Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Seminarski Rad
Klasifikacija teksta pomoću K-NN algoritma

Domagoj Tominac
Mentor: Doc.dr.sc. Bojana Dalbelo Bašić
Ak. godina 2003/2004

Sažetak: Klasifikacija teksta u predefinirani broj kategorija korištenjem K-NN algoritma.

Ključne riječi: Klasifikacija, K-NN, Fuzzy

Zagreb, rujan 2004.

Sadržaj

Uvod

Tema ovog seminarskog rada je klasifikacija teksta prema njegovom sadržaju u određeni broj već definiranih kategorija. Algoritam pomoću kojeg se obavlja klasifikacija teksta je K-najbližih susjeda (K-NN).

Ideja ovog algoritma je klasificirati neklasificirani dokument, u jednu od definiranih kategorija, na temelju predočenih klasificiranih primjera. Metoda pretpostavlja da se svi dokumenti mogu prikazati kao vektor u Euklidskom prostoru. Definira se skup klasificiranih primjera za učenje od dokumenata kojima je poznata kategorija. Rezultat klasifikacije najviše ovisi o primjerima iz ovoga skupa posto se algoritam zasniva na pronalaženju najsličnijeg dokumenta i zato je vrlo bitno kako odabrati (definirati) ovaj skup.

Najveći nedostatak ovog algoritma, kao i svih algoritama temeljenih na nekom skupu za učenje je velika količina računalnog vremena potrebna za klasifikaciju. Drugi je nedostatak što se klasifikacija obavlja uzimajući u obzir sve atribute (rijeci), tako da stvarno slični dokumenti mogu biti dosta udaljeni u vektorskom prostoru.

Algoritam K – najbližih susjeda

Metoda K – najbližih susjeda spada u metode koje se zasnivaju na pronalaženju najsličnijih dokumenata dokumentu kojeg želimo klasificirati. Algoritam podrazumijeva da se svaki dokument može prikazati kao vektor u n – dimenzionalnom prostoru.

Prikaz dokumenata

Prije same klasifikacije potrebno je dokumente, koji su zapisani kao niz znakova, prebaciti u oblik koji je pogodan za računanje. Svaki dokument se predočava kao vektor u *n* – dimenzionalnom vektorskom prostoru. Broj dimenzija *n* se definira kao broj različitih riječi u cijelom skupu uzoraka (dokumenata). Na taj način tvorimo matricu dokument – riječ, čije dimenzije odgovaraju umnošku broja dokumenata i broja riječi u cijelom skupu uzoraka. Svaki element matrice predstavlja težinu riječi a_{ij}, gdje je a_{ij} težina riječi *i* u dokumentu *j*.

Postoji nekoliko načina izračunavanja te težine:

1. Najjednostavnija metoda prikazivanja težina je binarna metoda:

$$a_{ij} = \begin{cases} 0 & \text{ako se riječ nije pojavila u dokumentu} \\ 1 & \text{ako se riječ pojavila u dokumentu} \end{cases}$$

2. Težina predstavlja broj pojavljivanja riječi u nekom dokumentu:

$$a_{ij} = f_{ij}$$

   Gdje je f_{ij} frekvencija (broj pojavljivanja) riječi *i* u dokumentu *j*.

3. Ako tekstualni dokumenti imaju različite dužine, gornja formula se normalizira:

$$a_{ij} = \frac{f_{ij}}{\sqrt{\sum_{s=1}^{n} (f_{sj})^2}}$$

   Gdje je *n* broj različitih riječi u skupu uzoraka.

4. Tf-idf metoda određivanja težina (eng. Term frequency – inverse document frequency)

$$a_{ij} = tfidf(a_{ij}) = f_{ij} \times \log(\frac{N}{n_i})$$

gdje je N – broj dokumenata, n_i broj dokumenata u kojima se riječ *i* pojavljuje barem jedanput.

5. Ako tekstualni dokumenti imaju različite dužine, gornja formula se normalizira na sljedeći način:

$$a_{ij} = \frac{f_{ij}}{\sqrt{\sum_{s=1}^{n} (tfidf(a_{sj}))^2}} \times \log(\frac{N}{n_i})$$

Algoritam

Algoritam podrazumijeva mogućnost prikazivanja svih dokumenata u n – dimenzionalnom prostoru, gdje se udaljenost između dva dokumenta definira kao Euklidska udaljenost.

$$d(x, y) = \sqrt{\sum_{r=1}^{n} (a_{rx} - a_{ry})^2}$$

Gdje je n – broj različitih riječi u skupu svih uzoraka (dokumenata), dok su a_{rx} i a_{ry} težine koje ima riječ *r* u dokumentu *x* odnosno u dokumentu *y*.

Prilikom klasifikacije definira se broj K, koji označava koliko se najbližih susjeda traženom dokumentu pronalazi. Najbliži dokument (susjed) je onaj koji ima najmanju udaljenost do neklasificiranog uzorka. Dokument koji se klasificira pridodjeliti će se onoj kategoriji kojoj pripada najviše od K najbližih susjeda.

Fuzzy Algoritam K – najbližih susjeda

Fuzzy k – nn algoritam je proširenje standardnog k – nn algoritma. Kod fuzzy k – nn algoritma se svakom dokumentu pridodjeljuje neizraziti skup kojim se definira pripadnost pojedinoj kategoriji, za razliku od standardnog k – nn algoritma gdje se pojedinom dokumentu pridružuje klasičan skup pripadnosti kategorija (pripada samo jednoj kategoriji u potpunosti).

Kod stvaranja skupa primjera potrebno je inicijalno iz klasičnog skupa preslikati pripadnost u neizraziti skup. Postoje različiti načini tog preslikavanja. U ovom seminaru korištena su dva načina preslikavanja. Prvi je proširenje klasičnog skupa na neizraziti, na način da se kategoriji kojoj dokument pripada pridruži 1, a kategorijama kojima ne pripada 0, tj. po formuli:

$$\mu_i(y) = \begin{cases} 1 & \text{za} \quad j = i \\ 0 & \text{za} \quad j \neq i \end{cases}$$

gdje μ_i označava stupanj pripadnosti dokumenta y kategoriji i, dok j označava kategoriju kojoj pripada dokument.

