Fakultet elektrotehnike i računarstva – Zagreb

Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Predmet: ZER17D1 Otkrivanje znanja u skupovima podataka

Smjer: Jezgra računarstva

Šk. god. 2002/2003

Nastavnik: Prof. dr. sc. Nikola Bogunović, Doc. dr. sc. Bojana Dalbelo-Bašić

Upotreba stabla odlučivanja kod testiranja znanja metodom kviza

seminarski rad

Branko Žitko

Zagreb, studeni 2003

Kazalo:

1 Uvod

Računalni sustavi namijenjeni potpori i poboljšanju procesa učenja i poučavanja su u sve široj uporabi. Tu se posebno ističu sustavi zvani inteligentni tutorski sustavi (ITS) i zasnivaju se na simulaciji učitelja, odnosno njegovih karakteristika [STAN1997]. Njihova osnovna funkcija je vođenje učenika kroz nastavni sadržaj kojeg je učitelj prethodno pripremio. Inteligentni tutorski sustavi mogu se koristiti u svakodnevnom učenju i poučavanju, kao i kod učenja i poučavanja na daljinu. Učenik je sada ne samo prostorno, nego i vremenski slobodan odlučiti kada će koristiti resurse sustava.

Kod ovakvih sustava nastavni sadržaj se kreira na osnovu ekspertnih znanja ovisno o tome što se želi učenika podučiti tijekom procesa nastave. Recimo da želimo učenika naučiti osnove računarstva što uključuje nastavne teme kao što su pokretanje računala, korištenje operativnog sustava, osnove računalnih mreža i osnove programiranja. Za takvu nastavu bi se nastavni sadržaj kreirao na osnovu ekspertnih znanja o tehničkoj i programskoj podrški računalnog sustava, znanja o operativnom sustavu računala i računalnim mrežama, i ekspertnog znanja o nekom programskom jeziku kao što je QBasic.

Ekspertno ili područno znanje u inteligentnim tutorskim sustavima je znanje o području unutar kojeg se učenik poučava. Takvo znanje mora biti razumljivo računalu i čovjeku. Ono mora biti strukturirano tako da ga se može pohraniti i obrađivati u računalu, te da se može prenositi s jednog računala na drugo. Struktura znanja treba osiguravati njegovo korištenje, tj. stjecanje znanja, pretraživanje znanja i zaključivanje pomoću znanja [FIRE1988].

Znanje općenito možemo podijeliti na deklarativno i proceduralno znanje. Takva podjela utječe na samu strukturu znanja i na način prezentacije elemenata znanja.

- Deklarativno znanje nastoji opisati postojanje činjenica, zakonitosti, i slično. Takvim znanjem se utvrđuje veza među elementima znanja, odnosno nastoji se opisati postojeća situacija.
- Znanje o tome kako se nešto odvija, odnosno kako elementi znanja surađuju u postizanju cilja spada u proceduralno znanje. Za proceduralno znanje je važno opisivanje samog procesa, odnosno elementi takvog znanja će biti koraci koji čine neki proces.

Deklarativno znanje o tehničkoj podrški računalnog sustava opisivalo bi elemente računalnog sustava kao što su ulazne i izlazne jedinice kao dijelove računalnog sustava, gdje je tipkovnica vrsta ulazne jedinice, a monitor vrsta izlazne jedinice. O tome kako se ostvaruje proces prikaza znaka na monitoru nakon što se pritisne tipka na tipkovnici govori proceduralno znanje. Skup elemenata znanja u ovom primjeru sadrži sljedeće elemente: monitor, prikaz znaka na monitoru, računalni sustav i ostale. Elementi se moraju nekako opisati tako da se ostvari odnos među njima, te redoslijed koraka u ostvarivanju procesa.

Nakon što je određen skup pojmova koji će biti pohranjeni u bazu znanja, slijedi definiranje odgovarajućih komponenti znanja koje će realizirati željeno područno znanje.

Određivanje komponenti znanja obuhvaća imenovanje elemenata znanja, opis njihovih svojstava, način organizacije elemenata znanja u kategorije, opis veza među elementima znanja i ograničenja u opisu elemenata znanja [PARC1988].

- Kod imenovanja elementa znanja određujemo oznaku koja će predstavljati taj element. Tako je "labrador" ime za vrstu psa, "Microsoft" je ime za firmu i tako dalje.
- Neki elementi znanja imaju svoja svojstva koja ih podrobnije opisuju i pomažu nam da kod pretraživanja znanja na osnovu vrijednosti tih svojstava pronađemo baš onaj element koji nam je potreban. Primjer svojstava i njihovih vrijednosti kod računalnog procesora Pentium su prikazani u tablici 1.1.

svojstvo	vrijednost svojstva
vrsta	Pentium III
glavna brzina	1.4 GHz
brzina sabirnice	133 MHz
priručna memorija	256 KB
broj nožica	370

Tablica 1.1 Svojstva i vrijednosti Pentium procesora

- Pretraživanje znanja se značajno može olakšati razvrstavanjem elemenata znanja u kategorije. Na slici 1.1 je primjer razvrstavanja znanja o životinjama u kategorije.

Slika 1.1 Hijerarhijska organizacija znanja o životinjama

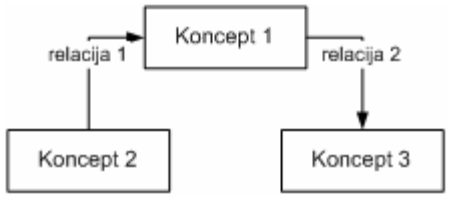
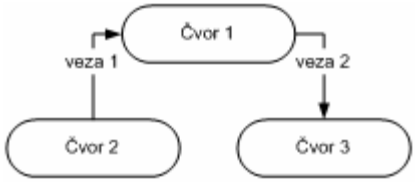
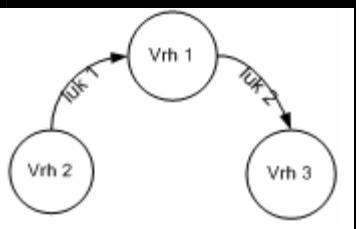
- Veze među elementima ovise o vrsti znanja koje se želi prikazati. Tako veze možemo podijeliti na deklarativne i proceduralne, odnosno na one veze koje opisuju statičke i dinamičke veze među elementima znanja. Kod prikaza deklarativnog znanja koriste se tri tipa deklarativnih veza:
    - Strukturalne deklarativne veze služe za opisivanje povezanosti elemenata znanja koji čine cjelinu, npr. opis povezanosti elemenata u elektroničkom sklopu.
    - Da bi opisali kategorije znanja koristit ćemo porodični tip deklarativnih veza. Recimo različite vrste životinja su opisane pomoću tih veza.
    - Veze zakonitosti kod deklarativnih veza govore o povezanosti elemenata znanja koji ne ovise o drugim elementima znanja. Tako je računalni sustav zasebna cjelina baš kao i računalna mreža. Ta dva sustava ako se povežu čine mrežni računalni sustav.

  Proceduralne veze opisuju korake akcija u obavljanju procesa. Tako su kod procesa fotosinteze elementi znanja biljka, kisik, sunčeve zrake i ugljični dioksid. Sami koraci u proceduri proizvodnje kisika u kojem sudjeluju biljka, sunčeve zrake i ugljični dioksid čine proceduralni tip veza.
- Ograničenja u opisu elemenata znanja pomažu kod provjere ispravnosti znanja kao i kod kreiranja samog znanja. Takva ograničenja se primjenjuju na svojstva elemenata, na njihovu hijerarhijsku povezanost i na same veze među elementima. Što se tiče ograničenja kod svojstava onda visina čovjeka ne može biti veća od 3 metra. Također kod povezivanja elemenata znanja kao što su čovjek i ruka, onda je ruka dio čovjeka, dok čovjek ne može biti dio ruke.

Različite metode za prikaz znanja nastoje što podrobnije opisati znanje i svu njegovu složenost. Jedna od metoda prikaza znanja su semantičke mreže.

