničili proces usklađivanja. Primjer je sustav koji može identificirati logičke subjekte i objekte. On može točno odrediti *Jack Ruby* kao odgovor na pitanje *Tko je ubio Leeja Harveyja Oswalda?* iz rečenice *Ruby je ubio Oswalda, a Oswald je ubio Kennedyja*. Takva pitanja mogu biti problem kod nekih pristupa zbog preklapanja riječi.

Većina sustava koji koriste takva gramatička ograničenja imaju poteškoće jer je u tim slučajevima potrebno ograničenja promatrati samo kao povlastice koje nisu obvezne i kod njih dolazi do preklapanja riječi kada njihova ograničenja nisu primjerena.

5.6 Odgovor

Za QA ocjenjivanje jedini odgovor koji većina sustava stvara je rangirana lista najboljih 5 odgovora, svaki odgovor je tekst string od n bajtova (gdje je $n = 50$ ili $n = 250$) izdvojen iz jednog od tekstova iz zbirke dokumenata.

Ova vrsta odgovora je nedovoljna u stvarnim aplikacijama. Prvo, string od n bajtova izdvojen iz teksta uglavnom je gramatički neispravan i nečitljiv. Potreban je link na izvorni dokument kako bi omogućio lingvistički kontekst ili je potrebno preformulirati string kako bi bio razumljiv. Drugo, korisnici mogu željeti više ili manje dokaza, tj. konteksta za odgovor, ovisno o tome koliko korisnik vjeruje sustavu i koliko već zna o temi iz koje postavlja pitanje. Treće, odgovori mogu biti složeniji ili opsežniji nego što korisnik očekuje. Sustav je taj koji mora donositi odluku o tome treba li skratiti svoj odgovor te može li inicirati dijalog s korisnikom kako bi mogao donijeti odluku kako postupiti.

6. ZAKLJUČAK

Na temelju provedene analize povijesnog razvoja QA sustava zaključujemo kako im je razvoj išao od jednostavnih i specijaliziranih za usko područje do današnjih puno složenijih i kvalitetnijih sustava koji su sposobni dati kratke i sažete odgovore na pitanja iz različitih domena.

U programiranju ovakvih sustava stručnjaci se susreću s mnogim problemima, kao što su velike baze podataka, različiti tipovi korisnika, brojne vrste pitanja i odgovora. Upravo zbog toga najveći problem još uvijek predstavlja odabir najtočnijeg i najsazetijeg odgovora s kojim će korisnik biti zadovoljan.

Daljnja istraživanja sustava koji odgovaraju na pitanja ići će u smjeru proučavanja razvoja raznih tehnika ocjenjivanja odgovora s ciljem poboljšanja učinkovitosti sustava koji odgovaraju na pitanja.

LITERATURA

- Allam, A., Haggag, M. (2012) „The Question Answering Systems: A Survey“, International Journal of Research and Reviews in Information Sciences 2(3), 211-221
- Andrenucci, A., Sneider, E. (2005) „Automated question answering: Review of the main approaches“, ICITA (1). pp. 514-519. IEEE Computer Society
- Belnap, N. D. and Steel, T. B. (1976) „*The Logic of Questions and Answers*“, Yale University Press, New Haven and London
- Benamara, F. (2004a) „Cooperative question answering in restricted domains: the WEBCOOP experiment“, Proceedings of the Workshop Question Answering in Restricted Domains, within ACL
- Benamara, F. (2004b) „Generating intensional answers in intelligent question answering systems“, Natural Language Generation (2), 11-20
- Benamara, F. (2008) „A semantic representation formalism for cooperative question answering systems“ Proceeding of Knowledge Base Computer Systems (KBCS)
- Benamara, F., Saint-Dizier, P. (2004c) „Lexicalisation strategies in cooperative question-answering systems“, Proceedings of the 20th international conference on Computational Linguistics. p. 1179. No. Cruse 1986 in COLING '04, Association for Computational Linguistics, Stroudsburg, PA, USA
- Bobrow, D., Kaplan, R., Kay, M., Norman, D., Thompson, H. and Winograd, T. (1977) GUS, a frame driven dialog system. Artificial Intelligence, 8: 155-173.
- Corella, F., Lewison, K. (2009, Oct) „A brief overview of cooperative answering“, Journal of Intelligent Information Systems 1(2), 123-157
- Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G. and Jacquemin, C. (2001) Terminological variants for document selection and question/answer matching. Proceedings Association for Computational Linguistics Workshop on Open-Domain Question Answering, pp. 46-53.
- Gaasterland, T. (1997, Sep) „Cooperative answering through controlled query relaxation“ IEEE Expert: Intelligent Systems and Their Applications 12(5), 48-59
- Gaasterland, T., Godfrey, P., Minker, J. (1992a) „An overview of cooperative answering“, Journal of Intelligent Information Systems 1(2), 123-157
- Gaasterland, T., Godfrey, P. (1992b) „Relaxation as a platform for cooperative answering“ Journal of Intelligent Information 1(3), 293-321
- Gaasterland, T., Godfrey, P., Minker, J., Novik, L. (1992c) „A cooperative answering system“, Logic Programming and Automated Reasoning, No. X, Springer, pp. 478-480