Kod drugog načina, pretpostavimo da postoji N kategorija, definirajmo n_i kao broj dokumenata koji pripadaju pojedinoj (i-toj) kategoriji , tada se skup pripadnosti definira:

$$\mu_i(y) = \begin{cases} 0.51 + (n_i/\mathrm{S}) * 0.49 & \text{za} \quad j = i \\ (n_i/\mathrm{S}) * 0.49 & \text{za} \quad j \neq i \end{cases}$$

gdje je

$$S = \sum_{i=1}^{N} n_i$$

Nakon definiranja skupa pripadnosti za sve dokumente u skupu primjera (uzoraka), postupak klasifikacije obavlja se traženjem K – najbližih susjeda i izračunavanjem skupa pripadnosti za novi dokument po sljedećoj formuli:

$$\mu_i(x) = \frac{\sum_{j=1}^{K} \mu_i(y_j)(1/\left\|x - y_j\right\|^{2/(m-1)})}{\sum_{j=1}^{K} (1/\left\|x - y_j\right\|^{2/(m-1)})}$$

gdje je K broj susjeda, μ_i stupanj pripadnosti dokumenta x (ili y) kategoriji i, **m** je parametar između 1 i 2, dok y_j označava dokument iz skupa K najbližih susjeda, $1 < i < N$, i $1 < j < K$.

Programsko ostvarenje

Za programsko ostvarenje korišten je programski jezik C#. U daljnjem tekstu navedeni su svi upotrebljeni moduli te dan kratki opis funkcije pojedinog modula. Detaljniji opis vidljiv je iz komentara u priloženom kodu.

Klasifikatoru je za uspješan rad potrebno pomoću korisničkog sučelja (koje će biti opisano kasnije), predati datoteke s popisom kategorija i skupovima dokumenata za učenje i testiranje. Datoteke u kojima su zapisani dokumenti iz seta za učenje i testiranje moraju biti zapisani u točno određenom XML formatu koji je opisan u dodatku.

Postupak klasifikacije se obavlja tako da se iz datoteke pročitaju kategorije i pohrane u objekt iz razreda *kategorijaList*. Potom se stvara objekt Skup_dok (razred *dokumentiSkup*) kojem se kao argument u konstruktoru predaje prethodno stvorena lista kategorija. Nakon što je stvoren taj objekt u njega se, pomoću metode *dokumentiSkup.ucitajDokumente,* učitavaju dokumenti za učenje.

Metoda *dokumentiSkup.ucitajDokumente(string datoteka, ref ProgressBar prozor)* učitava dokumente iz datoteke stvarajući prvo objekt klase *dokument* kojeg potom dodaje u listu dokumenata. Kao argument gornjoj metodi predaje se datoteka gdje su zapisani dokumenti i referenca na progressBar prozor.

Sljedeći korak je stvaranje skupa za učenje. Pomoću grafičkog sučelja odabire se metoda po kojoj će se vršiti klasifikacija, ovisno o odabranoj metodi definira se skup za učenje i odabire način prikaza dokumenata u vektorskom prostoru (način zapisa matrice). Ako je pri odabiru načina prikaza dokumenta odabran tfidf zapis ili normalizirani zapis pozvati će se metoda *dokumentiSkup.izracunaj_tfidf(byte zastavica,ref ProgressBar prozor)*, gdje zastavica označava način prikaza dokumenta i definira se pomoću grafičkog sučelja programa. Metoda izračunava težine riječi u dokumentima i računa normalizaciju. Ukoliko je pri odabiru metode odabrana fuzzy k-nn metoda u ovom dijelu će se pozvati metoda *dokumentiSkup.uci_set(bool zas,ref ProgressBar prozor),* koja za argumente prima zastavicu koja određuje na koji se način stvara fuzzy skup i referencu na progressBar prozor.

Nakon definiranja skupa za učenje pristupa se klasifikaciji. U objekt *Skup_klas* učitavaju se dokumenti iz skupa za testiranje, na isti način na koji su se učitavali dokumenti za učenje. Postupak

klasifikacije se odvija na način da se metodi, iz objekta *Skup_dok*, *dokumentiSkup.klasificiraj_KNN((dokument dok, int K, byte zast)* ili metodi *dokumentiSkup.klasificiraj_FKNN(dokument dok, double m, int K,byte zast)*, ovisno koja je metoda klasifikacije odabrana, za prvi argument daje dokument iz objekta *Skup_klas*. Ostali argumenti koje metode prima su broj m (samo za fuzzy), broj K i zastavica koja označava način prikaza dokumenta.

Kategorije iz zadane datoteke učitavaju se u objekt *kategorijaList*. Razred kojem pripada taj objekt je definiran ovako:

(Izvorni kod: kategorijaList.cs)

```
public class kategorijaList
	{
		private ArrayList lista_kategorija;
			//lista kategorija, popunjavana objektima razreda ktegorijaInfo

		//konstruktor
		public kategorijaList()

		//metode
		public bool ucitaj_kategorije(string datoteka)
			//metoda koja učitava kategorije iz datoteke u listu kategorija
		public bool dodaj_kategoriju(kategorijaInfo nova)
			//metoda koja dodaje novu kategoriju u listu
		public int dohvati_broj(string sifra)
			// dohvaća broj kategorije pomoću šifre kategorije
		public string dohvati_sifru(int broj)
			// dohvaća šifru kategorije pomoću broja kategorije
		public void povecaj_brojdok(int n)
			//metoda koja povećava brojač dokumenata koje sadrži pojedina
			kategorija (povećava broj_dokumenata u razredu kategorijaInfo)
		public int dohvati_brojdok(int n)
			//metoda koja vraća broj dokumenata koliko sadrži kategorija s brojem n
		public void postavi_mjere(int i,int j)
			//metoda koja postavlja mjere za ocjenu klasifikacije
		public void provjeri_kategorije(int n)
			//metoda koja vrši provjeru klasifikacije, argument je broj
			dokumenata u skupu za testiranje
```

```
        }
```

Lista kategorija se popunjava objektima razreda *kategorijaInfo*:

(Izvorni kod: kategorijaInfo.cs)

```
public class kategorijaInfo
        {
                public string ime;              //ime kategorije
                public string sifra;            //šifra kategorije
                public int broj;                //broj kategorije
                public int broj_dokumenata;     //broj dokumenata u skupu za učenje
                public int broj_dok;            //broj dokumenata u skupu za testiranje
                public int TP;                  //mjera za uspješnost klasifikacije
                public int FP;                  //mjera za uspješnost klasifikacije
                public int FN;                  //mjera za uspješnost klasifikacije
                public int TN;                  //mjera za uspješnost klasifikacije

                //konstruktori
                public kategorijaInfo(string ime)
                public kategorijaInfo(string ime, string sifra)
                public kategorijaInfo(string ime, int sifra)
                public kategorijaInfo(string ime, string sifra, int broj)

                //metode
                public void postavi_sifra(string sifra) //postavlja stringovnu šifru
                public void postavi_sifra(int sifra)    //postavlja brojčanu šifru
                public void postavi_sifra(string sifra, int broj) //postavlja obje šifre
                public void postavi_broj_dok(int n)
                        //metoda koja zapisuje broj dokumenata koji pripadaju ovoj kategoriji
```

Dokumenti nad kojima se obavlja klasifikacija, učitavaju se iz datoteke, čija je struktura zapisa objašnjena u dodatku, u objekt razreda *dokumentiSkup*. Razred *dokumentiSkup* između ostalog sadrži listu dokumenata, koja je popunjavana objektima iz razreda *dokument*. Prilikom učitavanja dokumenta u listu, prvo se učita dokument i stvori objekt tipa *dokument* koji se potom dodaje u listu. Prvo će biti opisan razred *dokument* a potom razred *dokumentiSkup*.

Razred *dokument* :

(Izvorni kod: dokument.cs)

```
public class dokument
	{
		private int _kategorija;   //broj kategorije kojoj dokument pripada
		public kategorijaFuzzySet kategorijaFuzzy;
			//fuzzy skup koji opisuje pripadnost dokumenta pojedinoj kategoriji
		public uint redni_broj;          //redni broj dokumenta
		public ArrayList lista_rijeci;   //lista rijeci koje sadrži dokument
		public uint broj_rijeci;         //broj riječi u dokumentu
		bool zas_fuzzy; //zastavica koja označuje jeli postavljen fuzzy skup
		public double norm; //korijen kvadratnih vrijednosti težina rijeci

		//konstruktori
		public dokument(int broj, int broj_kat)
		public dokument(int broj, kategorijaInfo kat)
		public dokument(uint broj, int broj_kat, uint n)

		//metode
		public bool dodaj_rijec(rijec nova)
			//metoda koja dodaje novu riječ u dokument
		public void poredaj()
			//metoda koja sortira rijeci u listi po broju rijeci
		public void postavi_fuzzy(int i)
			// metoda koja definira fuzzy skup na prvi način (opisano u uvodnom
			dijelu)
		public void postavi_fuzzy_2(int i,kategorijaList kategorije, int k)
			// metoda koja postavlja fuzzy skup na drugi način
		public int procijeni_kategoriju()
			//metoda koja iz fuzzy skupa, koji je pridodjeljen ovom dokumentu,
			dohvaća broj kategorije kojoj dokument najviše pripada
	}
```

Unutar gore opisanog razreda korišteni su još neki razredi. Za zapis (opis) rijeci korišten je razred *rijec*, dok je za opis Fuzzy skupa korišten razred kategorijaFuzzySet.

Razred *rijec* :

(Izvorni kod: rijec.cs)

```
public class rijec : IComparable
	{
		private uint _broj;              //jedinstveni broj rijeci
```

```
        private uint _broj_u_dok;        //označuje koliko se puta riječ pojavila
        private double _tezina;

        //konstruktori
        public rijec(string ime, uint broj, double tezina)
        public rijec(double tezina, int broj, string ime)
        public rijec(string ime, int broj, double tezina)
        public rijec(double tezina, int broj)

        //metode
        public void povecaj() //povećava broj pojavljivanja riječi u dokumentu
        public int CompareTo(object obj) //nadogradnja usporedbe
    }
```

Za definiranje fuzzy skupa koristi se razred *kategorijaFuzzySet:*

(Izvorni kod: kategorijaFuzzySet.cs)

```
public class kategorijaFuzzySet
    {
        private double[] fuzzySet; //polje s vrijednostima pripadnosti
        int broj;                  //broj kategorija(veličina skupa)

        //konstruktor
        public kategorijaFuzzySet(int n)
        //metode
        public void postavi_vrijednost(double vr, int index)
            //metoda koja postavlja vrijednost pripadnosti
        public double dohvati_vrijednost(int index
            //metoda koja dohvaća vrijednost pripadnosti
        public int dohvati_max()
            //metoda koja vraća index (broj) kategorije za koju je pripadnost
            funkcije najveća
    }
}
```

U gore opisanom razredu, index je broj od 1 do n (n je broj kategorija).

Dokumenti iz datoteke učitavaju se u objekt iz razreda *dokumentiSkup*

Razred *dokumentiSkup*:

(Izvorni kod: dokumentiSkup.cs)

```
public class dokumentiSkup
```

```
{
        public ArrayList lista_dokum;     //lista u kojoj se nalaze dokumenti
        private int broj_dok;             //broj dokumenata u skupu
        private int broj_rijeci = 0;      //broj različitih rijeci u čitavom skupu
        private byte[] lista_rijeci;      //lista koja sadrži podatak u koliko se
dokumenata pojavila rijec
        private byte[] lista_rijeci_pom; //lista koja sadrži podatak koja se riječ
pojavila u novom dokumentu
        public kategorijaList kategorije;//lista kategorija
        public int broj_dok_klas;         //broj dokumenata koji su se klasificirali


        //konstruktor
        public dokumentiSkup(kategorijaList kategorije)
                //argument je objekt s zadanim kategorijama
        //metode
        public void dodajDokument(dokument novi)
                //metoda koja dodaje dokument u listu
        public bool ucitajDokumente(string datoteka, ref  ProgressBar prozor)
                //metoda koja učitava dokumente iz zadane datoteke u listu dokumenata
        public void izracunaj_tfidf(byte zast, ref  ProgressBar prozor)
                //metoda koja računa tfidf matricu i koeficijent normu
        public void dodaj_kraj_poredaj(ref  ProgressBar prozor)
                //metoda sortira dokumente
        public void uci_set(bool zas,ref  ProgressBar prozor)
                //metoda koja postavlja fuzzy set
        public double udaljenost(dokument prvi, dokument drugi, byte zast)
                //metoda koja računa udaljenost između dokumenata
        private void dohvati_K(dokument dok, int K, ref ArrayList polje_K,byte zast)
                //metoda koja izračunava k susjeda
        public int klasificiraj_KNN(dokument dok, int K,byte zast)
                //metoda koja klasificira dokument pomoću K-nn
        public int klasificiraj_FKNN(dokument dok, double m, int K,byte zast)
                //metoda koja klasificira dokument pomoću fuzzy K-nn

}
```

Lista dokumenata se puni objektom iz razreda dokument.

Rezultati rada klasifikatora

Ispitivanje je provedeno na bazi novinskih članaka dnevnog lista *Vjesnik* iz razdoblja od 2000. do 2003. godine. Baza sadrži preko 92000 članaka preuzetih iz Hrvatskog nacionalnog korpusa. Dolazi u obliku jedne datoteke pisane u XML formatu. Članci su prikazani u vertikaliziranom obliku (svaka pojavnica u jedan redak), dok su stop riječi uklonjene.

Početak svakog članka označen je oznakom:

<doc type="article" file="vjGGGGMMDDKK01">

gdje pojedini dijelovi oznake imaju sljedeće značenje:

- Vj – označava pripadnost članka bazi članaka Vjesnik
- GGGG – godina pisanja članka
- MM – mjesec pisanja članka
- DD – dan pisanja članka
- KK – jedna od 11 kategorija u koju je razvrstan dokument

Kraj zapisa dokumenta označen je oznakom </doc>

Nad bazom od 92000 članaka izgrađeni su skupovi za učenje i testiranje, tako da svaki skup sadrži točno 10000 članaka podijeljenih u osam kategorija (1250 članaka po kategoriji). Gore navedene skupove generirao je Mislav Malenica u okviru svog diplomskog rada [5]. Detaljan opis postupka generiranja skupova razrađen je u gore navedenom radu i ovdje se neće detaljno objašnjavati

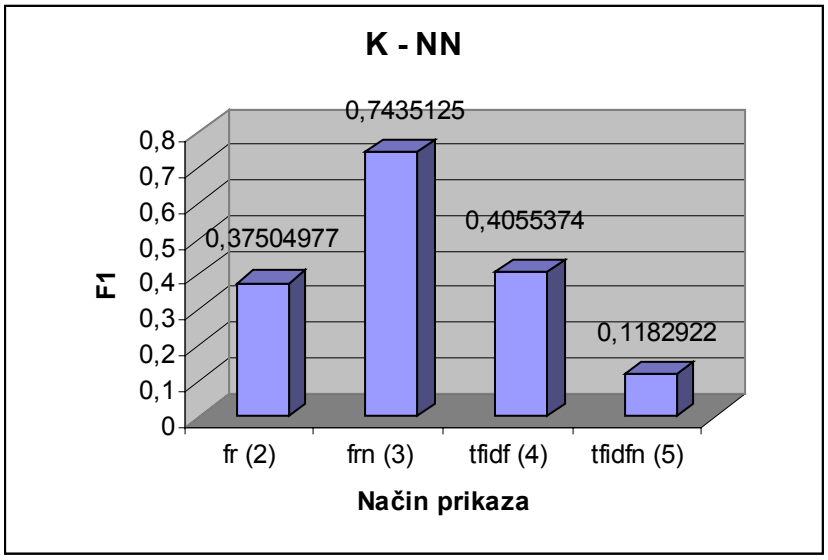
Nad generiranim skupovima provedena je morfološka obrada postupkom Automatske morfološke normalizacije koju je razvio dipl. ing. Jan Šnajder [6], asistent na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva.

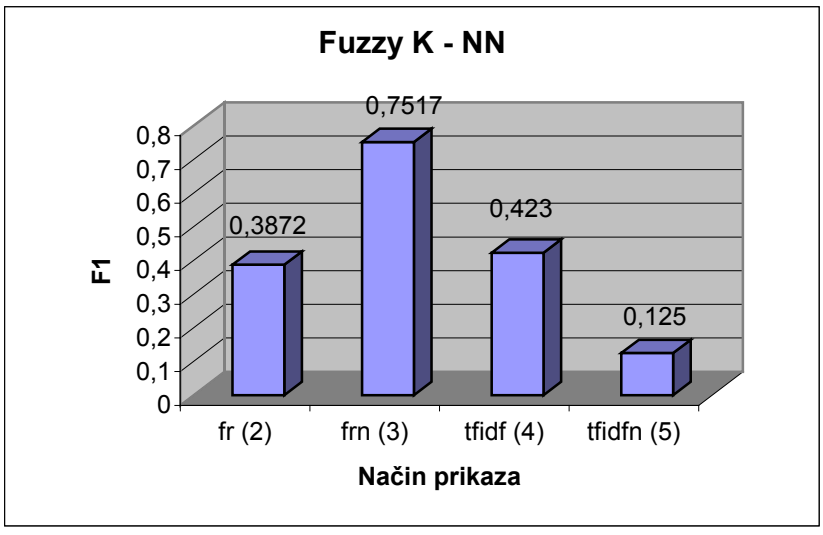
Tko stvoreni i obrađeni skupovi predstavljaju ulaz u ovaj klasifikator i nad njima se obavlja klasifikacija.

Mjerenja su obavljena nad skupovima za testiranje i učenje. Za klasifikaciju je uz K-nn metodu korištena i fuzzy k-nn metoda.