2 Semantičke mreže

Semantičke mreže su razvijene na osnovi modela asocijativne memorije čovjeka [QUIL1968]. Zbog toga se još zovu i asocijativne mreže. Znanje unutar asocijativne mreže se prikazuje pomoću koncepata i relacijama među njima. Kod semantičke mreže koncepti su čvorovi, a relacije se ostvaruju vezama među čvorovima. Podloga za takvu strukturu povezanih čvorova čine usmjereni grafovi u matematičkoj teoriji grafova gdje su vrhovi povezani jednosmjernim lukovima s ostalim ili istim vrhovima. Sličnost ovih prikaza dana je u tablici 2.1.

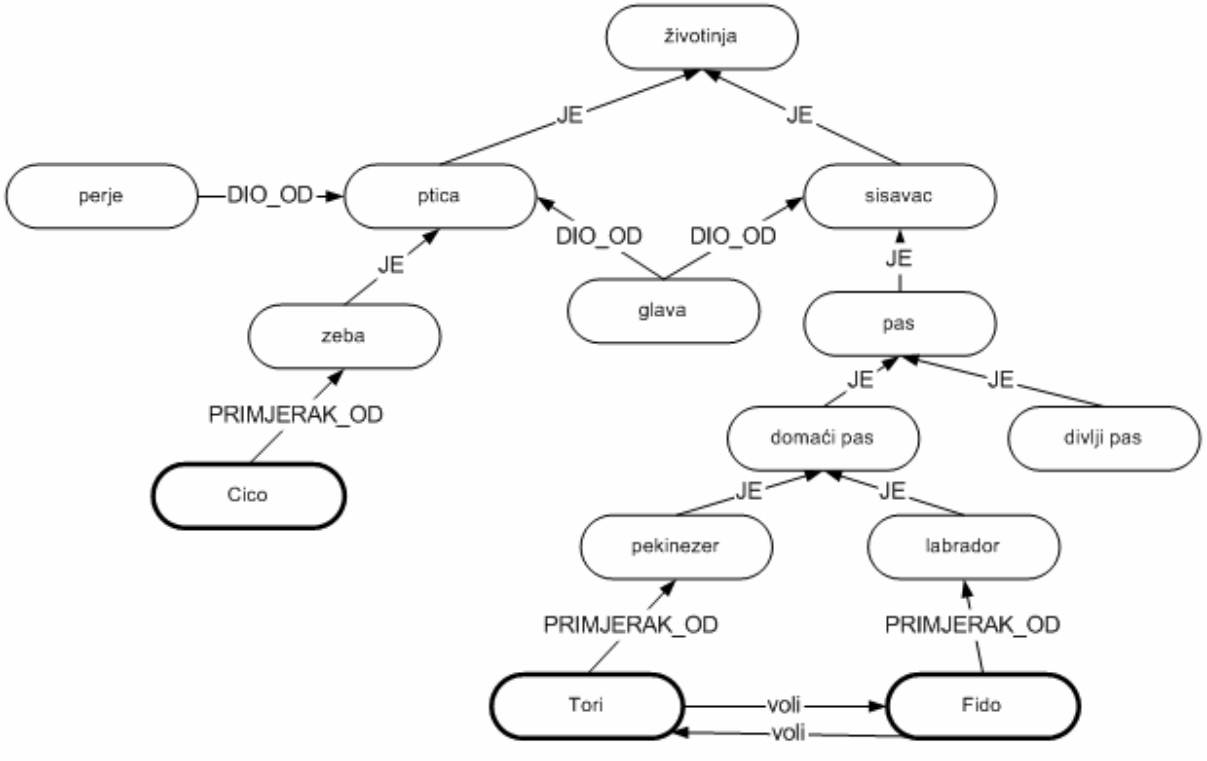
Asocijativna mreža	Semantička mreža	Usmjereni graf
		

Tablica 2.1 Prikazi asocijativne mreže, semantičke mreže i usmjerenog grafa

Često se kod semantičkih mreža koriste sljedeći tipovi veza [ANAS1998]:

- veze pripadnosti konceptu služe za hijerarhijsku kategorizaciju koncepata u semantičkoj mreži. Primjer je veza JE (IS_A) koja govori o odnosu između podčvora i nadčvora, gdje podčvor nasljeđuje svojstva nadčvora;
- veze koje govore o primjercima koncepata, kao što je veza PRIMJERAK_OD (INSTANCE_OF);
- veze koje strukturalno raščlanjuju koncept na dijelove, primjerice veza DIO_OD (PART_OF).

Uz takvu skupinu veza javlja se potreba korištenja različitih vrsta relacija među konceptima kao i relacije između njihovih primjeraka. Na slici 2.1 veza "voli" je takva vrsta veze koja govori o relaciji između dva primjerka domaćeg psa. Vide se i osnovna tri tipa veza, pa tako je Fido instanca labradora, a labrador je vrsta domaćeg pasa. Pas je vrsta sisavca, a oni imaju glavu, baš kao što je i ptice imaju.



Slika 2.1 Semantička mreža s primjerom klasifikacijske hijerarhije životinja

Čvorovi kao komponente znanja mogu imati svojstva. Prikaz svojstva nekog čvora se može vrlo jednostavno opisati uvođenjem okvira u semantičke mreže.

2.1 Semantičke mreže s okvirima

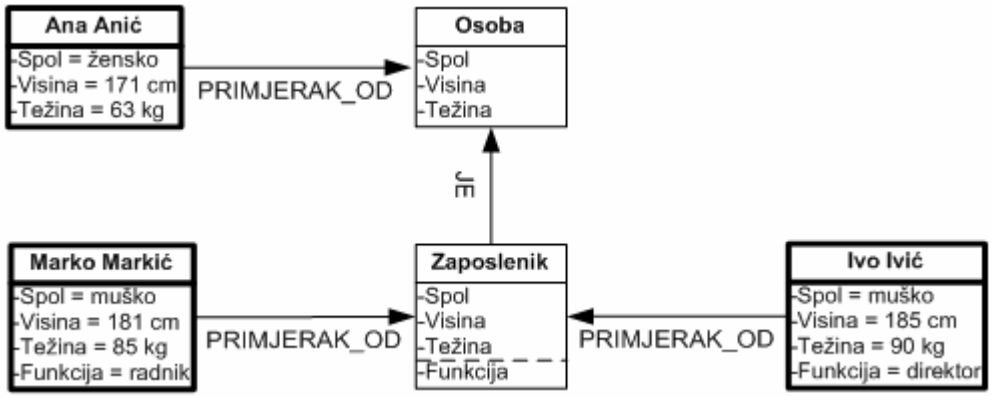
Okviri kod prikaza znanja [MINS1975] služe prilikom unošenja podataka u elemente znanja. Tako se kod semantičkih mreža čvor proširuje s okvirima koji sadrže otvore koji se popunjavaju podacima. Čvor kao element znanja posjeduje okvir čiji otvor predstavlja svojstvo (atribut), a popunjavanjem otvora se unosi sama vrijednost svojstva. Okviri se dijele na generičke i na okvire primjeraka.

- Pogodno je kod onih čvorova koji predstavljaju klasu definirati otvore okvira. Takvi generički okviri sadrže samo otvore bez punjenja.
- Okviri primjeraka će sadržavati punjenja otvora. Svaki primjerak iste klase može sadržavati različito punjenje istog otvora.

Na slici 2.2 čvor osoba koji predstavlja klasu sadrži okvir sa otvorima spol, visina i težina. Punjenja tih okvira sadrže primjerci čvora osoba.

Slika 2.2 Generički okviri i okviri primjerka kod osoba

Generički okviri se nasljeđuju zbog hijerarhijske organiziranosti čvorova u semantičkoj mreži. Veze koje dopuštaju nasljeđivanje svojstava omogućuju podčvoru da sadrži sve okvire sa svojstvima svojih nadčvorova, te se još može proširiti sa nekim dodatnim svojstvima, odnosno okvirom. Okviri primjerka će tada sadržavati punjenja svih naslijeđenih okvira.