- Green, B. F., Wolf, A. K., Chomsky, C. and Laughery, K. (1961) „BASEBALL: An automatic question answerer“, Proceedings Western Joint Computer Conference 19, pp. 219-224
- Harabagiu et al. (2001) „FALCON: Boosting knowledge for answer engines“, Proceedings 9th Text Retrieval Conference (TREC-9), NIST Special Publication 500-249 (Available at: <http://trec.nist.gov/pubs.html>) (11. 1. 2014.)
- Harrah, D. (1984) „The logic of questions“, Gabbay, D. and Guentner, F., editors, Handbook of Philosophical Logic, Vol. II, pp. 715-764, Reidel
- Hirschman, L., & Gaizauskas, R. (2001) „Natural language question answering: The view from here“ *Natural Language Engineering*, 7(4), 275-300.
- Hirschman, L., Light, M., Breck, E. and Burger, J. (1999) „Deep Read: A reading comprehension system“, Proceedings 37th Annual Meeting of the Association for Computational Linguistics, pp. 325-332
- Hovy, E., Gerber, L., Hermjacob, U., Junk, M. and Lin, C.-Y. (2001) „Question answering in Webclopedia“, Proceedings 9th Text Retrieval Conference (TREC-9), NIST Special Publication 500-249 (Available at: <http://trec.nist.gov/pubs.html>) (11. 1. 2014.)
- Jousse, F., Tellier, I., Tommasi, M., Marty, P. (2005) „Learning to extract answers in question answering: Experimental studies“, In CORIA (p. 85)
- Katz, B. (1997) „From sentence processing to information access on the world wide web“, Proceedings American Association for Artificial Intelligence (AAAI) Spring Symposium on Natural Language Processing for the World Wide Web, Stanford University, Stanford CA (Available at: <http://www.ai.mit.edu/people/boris/webaccess/>) (11. 1. 2014.)
- Kaufmann, E., Bernstein, A. (2010, Nov) „Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases“, Web Semantics: Science, Services and Agents on the World Wide Web 8(4), 377-393
- Kupiec, J. (1993) „Murax: A robust linguistic approach for question answering using an online encyclopedia“, roceedings Association for Computing Machinery Special Interest Group on Information Retrieval (ACM-SIGIR) '93, pp. 181-190. Pittsburgh, PA
- Lehnert, W. G. (1977) „*The Process of Question Answering*“, (No. RR-88). YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE
- Lehnert, W. G. (1977, August) „A conceptual theory of question answering“ In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume (1)*, pp. 158-164, Morgan Kaufmann Publishers Inc.
- Light, M., Mann, G. S., Riloff, E., & Breck, E. (2001) „Analyses for elucidating current question answering technology“, *Natural Language Engineering*, 7(4), 325-342
- Light, M., Mann, G. S., Riloff, E., & Breck, E. (2001) „Analyses for elucidating current question answering technology“, *Natural Language Engineering*, 7(4), 325-342
- Lin, J. (2002) „The Web as a resource for question answering: Perspectives and challenges“, Proceedings of the Third International Conference on Language Resources and Evaluation. pp. 2120-2127. No. Lrec, Citeseer
- Lopez, V., Motta, E. (2006) „Powersqua: Fishing the semantic web“, Semantic Web: Research and Applications
- Lopez, V., Uren, V., Sabou, M., Motta, E. (2011) „Is question answering fit for the semantic web?: a survey“, Semantic Web, 2(2), 125-155
- Maybury, M. (2004) „New directions in question answering“, Elements, pp. 533-558
- McGuinness, D.L. (2004) „Question Answering on the Semantic Web“, IEEE Intelligent Systems pp. 6-9
- Melo, D., Pimenta Rodrigues, I., & Beires Nogueira, V. (2013) „A Review on Cooperative Question-Answering Systems“