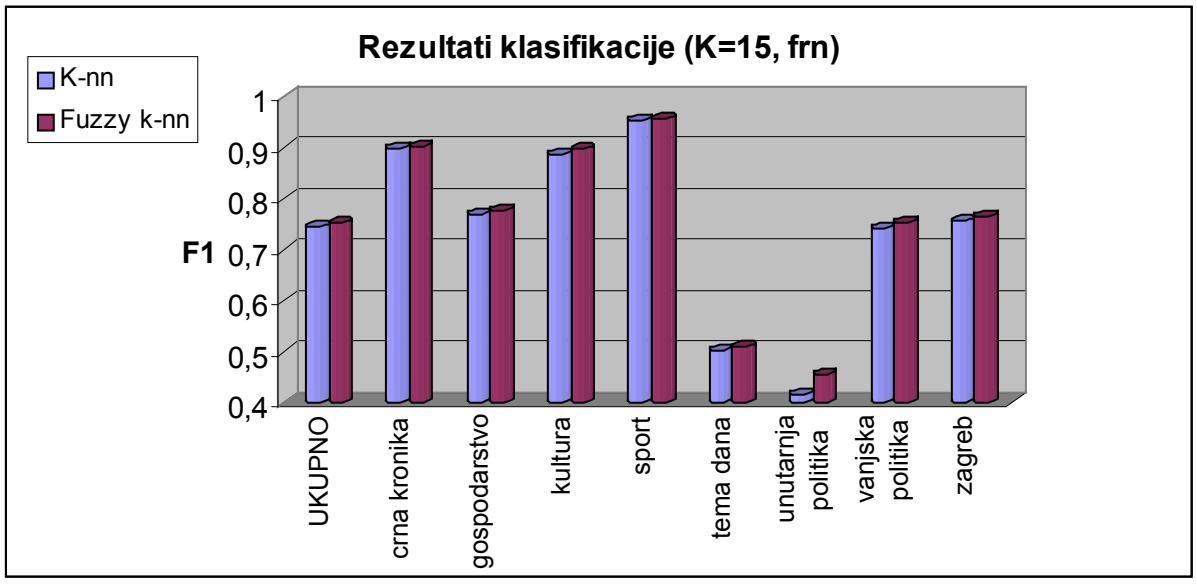
Uspješnost rada klasifikatora ovisno o načinu prikaza dokumenta

Načini prikaza dokumenta u vektorskom prostoru objašnjeni su u prvom poglavlju ovog seminarskog rada. Mjerenja su obavljena za četiri načina prikaza dokumenta i to za frekvenciju riječi (2), normaliziranu frekvenciju riječi (3), tfidf zapis (4) i normalizirani tfidf zapis. Parametar k je postavljen na 15.