Slika 2.3 Nasljeđivanje okvira u semantičkoj mreži

Na slici 2.3 zaposlenik je naslijedio okvir od osobe i još mu je dodan otvor funkcija.

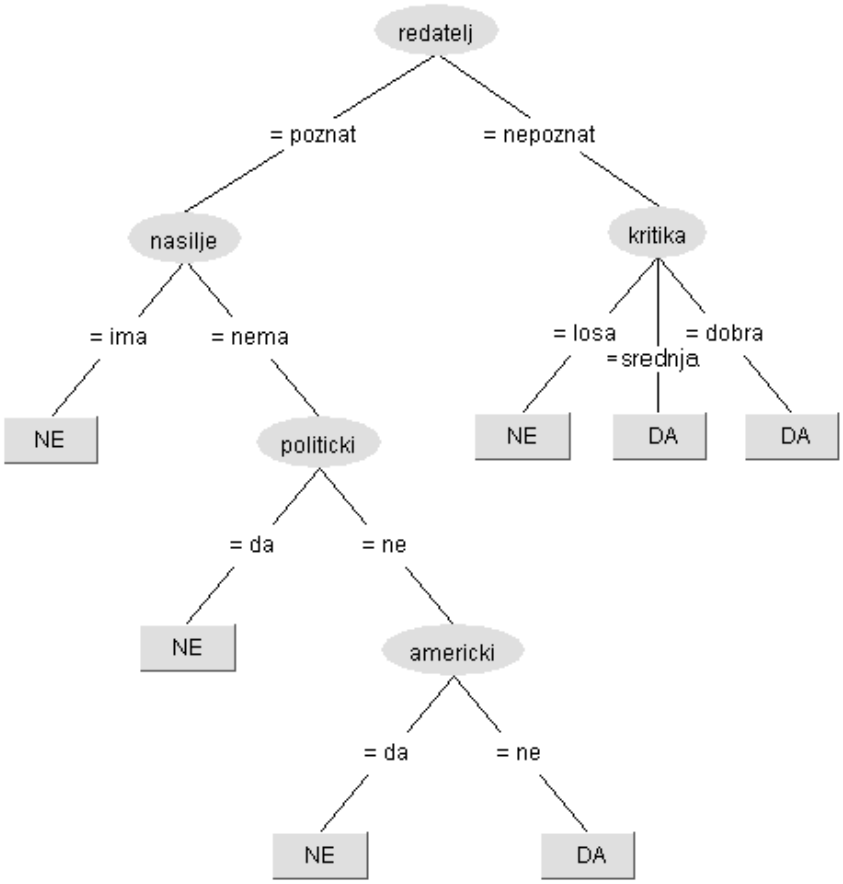
Semantička mreža s okvirima je jedna od metoda prikaza znanja u inteligentnim tutorskim sustavima. U ovom seminaru će poslužit za opisivanje deklarativnog znanja. Nastavni sadržaj koji se izlaže učeniku sadržavat će međusobno povezane čvorove čija će se svojstva unositi pomoću okvira. Prilikom ispitivanja znanja učenika primijenit će se druga metoda prikaza znanja koja će koristiti hijerarhijsko svojstvo semantičke mreže s okvirima.

3 Stabla odlučivanja

Nakon što je opisana metoda prikaza područnog znanja, može se pristupiti procesu ispitivanja tog znanja. Stabla odlučivanja ([QUIN1975], [GIAR1998]) osim što omogućuju prikaz znanja, također služe i kod zaključivanja pomoću znanja. Ona se baziraju na metodi "podijeli pa vladaj" gdje se skup primjera dijeli na podskupove i proces podijele se rekurzivno ponavlja sve dok se ne dođe do jednog primjera koji zapravo vrši klasifikaciju znanja.

Elementi znanja kod stabla odlučivanja su čvorovi i grane. Grane povezuju "roditeljske čvorove" s "dječjim čvorovima". Čvor bez roditelja naziva se "korijenski čvor", a čvorovi bez djece su "listovi". Listovi se još nazivaju "čvorovi odgovora" jer oni predstavljaju sva moguća rješenja zadanog problema. Svi ostali čvorovi su "čvorovi odluke".

Slika 3.1 daje primjer stabla odlučivanja kod određivanja o tome da li ćemo gledati neki film. Čvorovi odgovora su dakle "DA" ili "NE". Čvorovi odlučivanja sadrže pitanja o redatelju, kritici filma, o tome ima li nasilja u filmu itd., dok grane predstavljaju odgovore na pitanja. Ako je redatelj nepoznat onda ćemo ovisno o kritici odlučiti da li ćemo gledati film. U slučaju da je redatelj poznat, tada se odgovara na daljnja pitanja, sve dok ne dođemo do odluke o gledanju filma.



Slika 3.1 Primjer stabla odlučivanja

Na ovaj način smo film klasificirali u filmove koje ćemo gledati i u one koje nećemo gledati. Općenito znanje možemo podijeliti u više od dvije klase, kao što i grane u stablu odlučivanja mogu imati više vrijednosti. Tako filmove možemo podijeliti u odlične, dobre, prosječne i loše, a grane čvora "kritika" mogu imati sljedeće vrijednosti: loša, srednja, dobra. Prvi kriterij za gledanje filma ne mora biti redatelj. Kod odluke o gledanju filma može se prvo gledati da li ima nasilja u filmu ili se može izabrati bilo koji drugi čvor odluke za korijen stabla. Isto tako se može izabrati bilo koji preostali čvor odluke kao korijen podstabla korijenskog čvora. Stablo se može izgraditi na nekoliko načina, odnosno, odluke se mogu nalaziti u bilo kojem čvoru stabla.

Kod procesa izgradnje stabla odlučivanja potrebno je pripremiti skup podataka nad kojima će se vršiti klasifikacija. Kod pripreme se podaci razvrstavaju u tablicu gdje stupci tablice sadrže vrijednosti određenog atributa.

vrijeme	temperatura	vlaga	vjetrovito	igrati
sunčano	toplo	visoka	da	NE
sunčano	toplo	visoka	ne	NE
oblačno	toplo	visoka	da	DA
kišovito	blago	visoka	da	DA
kišovito	hladno	normalna	da	DA
kišovito	hladno	normalna	ne	NE
oblačno	hladno	normalna	ne	DA
sunčano	blago	visoka	da	NE
sunčano	hladno	normalna	da	DA
kišovito	blago	normalna	da	DA
sunčano	blago	normalna	ne	DA
oblačno	blago	visoka	ne	DA
oblačno	toplo	normalna	da	DA
kišovito	blago	visoka	ne	NE

Tablica 3.1 Primjer skupa podataka o vremenu

Tablica 3.1 daje primjer skupa podataka gdje su svi atributi nominalni, odnosno skup vrijednosti atributa je konačan. Nominalni atribut "igrati" ima dvije vrijednosti: DA i NE. Ako je vrijednost atributa "igrati" DA, onda će se igrati na otvorenom terenu. Primjer u skupu podataka predstavlja jedan red u tablici. Jedan od primjera kada se neće igrati na otvorenom terenu kaže; ako je vrijeme sunčano, topla temperatura, visoka vlaga i ako je vjetrovito, onda se neće igrati. Čvorovi kod stabla odlučivanja testirat će vrijednosti atributa i imat će onoliko grana koliko atribut ima različitih vrijednosti. Kod brojčanih atributa skup vrijednosti se dijeli u dva ili više podskupova.

Problem učenja iz skupa podataka ostvaruje se izgradnjom stabla odluke po principu "podijeli pa vladaj". Čvorovi stabla testiraju pojedine atribute. Kod nominalnih atributa se atribut uspoređuje s konstantom ili se primjenjuje neka funkcija za usporedbu. Ovisno o rezultatu usporedbe se bira grana čvora i stablo se dalje gradi rekurzivnim postupkom sve dok se ne dođe do listova stabla koji daju klasifikaciju odabranog primjerka iz skupa podataka.