- Milward, D. and Thomas, J. (2000) „From information retrieval to information extraction“, Proceedings Association for Computational Linguistics (ACL) Workshop on Recent Advances in Natural Language Processing and Information Retrieval (Available at: <http://www.cam.sri.com/html/highlight.html>) (18. 1. 2014.)
- Minker, J. (1998) „An overview of cooperative answering in databases“, Flexible Query Answering Systems pp. 282-285
- Moldovan et al. (2000) „LASSO: A tool for sur_ng the answer net“, Proceedings 8th Text Retrieval Conference (TREC-8), NIST Special Publication 500-246 (Available at: <http://trec.nist.gov/pubs.html>) (13. 1. 2014.)
- Molla Aliod, D., Berri, J. and Hess, M. (1998) „A real world implementation of answer extraction“, Proceedings 9th International Conference on Database and Expert Systems Applications Workshop `Natural Language and Information Systems' (NLIS'98), pp. 143-148
- Pantel, P., Lin, T., & Gamon, M. (2012) „Mining entity types from query logs via user intent modeling“, In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume (1)*, pp. 563-571
- Prager, J. (2001) „One search engine or two for question answering“, Proceedings 9th Text Retrieval Conference (TREC-9), NIST Special Publication 500-249 (Available at: <http://trec.nist.gov/pubs.html>) (11. 1. 2014.)
- de Sena, G.J., Furtado, A.L. (1998) „Towards a cooperative question-answering model“, Flexible Query Answering Systems 1495, 354-365
- Scott, S. and Gaizauskas, R. (2001) „University of Sheffield TREC-9 Q & A System“, Proceedings 9th Text Retrieval Conference (TREC-9), NIST Special Publication (Available at: <http://trec.nist.gov/pubs.html>) (11. 1. 2014.)
- Simmons, R. F. (1965) „Answering English questions by computer: a survey“, *Communications of the ACM*, 8(1), 53-70.
- Sparck Jones, K. and Willett, P., editors. (1997) „*Readings in Information Retrieval*“ Morgan Kaufmann, San Francisco, CA
- Srihari, R. and Li, W. (2000) „Information extraction supported question answering“, Proceedings 8th Text Retrieval Conference (TREC-8), NIST Special Publication 500-246 (Available at: <http://trec.nist.gov/pubs.html>) (11. 1. 2014.)
- Turing, A. (1950) „Computing machinery and intelligence“, *Mind*, 59(236): 433-460.
- Voorhees, E. M., & Tice, D. M. (2000, July) „Building a question answering test collection“, In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 200-207), ACM
- W. A. Woods (1977) „Lunar rocks in natural english: Explorations in natural language question answering“, In A. Zampolli, editor, *Linguistic Structures Processing*, volume (5) of *Fundamental Studies in Computer Science*, pages 521–569, North Holland,
- Wilensky, R., Arens, Y., & Chin, D. (1984) „Talking to UNIX in English: an overview of UC“, *Communications of the ACM*, 27(6), 574-593
- Winograd, T. (1972) *Understanding Natural Language*, Academic Press, New York
- Woods, W. (1973) „Progress in natural language understanding - an application to lunar geology“ American Federation of Information Processing Societies (AFIPS) Conference Proceedings, 441-450

Jasminka Tomljanović¹

Marina Krsnik²

Mile Pavlić³

Professional paper

UDK 007

QUESTION ANSWERING SYSTEMS⁴

ABSTRACT

The question answering systems are a way of information retrieval that responds to the question asked in natural language. They consist of three main components : question classification, information retrieval and response extraction. The implementation of large-scale joint assessments such as TREC conference, has created a powerful interest community and accelerated the progress in the study of question answering systems. This paper presents the problem domains in answering questions. It reviews the historical development of the question answering systems, from the simplest highly specialized ones to the present more sophisticated and higher quality systems that are able to give us short and concise answers to questions belonging to a variety of domains. The systems are analyzed according to the purpose. The paper provides an overview of previous research on question answering systems. It defines the general architecture of question answering systems and analyzes current approaches to every phase of architecture. The question answering systems are still under development and experimentation. In conclusion, the selection of the most accurate and the most concise response that will satisfy the user, still remains the largest problem in question answering systems.

Keywords: question answering systems, TREC conference, architecture for question answering systems

¹ MSc, Senior Lecturer, Polytechnic of Rijeka, Vukovarska 58, Rijeka, Croatia. E-mail: jasminka.tomljanovic@veleri.hr

² Student, Department of informatics, University of Rijeka, Radmile Matejčić 2, Rijeka, Croatia.
E-mail: marina.krsnik@inf.uniri.hr

³ PhD, Full Professor, Department of informatics, University of Rijeka, Radmile Matejčić 2, Rijeka, Croatia.
E-mail: mile.pavlic@ris.hr

⁴ Received: 16. 2. 2014.; accepted: 5. 5. 2014.