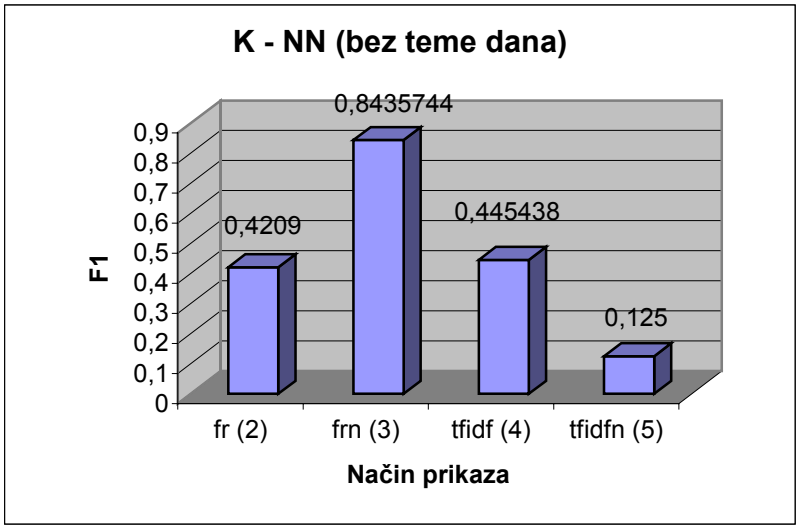


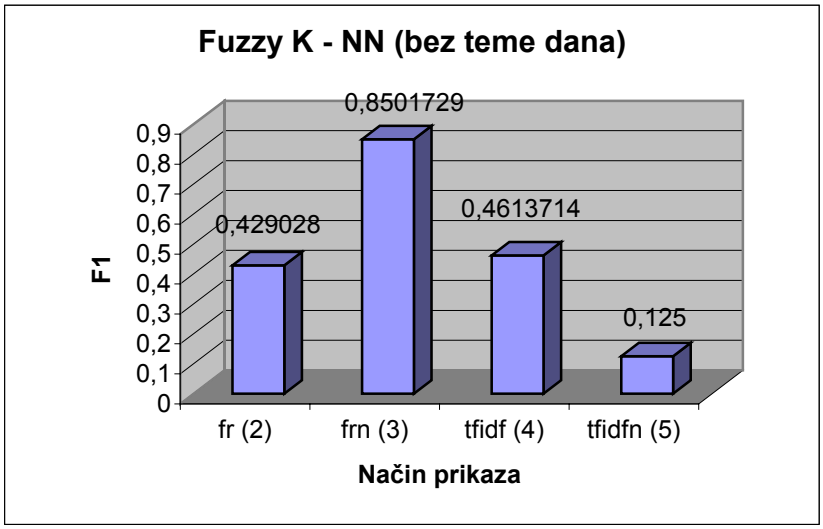


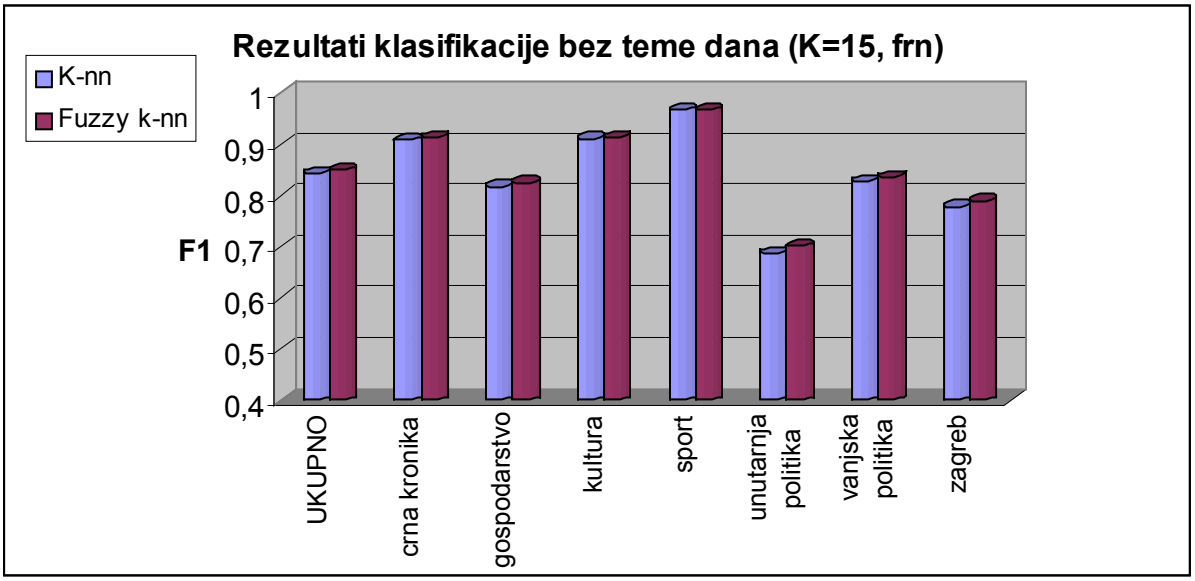
Jednostavno se zaključuje da klasifikator daje najbolje rezultate prilikom korištenja normalizirane frekvencije riječi. Sljedećim grafikonom prikazani su rezultati klasifikacije po kategorijama.



Pokazalo se da su rezultati klasifikacije za kategorije «tema dana» i «unutarnja politika» jako loši i negativno utječu na ukupne rezultate klasifikacije. Pošto kategorija «tema dana» ne predstavlja kategoriju koja ima neko posebno značenje, provedeno je mjerenje bez te kategorije. Kategorija je izbačena i iz seta za učenje i iz skupa za testiranje, tako da se sada u oba skupa nalazi 8750 članaka podijeljenih u 7 kategorija.



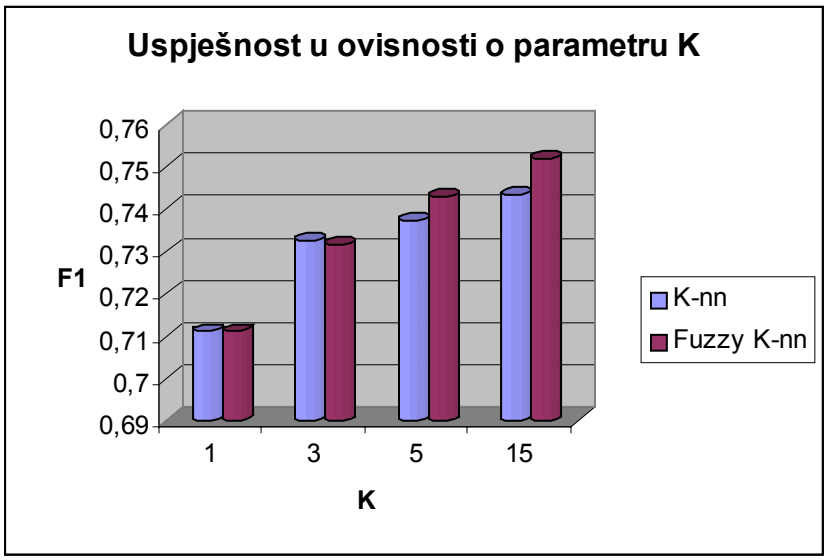


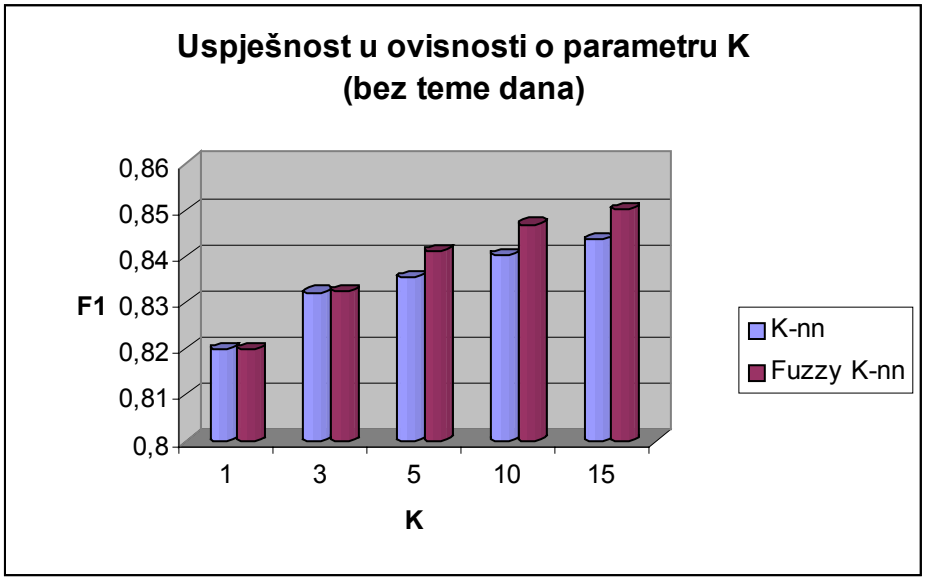


Pokazalo se da izbacivanjem kategorije «tema dana» rezultati klasifikacije porastu za 10-ak posto.

Uspješnost rada klasifikatora u ovisnosti o parametru K

Prijašnjim se mjerenjima pokazalo da klasifikator najbolje radi za način prikaza dokumenata pomoću normalizirane frekvencije riječi, stoga se za daljnja mjerenja uzima samo ovaj način prikaza. Mjerenja će se opet obaviti u dva navrata, uzimajući u obzir kategoriju «tema dana» i bez nje.