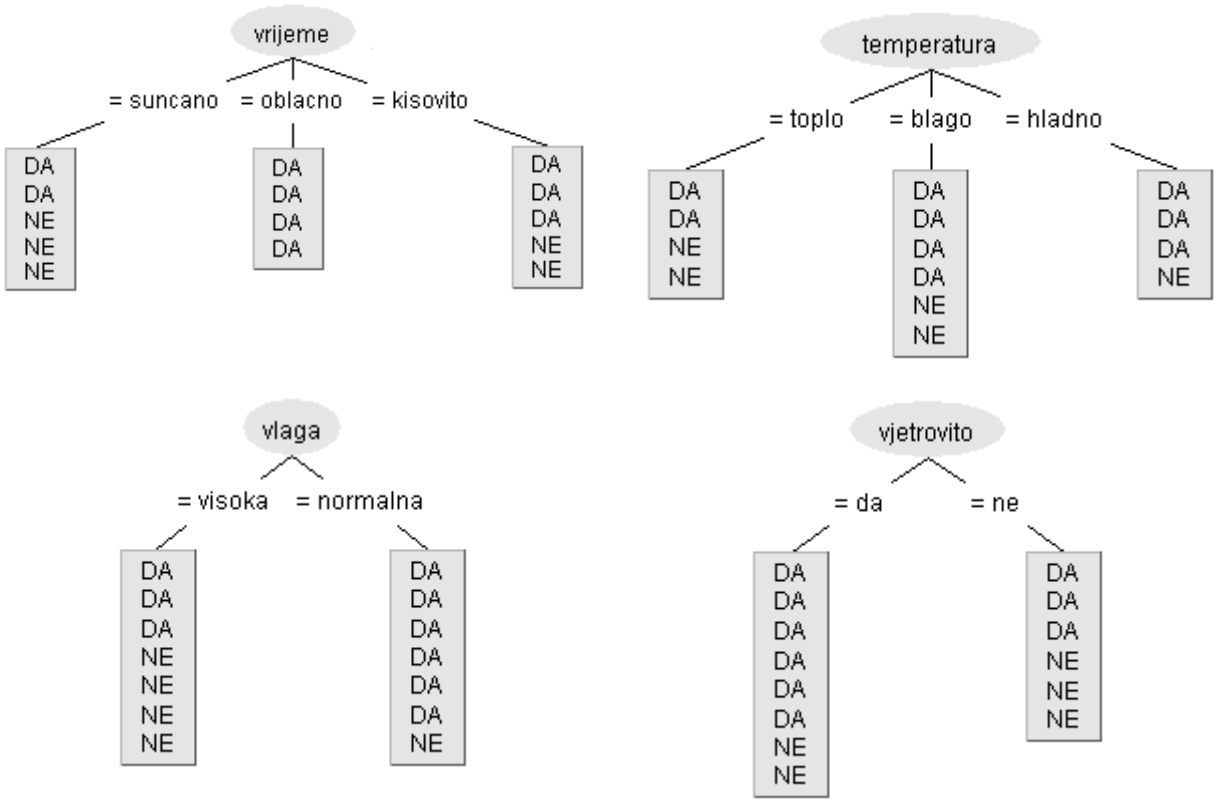
Stablo odlučivanja također vrši klasifikaciju onih primjera koji se ne nalaze u skupu podataka. Zbog toga se može reći kako stablo odlučivanja vrši indukciju znanja.

Postoji nekoliko algoritama za izgradnju stabla odlučivanja, a svi oni se baziraju na ID3 algoritmu. ID3 je jedan od algoritama za klasifikaciju pomoću kojeg se pokušava riješiti problem važnosti atributa u cjelokupnom odlučivanju, odnosno klasificiranju.

3.1 ID3 algoritam

ID3 je najjednostavniji algoritam [QUIN1975] za izgradnju stabla odlučivanja i može se primijeniti kod onog znanja gdje su podaci opisani nominalnim atributima.

Na skupu podataka danom u tablici 3.1 pokazat ćemo izgradnju stabla odlučivanja pomoću ID3 algoritma. Ciljni atribut na osnovu kojeg će se vršiti klasifikacija će biti da odluka da li će se igrati na otvorenom terenu ili ne. Ako je vrijeme sunčano onda imamo 2 primjera igranja na otvorenom i 5 primjera kada se neće igrati na otvorenom terenu. Na slici 3.2 je za svaki atribut i za svaku vrijednost atributa prikazano koliko primjera imaju za ciljni atribut vrijednost DA (igranje na otvorenom terenu), odnosno NE (ne igranje na otvorenom terenu).



Slika 3.2 Podjela stabla po atributima

Prvo je potrebno odabrati atribut koji ćemo staviti u korijen stabla. Za svaki atribut i za svaku vrijednost atributa prebrojava se koliko puta dana grana čvora vodi proces odlučivanja u određenu vrijednost ciljnog atributa.

Primjeri u ovom skupu podataka se dijele u dvije klase, S_{DA} i S_{NE}. Recimo da skup primjera S sadrži d elemenata iz klase S_{DA} i n elemenata iz klase S_{NE}. Po teoriji informacija se količina informacije, potrebna da se odredi pripadnost proizvoljnog primjerka iz skupa podataka S nekoj od klasa S_{DA} ili S_{NE}, računa se po formuli:

$$I(d,n) = -\frac{d}{d+n}\log_2\frac{d}{d+n} - \frac{n}{d+n}\log_2\frac{n}{d+n} \tag{1}$$

Formula (1) predstavlja informacijsku vrijednost kojom se mjeri koliko dobro dani atribut razdvaja primjerke prema njihovoj klasifikaciji. Informacijska vrijednost će biti 0 ako svi primjerci pripadaju istoj klasi. Recimo da skup S ima 5 primjera i svih 5 pripadaju klasi S_{DA}, odnosno 0 primjera se nalazi u S_{NE}. Tada će informacijska dobit iznositi 0. Informacijska vrijednost će biti maksimalna samo u slučaju kada skup S sadrži istovjetan broj primjera u svakoj klasi.

U teoriji informacije se mjera čistoće, koju ćemo koristiti prilikom određivanja homogenosti skupa primjera, naziva prosječna informacijska vrijednost. Pretpostavit ćemo da se skup podataka S u odnosu na atribut A dijeli u m podskupova $S_i, i \in \{1,2,...,m\}$. Ako skup S_i sadrži d_i primjera koji pripadaju klasi S_{DA} i n_i primjera koji pripadaju klasi S_{NE}, tada se prosječna informacijska vrijednost atributa A potrebna za klasifikaciju primjera u sve podskupove S_i dobiva po formuli:

$$I(A) = \sum_{i=1}^{m}\frac{d_i + n_i}{d+n} I(d_i, n_i) \tag{2}$$

Uz izračunatu prosječnu informacijsku vrijednost atributa A potrebno je definirati mjeru efektivnosti od A u odnosu na dani skup podataka S. Informacijski dobitak (gain) je mjera kojom se određuje očekivana redukcija prosječne informacijske vrijednosti atributa A uzrokovana razdvajanjem primjera na osnovu tog atributa, odnosno:

$$D(S,A) = I(d,n) - I(A) \tag{3}$$

Na osnovu podjele stabla odlučivanja po atributima na slici 3.2 prikazat ćemo izbor korijenskog atributa. U slučaju atributa vrijeme imamo sljedeće informacijske vrijednosti u njegovim listovima:

$$I(2,3) = 0.971$$
$$I(4,0) = 0$$
$$I(3,2) = 0.971$$

Prosječna informacijska vrijednost atributa vrijeme se sada mjeri po formuli (2):

$$I(vrijeme) = \frac{2+3}{9+5}0.971 + \frac{4+0}{9+5}0 + \frac{3+2}{9+5}0.971 = 0.693$$

U odnosu na cijeli skup podataka informacijska dobit se mjeri po (3):

$$D(vrijeme) = I(9,5) - I(vrijeme) = 0.940 - 0.693 = 0.247$$

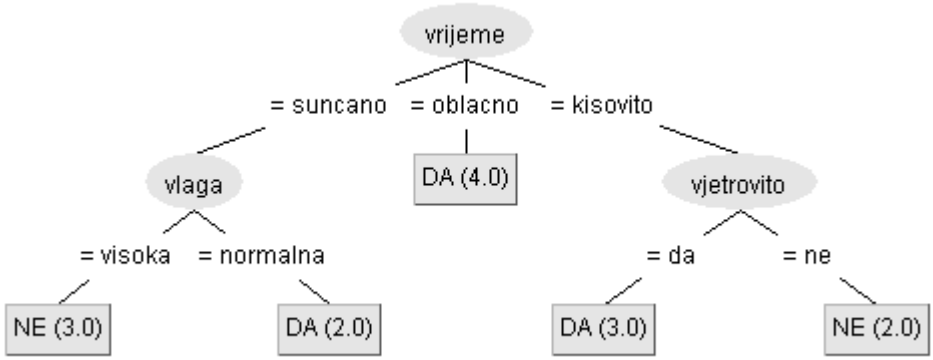
Na isti način ćemo izračunat informacijsku dobit za preostala 3 atributa:

D(temperatura)=0.029

D(vlažnost)=0.152

D(vjetrovito)=0.048

Atribut vrijeme ima najveću informacijsku dobit i zbog toga će on biti u korijenu stabla. Nakon toga za svaku granu atributa vrijeme se postupak rekurzivno ponavlja dok se ne dobije stablo kao na slici 3.3.



Slika 3.3 Stablo odlučivanja

ID3 algoritam se zaustavlja kada svi atributi budu iskorišteni u nekoj grani stabla, ili dok svi primjerci koji pripadaju čvoru imaju istu klasu, tj. kad je informacijska dobit primjerka 0. Na slici 3.4 je dan pseudokod ID3 algoritma izrade stabla odlučivanja.

```
funkcija ID3(R,C,S) vraća stablo odlučivanja;
Ulazi: R: skup nezavisnih atributa;
       C: ciljni(zavisni) atribut;
       S: skup primjera za učenje;
počni
       Ako je  S prazan, tada napravi jedan čvor s vrijednošću Pogrešno;
       Ako se S sastoji od primjera s istom vrijednošću ciljnog atributa
       tada napravi jedan krajnji čvor s tom vrijednosti ciljnog atributa;
       Ako je R prazan, tada napravi jedan (krajnji) čvor s vrijednosti koja je najčešća za
       ciljni atribut za skup S; (u tom slučaju stablo će vjerojatno raditi i pogrešne
       klasifikacije na skupu S, u mjeri u kojoj su zastupljeni primjeri ostalih klasa);
       Neka je A atribut s najvećom vrijednosti Dobiti(A) između svih atributa u R;
```

```
        Neka su {aj| j=1,2, .., m} vrijednosti atributa A;
        Neka su {Sj| j=1,2, .., m} podskupovi S koji se sastoje od primjera koji imaju aj za
        atribut A;
        Napravi stablo s korijenom označenim A, te granama a1, a2, ..., am koja vode na stabla
        (ID3(R-{A}, C, S1), ID3(R-{A}, C, S2),.....,ID3(R-{A}, C, Sm);
        Rekurzivno primjeni ID3 to na podskupove {Sj| j=1,2, .., m} sve dok oni nisu prazni;
kraj
```

Slika 3.4 ID3 algoritam

3.2 Problem višegranajućih atributa

Kod računanja količine informacije potrebne da se specificira klasa primjera uvodi se entropija. Entropija kao mjera čistoće informacije kod klasificiranja skupa podataka na osnovi atributa *A* dobiva se po formuli

$$E(A) = -\sum_{i=1}^{m} p_i \log_2 p_i \qquad (4)$$

gdje su p_i vjerojatnosti da se primjer nađe u poskupu S_i. Atribut *A* dijeli skup *S* na *m* disjunktnih podskupova $S_i, i \in \{1,..,m\}$.

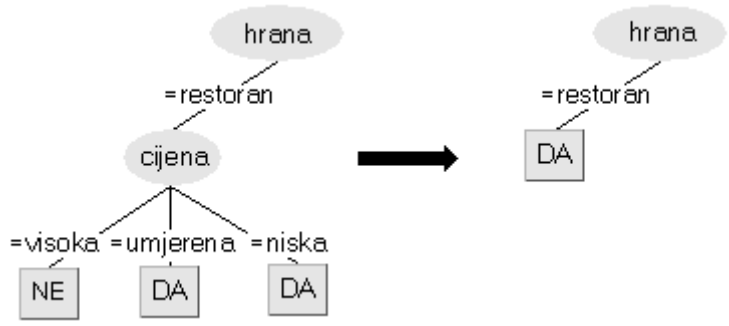
U slučaju kada atribut ima veliki broj mogućih vrijednosti i kada ima drugačiju vrijednost za svaki primjer iz skupa podataka, tada će entropija u tom čvoru biti 0. Takva vrsta atributa se zove identifikacijski atribut i on bi prvi trebao izvršiti klasifikaciju primjera, tj. trebao bi se nalaziti u korijenskom čvoru. Grananje na osnovu identifikacijskog atributa nije pogodno za otkrivanje novih primjera koji se ne nalaze u skupu podataka i nije pogodan za stavljanje u korijen stabla odlučivanja. Zbog toga je uvedena modifikacija mjere dobiti količine informacija. Faktor dobitka (gain ratio) se dobiva kao omjer informacijske dobiti i ukupne informacijske vrijednosti atributa A.

$$FD(A) = \frac{D(A)}{E(A)} \qquad (5)$$

Kod faktora dobitka se u obzir uzimaju i veličina čvorova djece. U slučaju kad je entropija atributa vrlo mala, onda faktor dobitka može biti vrlo velik. Tada se kod izrade stabla odlučivanja izabiru oni atributa čiji je faktor dobitka najveći, ukoliko je entropija tog atributa veća ili jednaka srednjoj vrijednosti entropija svih atributa.

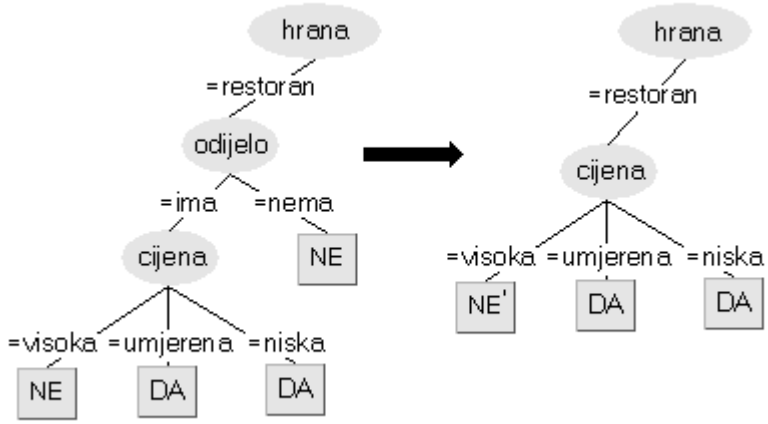
3.3 Smanjivanje stabla odlučivanja

Radi izbjegavanja rasta stabla koriste se algoritmi za smanjivanje unatrag i za smanjivanje unaprijed. Postoje dvije metode smanjivanja unatrag: zamjena podstabla i podizanje podstabla. Primjer smanjivanja unatrag koji vrši zamjenu podstabla dan je na slici 3.5. Cijelo podstablo s korijenskim čvorom "cijena" je zamijenjeno jednim jedinim listom. Očito je da se time smanjila preciznost kod pripremnog skupu podataka, ali će se zato primjeri iz nezavisnog skupa podataka bolje i brže klasificirati.