Zaključak

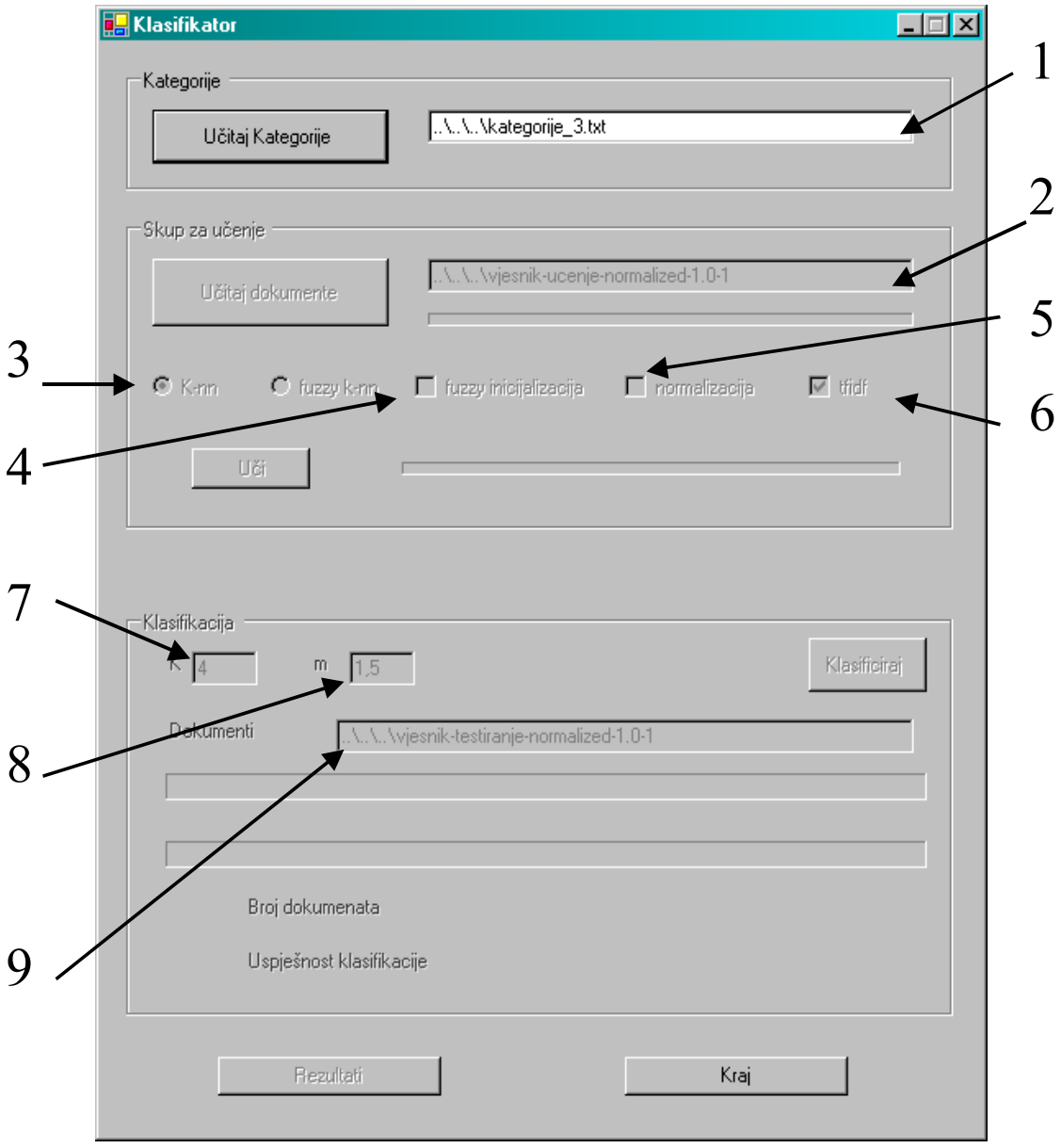
Mjerenjima se pokazalo da je metoda najuspješnija za K=15 i način prikaza dokumenata u vektorskom prostoru koristeći normaliziranu frekvenciju riječi. Tablično su pokazani detaljniji rezultati klasifikacije za gore navedene postavke:

kategorija	Cijela baza		Baza bez teme dana	
	k-nn	Fuzzy k-nn	k-nn	Fuzzy k-nn
crna kronika	0,90	0,90	0,91	0,91
gospodarstvo	0,77	0,78	0,82	0,82
kultura	0,89	0,90	0,91	0,91
sport	0,95	0,96	0,97	0,97
tema dana	0,50	0,51		
unutarnja politika	0,41	0,45	0,69	0,70
vanjska politika	0,74	0,75	0,83	0,84
Zagreb	0,76	0,76	0,78	0,79
UKUPNO	0,74	0,75	0,84	0,85

Pokazalo se da prilikom svih provedenih mjerenja fuzzy k – nn metoda daje bolje rezultate od k-nn metode. Razlog tome je što k – nn ne može zaključiti u koju kategoriju spada dokument ako se unutar K najbližih susjeda pojavi jednak broj dvije ili više kategorija. Fuzzy k-nn rješava taj problem pridodjeljivajući svakom dokumentu postotak pripadnosti svakoj kategoriji i onda uzimajući najveću vrijednost.

Korisničko sučelje

Prilikom pokretanja programa pojavljuje se prozor kao na slici 1.



Slika 1.

U polje, označeno brojem jedan, upisuje se datoteka u kojoj se nalazi popis kategorija, potom se klikne na «Učitaj kategorije». Nakon što program učita kategorije, u polje označeno brojem dva,

upisuje se ime datoteke u kojoj se nalaze zapisani dokumenti, u formatu koji je opisan u dodatku. Potom se klikne na «Učitaj dokumente».