Slika 3.5 Smanjivanje unatrag zamjenom podstabla

Podizanje podstabla kod smanjivanja unatrag koriste popularni C4.5 algoritmi [QUIN1993]. Na slici 3.6 se cijelo podstablo s korijenskim čvorom "cijena" približilo korijenu stabla. List "NE" čvora "odijelo" se tada prekvalificirao u novo podstablo s korijenskim čvorom "cijena".



Slika 3.6 Smanjivanje unatrag podizanjem podstabla

Kod smanjivanja stabla unaprijed prvo se odredi pravilo kada će se prestati dodavati podstabla. Ta se metoda pokazala teško izvedivom u praksi, pa se ne primjenjuje u većini

algoritama. Važno je naglasiti kako se smanjivanjem stabla otvara prostor greškama u procesu odlučivanja.

3.4 Procjena pogreške

Izbor metode smanjivanja stabla unatrag ovisit će procjeni pogreške. Odluka o tome da li će se čvor zamijeniti, ovisi o usporedbi količine pogreške čvora zajedno s kombiniranom količinom pogreške njegove djece. Čvor će se zamijeniti ako je količina pogreške manja od kombinirane pogreške njegove djece. Za svih N primjera koji dosežu neki čvor neka je E broj primjera koji pripadaju klasi s najvećim brojem primjera. Neka je $f = \frac{E}{N}$ procjena mjera pogreške, a q stvarna mjera pogreške. Određivanje granice pouzdanosti na osnovu faktora pouzdanosti c se određuje Bernoulijevim procesom:

$$\Pr\left(-z < \frac{f-p}{\sqrt{p(1-p)/N}} < z\right) = c \tag{6}$$

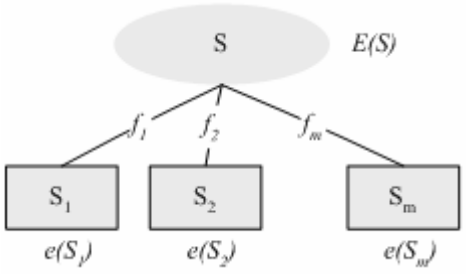
C4.5 algoritam koristi faktor pouzdanosti c=0.25% što daje granicu pouzdanosti z=0.69. Procjena iznosa pogreške računa se po formuli:

$$e = \frac{f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}} \tag{7}$$

Neka se skup primjera S dijeli u m disjunktnih klasa S_i po svakom listu kao što je prikazano na slici 3.7. Iznos procjene pogreške $e(S_i)$ se za svaki list računa po formuli (7) gdje su f_i omjeri broja primjera koji pripadaju skupu S_i i N primjera skupa S. Kombinacija količine pogreške listova se određuje po formuli:

$$e(S_1, S_2, .., S_m) = \sum_{i=1}^{m} f_i \, e(S_i) \tag{8}$$

Za procjenu iznosa pogreške čvora se uzima minimum skupa $\{e(S), e(S_1,..,S_m)\}$. Ako je količina pogreške čvora E(S) manja od kombinirane količine pogrešaka listova $e(S_1,..,S_m)$ tada će se čvor izbaciti.



Slika 3.7 Procjena iznosa pogreške čvora i njegovih listova

4 Upotreba stabla odlučivanja kod klasificiranja znanja u semantičkoj mreži s okvirima

Prikaz područnog znanja pomoću semantičke mreže s okvirima poslužit će kao element nastavnog sadržaja koji se prezentira učeniku. Učenik nakon procesa učenja pristupa procesu testiranja stečenog znanja. Inteligentni tutorski sustav će na osnovu područnog znanja generirati pitanja i ponuditi odgovore na izbor učeniku. Samo testiranje učenika pomoću skupa pitanja i ponuđenih odgovora predstavlja kviz koji se zasniva na procesu otkrivanja znanja u skupovima podataka.

Otkrivanje znanja u skupovima podataka je proces pronalaženja potencijalno korisnih novih informacija iz zadanog znanja. Definira se kao aktivnost otkrivanja informacija čiji je cilj otkrivanje skrivenih činjenica iz skupa podataka. Proces otkrivanja znanja kao vrsta obrade podataka ima sljedeće korake:

1. Razumijevanje problema
2. Razumijevanje podataka
3. Priprema podataka
4. Modeliranje
5. Evaluacija modela
6. Primjena

Svi navedeni koraci će biti pokazani na primjeru kviza nad područnim znanjem o automobilima.

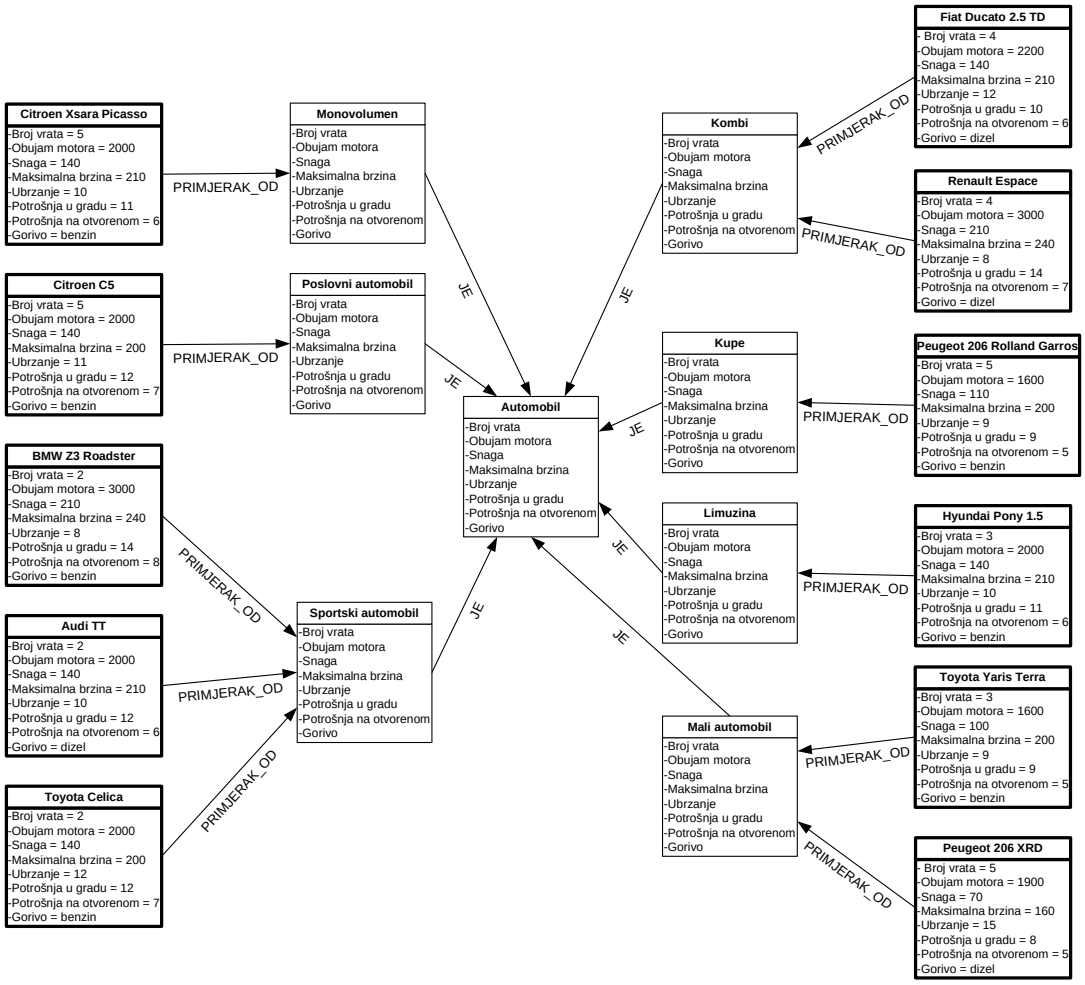
4.1 Razumijevanje problema

Automobile možemo općenito svrstati u nekoliko osnovnih kategorija, kao što su: sportski automobil, kombi, poslovni automobil, kupe, itd. Da bi što preciznije opisali automobil navest ćemo i neka njegova svojstva za koja smatramo da su bitna. Primjeri svojstava

automobila su broj vrata, obujam motora, snaga, maksimalna brzina, ubrzanje, potrošnja goriva u gradu, potrošnja goriva na otvorenom i vrsta goriva kojeg upotrebljava.

Na slici 4.1 je dana semantička mreža s okvirima skupa primjeraka automobila razvrstanih u nekoliko osnovnih kategorija. Znanje prikazano na ovakav način omogućuje učeniku usvajanje deklarativnog znanja o vrstama automobila i njihovih svojstava, primjerice, on može naučiti da je Toyota Yaris Terra vrsta malog automobila koji upotrebljava benzinsko gorivo i ima maksimalnu brzinu 200 km/h.

Nakon učenja obično slijedi proces provjere znanja. Kviz je jedna od metoda provjere znanja u inteligentnim tutorskim sustava gdje se pitanja unose od strane učitelja ili generiraju na osnovu povezanosti elemenata iz područnog znanja.



Slika 4.1 Prikaz znanja o automobilima pomoću semantičke mreže s okvirima

Za primjere automobila gledat ćemo pitanja koja se generiraju po ovoj shemi: "Ako je brzina automobila 170 km/h i upotrebljava dizel gorivo, da li je to sportski automobil?" ili "Ako je

snaga automobila 60 kW, potrošnja u gradu 8 litara i obujam motora 1500 cm^3, kojoj vrsti onda pripada?".

Primjerice, automobil koji ima maksimalnu brzinu 170 km/h i upotrebljava dizel gorivo može i ne mora biti opisan u semantičkoj mreži s okvirima. Zbog toga je taj primjer automobila nezavisan u odnosu na područno znanje o automobilima, ali se ipak može pridijeliti nekoj od postojećih kategorija. Indukcijom znanja se učenika uči razvrstavanju znanja po analogiji bez obzira što se do tada nikad nije susreo sa stvarnim znanjem.

Cilj nam je razvrstati znanje o automobilima u kategorije i zato je potrebno upotrijebiti neki od klasifikacijskih algoritama, recimo stabla odlučivanja.

4.2 Razumijevanje i priprema podataka

Podaci koji će uči u obradu moraju se prethodno razumjeti. Pod razumijevanjem se smatra određivanje atributa koji opisuju primjere i popunjavanjem vrijednosti atributa za svaki primjer. Na slici 4.1 se vidi kako čvor automobil sadrži okvir s otvorima koji opisuje svojstva automobila. Očito je onda da će otvori predstavljati atribute skupa podataka. Ti otvori kod primjeraka automobila imaju i svoja punjenja koja su zapravo vrijednosti atributa. Automobili će se razvrstavati u kategorije, odnosno, potrebno je imati ciljni atribut "vrsta" koji će sadržavati vrijednosti kao što su: kombi, kupe, poslovni automobil i tako dalje. Identifikacijski atribut "model" sadržavat će za svoje vrijednosti sve primjerke automobila. Tablica 4.1 prikazuje atribute i skup podataka o automobilima. Radi jednostavnosti pretpostavit ćemo da su svi atributi nominalni, odnosno imaju konačan skup vrijednosti.

Model	broj vrata	obujam motora	snaga	maksimalna brzina	ubrzanje	potrošnja grad	potrošnja otvoreno	Gorivo	vrsta
Audi TT	2	2000	140	210	10	12	6	Dizel	sportski
BMW Z3	2	3000	210	240	8	14	8	Benzin	sportski
Citroen C5	5	2000	140	200	11	12	7	Benzin	poslovni
Citroen Xsara Picasso	5	2000	140	210	10	11	6	Benzin	monovolumen
Fiat Ducato 2.5 TD	4	2200	140	210	12	10	6	Dizel	kombi
Hyundai Pony 1.5	3	2000	140	210	10	11	6	Benzin	limuzina
Peugeot 206 RG	5	1600	110	200	9	9	5	Benzin	kupe
Peugeot 206 XRD	5	1900	70	160	15	8	5	Dizel	mali
Renault Escape	4	3000	210	240	8	14	7	Benzin	kombi
Toyota Celica	2	2000	140	200	12	12	7	Benzin	sportski
Toyota Yaris Terra	3	1600	110	200	9	9	5	Benzin	mali

Tablica 4.1 Skup podataka o automobilima

Pripremanjem podataka stvara se skup podataka koji će predstavljati ulaz u proces modeliranja. Samo modeliranje će napraviti inačica C4.5 algoritma implementirana u programskom paketu WEKA razvijenog na sveučilištu Waikato na Novom Zelandu [WITF1999]. Programski paket WEKA ima poseban ARFF format datoteka za spremanje skupova podataka. Rezultat pretvorbe podataka iz tablice 4.1 u ARFF format prikazan je na slici 4.2.

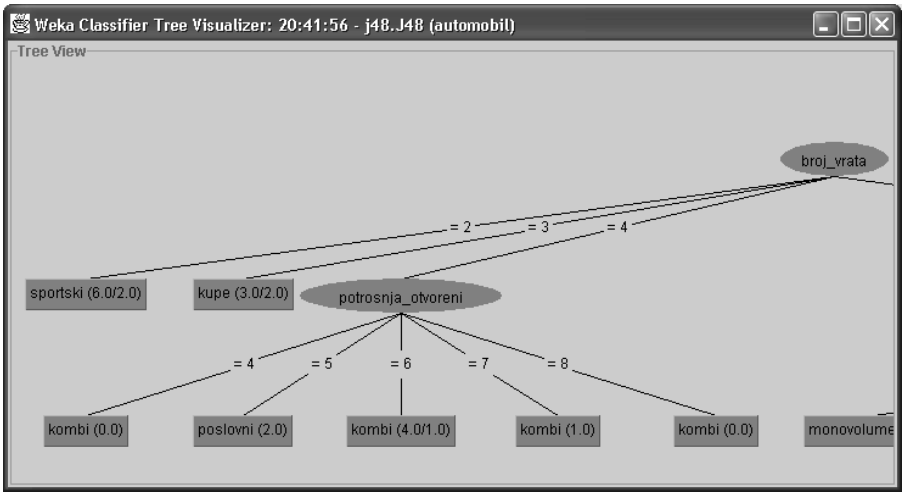
```
@relation automobil

@attribute broj_vrata {2, 3, 4, 5}
@attribute obujam_motora {1100, 1400, 1500, 1600, 1800, 1900, 2000, 2200, 3000}
@attribute snaga {60, 70, 80, 90, 110, 120, 140, 160, 210}
@attribute max_brzina {160, 170, 180, 190, 200, 210, 220, 230, 240}
@attribute ubrzanje {8, 9, 10, 11, 12, 13, 14, 15, 17, 19}
@attribute potrosnja_grad {6, 7, 8, 9, 10, 11, 12, 13, 14}
@attribute potrosnja_otvoreni {4, 5, 6, 7, 8}
@attribute gorivo {benzin, dizel}
@attribute vrsta {kombi, kupe, limuzina, mali, monovolumen, poslovni, sportski}

@data
2,2000,140,210,10,12,6,dizel,sportski
2,3000,210,240,8,14,8,benzin,sportski
5,2000,140,200,11,12,7,benzin,poslovni
5,2000,140,210,10,11,6,benzin,monovolumen
4,2200,140,210,12,10,6,dizel,kombi
3,2000,140,210,10,11,6,benzin,limuzina
5,1600,110,200,9,9,5,benzin,kupe
5,1900,70,160,15,8,5,dizel,mali
4,3000,210,240,8,14,7,benzin,kombi
2,2000,140,200,12,12,7,benzin,sportski
3,1600,110,200,9,9,5,benzin,mali
```

Slika 4.2 prikaz skupa podataka u ARFF formatu

ARFF datoteka podržava nominalne i numeričke atribute. Uz ime atributa se kod nominalnih vrijednosti definira i konačan skup vrijednosti koje oni mogu poprimiti. Nakon definicije atributa slijede podaci o primjerima.

4.3 Modeliranje i evaluacija modela

U tablici 4.1 se nalazi samo jedan manji podskup primjera automobila. Modeliranje će se vršiti na skupu od 34 različitih primjera automobila jer broj primjeraka mora biti značajno veći od broja različitih kategorija. Za tehniku klasifikacije skupa podataka u procesu modeliranja znanja izabrat ćemo prethodno opisana stabla odlučivanja. Opisani skup podataka o automobila je otvoren unutar programskog paketa WEKA i izabran je J48 algoritam koji je inačica C4.5 klasifikatora. Stablo odlučivanja nastalo primjenom J48 klasifikatora dano je na slikama 4.3 i 4.4.



Slika 4.3 Prikaz stabla odlučivanja pomoću programskom paketu WEKA

Zbog veličine stabla odlučivanja je na slici 4.4 dan tekstualni prikaz stabla. Pokraj svakog lista nalazi se broj primjeraka koji ulaze u tu klasu zajedno sa iznosom procjene pogreške.

```
broj_vrata = 2: sportski (6.0/2.0)
broj_vrata = 3: kupe (3.0/2.0)
broj_vrata = 4
|   potrosnja_otvoreni = 4: kombi (0.0)
|   potrosnja_otvoreni = 5: poslovni (2.0)
|   potrosnja_otvoreni = 6: kombi (4.0/1.0)
|   potrosnja_otvoreni = 7: kombi (1.0)
|   potrosnja_otvoreni = 8: kombi (0.0)
broj_vrata = 5
|   ubrzanje = 8: monovolumen (0.0)
|   ubrzanje = 9: kupe (1.0)
|   ubrzanje = 10: monovolumen (2.0)
|   ubrzanje = 11: limuzina (4.0/2.0)
|   ubrzanje = 12
|   |   obujam_motora = 1100: monovolumen (0.0)
|   |   obujam_motora = 1400: monovolumen (0.0)
|   |   obujam_motora = 1500: monovolumen (0.0)
|   |   obujam_motora = 1600: monovolumen (4.0)
|   |   obujam_motora = 1800: monovolumen (0.0)
|   |   obujam_motora = 1900: monovolumen (0.0)
|   |   obujam_motora = 2000: mali (2.0)
|   |   obujam_motora = 2200: monovolumen (0.0)
|   |   obujam_motora = 3000: monovolumen (0.0)
|   ubrzanje = 13: poslovni (1.0)
|   ubrzanje = 14: monovolumen (3.0)
|   ubrzanje = 15: limuzina (1.0)
|   ubrzanje = 17: monovolumen (0.0)
|   ubrzanje = 19: monovolumen (0.0)
```

Slika 4.4 Stablo odlučivanja

4.4 Primjena rezultata modela

Skup atributa koji čine čvorove u stablu odlučivanja su: broj vrata, potrošnja u gradu, potrošnja na otvorenom, ubrzanje i maksimalna brzina. Oni će biti elementi koji će ući u proces generiranja pitanja. Kreiranje pitanja zahtijeva izbor neke od staza u stablu

odlučivanja, odnosno izbor atributa i vrijednosti koji vode do ciljne vrijednosti. Na primjer, ako izaberemo automobil s 5 vrata onda dolazimo do čvora ubrzanje i slučajno biramo neku od vrijednosti, npr. 12 m/s^2. Ovim putom smo došli do čvora obujam motora i izborom grane s vrijednošću 1800 cm^3, doći ćemo do monovolumena.

Na osnovu ovog puta u stablu odlučivanja možemo generirati sljedeće dvije vrste pitanja:

1. Ako automobil ima 5 vrata, ubrzanje 12 m/s^2 i obujam 1800 cm^3, da li je to poslovni auto?
2. Ako automobil ima 5 vrata, ubrzanje 12 m/s^2 i obujam 1800 cm^3, koja je to vrsta automobila?

Inteligentni tutorski sustav će kod testiranja učenika metodom kviza koristiti i ovakvu vrstu pitanja. Kao što je ranije rečeno, testiranjem na pitanjima nastalih korištenjem stabla odlučivanja može se provjeriti znanje koje izlazi iz primjera opisanih semantičkom mrežom s okvirima. Primjerice, ako imamo automobil koji nije opisan u danom skupu podataka

5 Zaključak

Inteligentni tutorski sustavi mogu imati različite metode za prikaz znanja. Semantička mreža s okvirima kao jedna od metoda prikaza znanja je u ovom seminaru korištena za izgradnju deklarativnog znanja. Različiti tipovi veza semantičke mreže poslužili su za iskazivanje hijerarhijskog odnosa među pojmovima, strukturalnu pripadnost pojma nekom pojmu i ostale relacijske veze među pojmovima. Nadređeni i podređeni pojmovi vezivali su se vezama koje dopuštaju nasljeđivanje svojstava. Otvori u okvirima poslužili su za definiranje svojstava pojma, a punjenja istih okvira imaju oni pojmovi koji predstavljaju primjerke pojmova sa svojstvima.

Za testiranje, u inteligentnim tutorskim sustavima, se najčešće koristi metoda kviza. Pitanja u kvizu se u većini ovakvih sustava unose od strane učitelja. Prikaz znanja pomoću semantičke mreže s okvirima omogućuje generiranje pitanja od strane računala. Jedna vrsta pitanja koja služe za klasificiranje znanja mogu se generirati korištenjem stabla odlučivanja kao jedne od metoda prikaza znanja. Takva pitanja u svojoj strukturi koriste svojstva i vrijednosti svojstava pojmova na osnovu kojih se odgovaranje na pitanje pojam svrstava u neku od klasa pojmova.

Prednost ovakvog ispitivanja znanja, osim što se pitanja generiraju od strane računala, je u tome što se mogu klasificirati i pojmovi koji nisu opisani u semantičkoj mreži s okvirima.

6 Literatura

[FIRE1988] M. W. Firebaugh: "Artificial Intelligence: A Knowledge-Based Approach", PWS Publishers / Boyd & Fraser, 1988.

[PARC1988] K. Parsaye, M. Chignell: "Expert systems for experts", New York, John Wiley & Sons, Inc., 1988.

[QUIL1968] M.R. Quillian: "Semantic memory", in M. Minsky (ed.):, Semantic Information Processing, MIT Press, 1968.

[ANAS1998] A. Analyti, N. Spyratos, P. Constantopoulos: "On the Semantics of a Semantic Network", Fundamenta Informaticae, Vol. 36, No. 2-3, pp. 109-144, 1998.

[MINS1975] M. Minsky: "A framework for representing knowledge" in P. Winston (ed.):, In The Psychology of Computer Vision, McGraw-Hill, New York, pp. 211-277, 1975.

[GIAR1998] J. Giarratano, G. Riley: 1998, "Expert Systems: Principles and Programming", (3 ed.), Boston: PWS Publishing, 1998.

[QUIN1975] J. Ross Quinlan: "Induction of decision trees", in "Machine Learning", Vol. 1, No. 1.,pp. 81-106, 1975.

[QUIN1993] J. R. Quinlan: "C4.5: Programs for Machine Learning", Morgan Kauffmann, Los Altos, CA, 1993.

[STAN1997] S. Stankov: "Izomorfni model sustava kao osnova računalom poduprtog poučavanja načela vođenja", doktorska disertacija, Fakultet elektrotehnike, strojarstva i brodogradnje, Sveučilište u Splitu, Split, 1997.

[WITF1999] I. H. Witten, E. Frank: " Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kauffmann, 1999. http://www.cs.waikato.ac.nz/ml/weka/