Nakon što program učita dokumente potrebno je definirati skup za učenje, odabrati način prikaza dokumenta i metodu klasifikacije. Metoda klasifikacije odabire se pritiskom na ženjenu metodu (oznaka 3) a način prikaza dokumenta postavljenjem kvačice na polja označena brojem 5 i 6. Ukoliko je odabrana fuzzy klasifikacija potrebno je odabrati način stvaranja fuzzy skupa. Ukoliko se, u kučicu označenu brojem 4, postavi kvačica odabran je drugi način, u suprotnom prvi način. Razlika između prvog i drugog načina objašnjena je u uvodu. Nakon definiranja tih parametara potrebno je kliknuti na «Uči»

Konačno, kada se definira skup za učenje, pristupa se klasifikaciji. Klasifikacija se obavlja klikom na «Klasificiraj». Prije je potrebno navesti datoteku u koju se nalaze dokumenti koje treba klasificirati odnosno skup za testiranje (oznaka 9), i definirati parametri K (oznaka 7) i m (oznaka 8).

Rezultati klasifikacije dobivaju se pritiskom na «Rezultati».

Dodatak

Struktura datoteke s popisom kategorija

Kategorije se navode pojedinačno u svaki red i to na način da se prvo navede šifara a zatim naziv kategorije. Primjer jedne datoteke je prikazan na slici 2.

Slika 2.

Struktura datoteke s dokumentima

Dokumenti su navedeni u XML datoteci u vertikaliziranom obliku (svaka pojavnica u jedan redak). Pojavnica se zapisuje u obliku:

_X,

gdje X predstavlja jedinstveni broj rijeci (pojavnice). Početak svakog članka označen je oznakom:

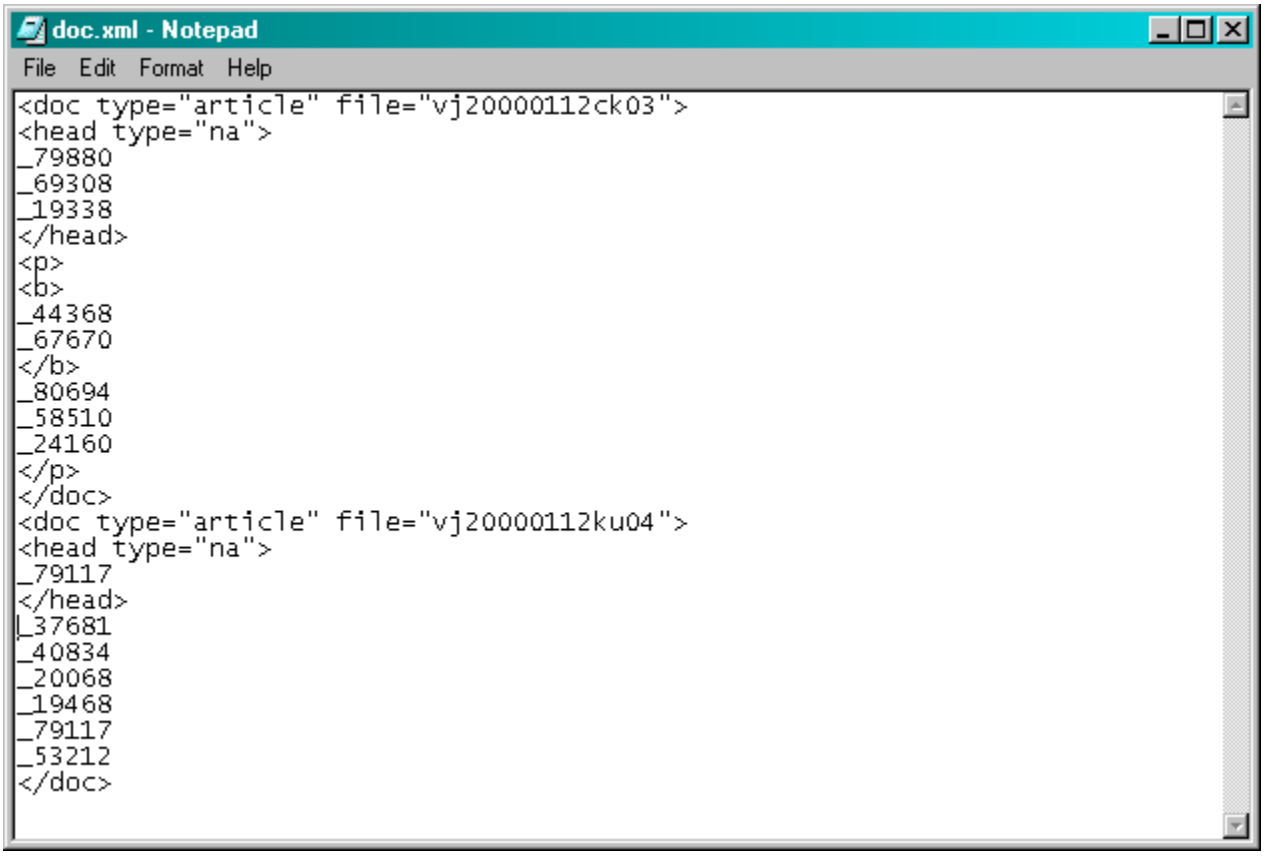
<doc type="article" file="vjGGGGMMDDKK01">,

gdje pojedini dijelovi oznake imaju sljedeće značenje:

- Vj – označava pripadnost članka bazi članaka Vjesnik
- GGGG – godina pisanja članka
- MM – mjesec pisanja članka
- DD – dan pisanja članka
- KK – jedna od 11 kategorija u koju je razvrstan dokument

Kraj zapisa članka označen je oznakom </doc>.

Ostale oznake (odlomak, novi redak i drugi znakovi xml-a) se zanemaruju. Primjer zapisa datoteke je prikazan na slici 3.



Slika 3

Literatura

[1] Joon H. Han and Yoon K. Kim, "A Fuzzy K-NN Algorithm using Weights from the Variance of Membership Values", IEEE, 1999

[2] Shixin Yu, Steve de Backer and Paul Scheunders, "Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery", Pattern Recognition Letters 183-190, 2002

[3] Haiyun Bian and Lawrence Mazlack, "Fuzzy-Rough Nearest-Neighbor Classification Approach", University of Cincinnati

[4] L. Gyergyek, N. Pavešić, S. Ribarić, "Uvod u raspoznavanje uzoraka", Tehnička knjiga, Zagreb, 1988

[5] Mislav Malenica, "Primjena jezgrenih metoda u kategorizaciji teksta", Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu

[6] Jan Šnajder, http://www.zemris.fer.hr/~jan, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu