



T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
MAKİNE ÖĞRENİMİNE GİRİŞ

MAKİNE ÖĞRENİMİ İLE METİN BENZERLİĞİ BULMA

Hazırlayan
1560007 Kübra Nur Saruhan

Danışman
Prof. Dr. Erdal KILIÇ

Ocak 2019
SAMSUN

İçindekiler

Giriş	3
Gerçekleştirme.....	3
Eğitim Ve Test Veri Seti	4
Kelime Köklerinin Bulunması.....	5
Metinlerin Frekans Değerlerinin Bulunması	6
Eğitim ve Test Veri Setlerinin Arff Olarak Kaydedilmesi	6
KNN Algoritması	8
Program Arayüzü.....	9
Sonuç	10

Giriş

Türkçe metinler arası benzerlik bulma işlemini makine öğrenimi sınıflandırma algoritmaları kullanılarak gerçekleştireceği bir projedir. Sınıflandırma algoritmalarından önce veri kullanıma uygun hale getirilmesi gerekir. Benzerlik bulma işlemi için ilk olarak metindeki kelimelerin köklerine bakılması gerekir. Türkçe sondan eklemeli bir dil olduğu için kelimenin eklerinden arındırılıp köklerinin bulunması gerekir. Kökleri bulunduktan sonra metin frekans değerleri oluşturulmalı ve sınıflandırma algoritmasına göre benzerlik bulunmalıdır.

Gerçekleştirme








































Proje, Python dilinde gerçekleştirilmiştir. Geliştirme ortamı olarak Spyder kullanılmıştır.

Metin köklerinin bulunması için Zemberek kütüphanesi kullanılmıştır. Python'nun makine öğrenmesi için sağladığı kütüphanelerden gerekli olanları kullanılmıştır.



















Eğitim Ve Test Veri Seti

Eğitim ve test veri seti için internette bulunan farklı kaynakların kitap özetlerinden yararlanılarak veri seti oluşturulmuştur. Her kitap için 3 farklı özet şeklinde oluşturularak benzerlik oranlarının doğruluğunun daha iyi anlaşılması hedeflenmiştir. Eğitim seti için 54 veri, test verisi için eğitim setinin içinden seçilen 18 veri şeklinde oluşturulmuştur.

Eğitim Verileri:

								
ask.txt	ask2.txt	ask3.txt	calikusu.txt	calikusu2.txt	calikusu3.txt	dijitalKale.txt	dijitalKale2.txt	dijitalKale3.txt
								
dokuzuncuHariciyeKogusu.txt	dokuzuncuHariciyeKogusu2.txt	dokuzuncuHariciyeKogusu3.txt	eylul.txt	eylul2.txt	eylul3.txt	kayipSembol.txt	kayipSembol2.txt	kayipSembol3.txt
								
kiralikKona.txt	kiralikKona2.txt	kiralikKona3.txt	kucukAga.txt	kucukAga2.txt	kucukAga3.txt	kurkMantoluMadonna.txt	kurkMantoluMadonna2.txt	kurkMantoluMadonna3.txt
								
kuyucakliYusuft.txt	kuyucakliYusuft2.txt	kuyucakliYusuft3.txt	olasiliksiz.txt	olasiliksiz2.txt	olasiliksiz3.txt	saatleriAyarlamaEnstitusu.txt	saatleriAyarlamaEnstitusu2.txt	saatleriAyarlamaEnstitusu3.txt
								
sefiller.txt	sefiller2.txt	sefiller3.txt	sinekliBakkal.txt	sinekliBakkal2.txt	sinekliBakkal3.txt	sucVeCeza.txt	sucVeCeza2.txt	sucVeCeza3.txt
								
ucurtmaAvcisi.txt	ucurtmaAvcisi2.txt	ucurtmaAvcisi3.txt	yaban.txt	yaban2.txt	yaban3.txt	yaprakDokumu.txt	yaprakDokumu2.txt	yaprakDokumu3.txt

Test Verileri:

								
ask.txt	calikusu.txt	dijitalKale.txt	dokuzuncuHariciyeKogusu.txt	eylul3.txt	kayipSembol.txt	kiralikKona.txt	kucukAga.txt	kurkMantoluMadonna.txt
								
kuyucakliYusuft.txt	olasiliksiz.txt	saatleriAyarlamaEnstitusu.txt	sefiller.txt	sinekliBakkal.txt	sucVeCeza.txt	ucurtmaAvcisi.txt	yaban.txt	yaprakDokumu.txt

Kelime Köklerinin Bulunması

Metin benzerliği ele alındığında kelimeler üzerinden yola çıkmamız gerekir. Türkçe sonradan eklemeli bir dil olduğu için kelimenin kökü aynı olsa dahi makine eklerle birlikte farklı bir kelime algılayacaktır. Bunu makinenin aynı olduğunu anlayabilmesi için kelime köklerinin bulunması gerekir. Doğal dil işleme konusu olan bu konuda Türkçe metinlerin köklerinin bulunması için yazılmış Zemberek kütüphanesi kullanılmıştır.

Kelime Kökü Bulunurken Gerçekleştirilen İşlemler:

- Okunan metnin bütün kelimeleri küçük harfe çevrilir.
- Noktalama işaretleri temizlenir.
- “Ve, veya...” gibi anlamsız kelimeler nltk.corpus stopwords kütüphanesi kullanılarak çıkartılır.
- Zemberek jar dosyası projeye jpype kütüphanesi kullanılarak çalıştırılabilir hale getirilir. Projede java kodunun çalışabilmesi için bilgisayardaki jvm.dll uzantısının eklenmesi gerekir. Kodu aşağıdaki gibidir.

```
def zemberek():
    jvmDllPath = r"C:\Program Files\Java\jdk1.8.0_171\jre\bin\server\jvm.dll"
    try:
        jpype.startJVM(jvmDllPath, "-ea", "-Djava.class.path=C:/Users/saruhan/Desktop/Machine Learning Text Similarity/zemberek-tum-2.0.jar")
    except:
        pass
    Tr = jpype.JClass("net.zemberek.tr.yapi.TurkiyeTurkcesi")
    tr = Tr()
    Zemberek = jpype.JClass("net.zemberek.erisim.Zemberek")
    zemberek = Zemberek(tr)
    return zemberek
```

- Zemberek projede çalışır hale getirildikten sonra Zemberek'in bir fonksiyonu olan `kelimeCozumle()` ile kelime kökleri bulunur.

Eğitim setini kök listesinin bir kısmı aşağıdaki gibidir.

['abis', 'aci', 'ad', 'adalet', 'adam', 'aday', 'adeta', 'adlan', 'adım', 'affet', 'ahlaki', 'aile', 'ajan', 'akla', 'akraba', 'akıl', 'aksam', 'al', 'alan', 'aldat', 'aldır', 'ali', 'alkol', 'alt', 'altın', 'alın', 'alışveriş', 'ameliyat', 'amir', 'an', 'ana', 'ancak', 'anla', 'anlam', 'anlat', 'anlas', 'anne', 'anormal', 'ant', 'anı', 'apartman', 'ara', 'araştırma', 'araştırman', 'arkadas', 'art', 'artık', 'asalet', 'asistan', 'asker', 'asri', 'asıl', 'at', 'ata', 'ates', 'atıl', 'avcı', 'avukat', 'ay', 'ayak', 'ayar', 'ayarla', 'aydın', 'aynen', 'aynı', 'ayrıl', 'ayrım', 'ayrış', 'ayrış', 'azap', 'azar', 'azı', 'aç', 'açan', 'acı', 'acıkla', 'ağa', 'ağac', 'ağır', 'aşığa', 'aşk', 'aşkın', 'aşık', 'baba', 'babacan', 'bacak', 'bahçe', 'bahçıvan', 'bak', 'bakan', 'bakkal', 'balta', 'banka', 'bar', 'barikat', 'barın', 'basit', 'bazi', 'bag', 'bagdas', 'bagla', 'bagım', 'baş', 'başar', 'başarı', 'başka', 'başkahraman', 'başkan', 'başkent', 'başla', 'basyapıt', 'beddua', 'bedeni', 'bekle', 'beklenti', 'bela', 'belediye', 'belin', 'ben', 'benze', 'benzer', 'benzet', 'beraber', 'beri', 'besle', 'betimle', 'bey', 'beyin', 'beğeni', 'bes', 'bil', 'bildik', 'bildir', 'bile', 'bilgi', 'bilgin', 'bilgisayar', 'bilim', 'bin', 'binlerce', 'bir', 'birader', 'biraz', 'birbir', 'birdenbire', 'bিরer', 'biri', 'birileri', 'birles', 'birlik', 'birlikte', 'birçok', 'bitir', 'bitkin', 'biçim', 'bocala', 'bomba', 'borc', 'botanik', 'boy', 'boyun', 'bozan', 'bozuk', 'bozul', 'boş', 'boşan', 'boşanma', 'bu', 'buğün', 'bul', 'bulan', 'buluşacı', 'bulun', 'bulundur', 'bulus', 'buna', 'bura', 'burs', 'böcek', 'bölge', 'bölü', 'böyle', 'böylece', 'bürokrasi', 'büsbütün', 'bütün', 'büyü', 'büyük', 'bık', 'bırak', 'cami', 'can', 'cehennem', 'celal', 'cemil', 'cennet', 'cephe', 'cevap', 'cevapla', 'ceza', 'ciddi', 'cihan', 'cila', 'cinayet', 'ciğer', 'coşku', 'cüce', 'daha', 'dahil', 'dahili', 'daima', 'dair', 'dal', 'damat', 'dan', 'davran', 'dayan', 'dayı', 'dağ', 'de', 'dede', 'dedikodu', 'defa', 'defter', 'dek', 'delikanlı', 'dene', 'deneme', 'deney', 'deneyim', 'depo', 'dere', 'derce', 'dergi', 'derin', 'ders', 'destek', 'destekle', 'detay', 'devam', 'devlet', 'değer', 'değerler', 'değil', 'değın', 'değış', 'değışik', 'değışim', 'değışim', 'dijital', 'dikkat', 'dil', 'dilenci', 'dini', 'dinle', 'diz', 'dizin', 'diğer', 'doktor', 'doktora', 'dokuz', 'dolap', 'dolar', 'dolay', 'dolas', 'dolu', 'doruk', 'dost', 'doğ', 'doğan', 'doğacılama', 'doğru', 'doğrul', 'doğum', 'dram', 'dul', 'dur', 'durum', 'duy', 'duyur', 'duyu', 'dön', 'dönm', 'döner', 'dönüş', 'dört', 'dükkân', 'dünya', 'dürüst', 'duyun', 'düzen', 'düzenle', 'dügün', 'düş', 'düşkün', 'düşman', 'düşün', 'düşünür', 'düşür', 'dır', 'ede', 'edebiyat', 'eden', 'edin', 'editör', 'efendi', 'ekle', 'ekonomik', 'el', 'elde', 'ele', 'elif', 'elli', 'elmas', 'emek', 'emekli', 'emin', 'emin', 'engelle', 'enstitü', 'epeyce', 'er', 'erdem', 'erkek', 'ertesi', 'eser', 'esinle', 'esir', 'eski', 'esrarengiz', 'et', 'etki', 'etkile', 'ev', 'evin', 'evlat', 'evlen', 'evli', 'evlilik', 'evliya', 'evren', 'ez', 'eğitim', 'eğlence', 'esin', 'eskiya', 'esya', 'fabrika', 'fahiş', 'faik', 'fakat', 'fakin', 'fakülte', 'fare', 'fark', 'fatih', 'fayda', 'fazla', 'felaket', 'felsefe', 'feri', 'fikin', 'fikri', 'film', 'flört', 'fırsat', 'fırtına', 'gayret', 'gazete', 'gazino', 'gebe', 'gece', 'gecelik', 'gel', 'gelecek', 'gelin', 'gelin', 'gelis', 'gelistir', 'genel', 'genişle', 'genc', 'gerek', 'geri', 'gerçek', 'gerçekle', 'gerçil', 'getir', 'getiri', 'gezi', 'geç', 'geçen', 'geçer', 'geçim', 'geçin', 'geçir', 'geçmiş',

Metinlerin Frekans Değerlerinin Bulunması

Kelime köklerinin bulunmuş metinlerin benzerlik oranlarının hesaplanabilmesi için sayısal değerlere çevrilmesi gerekir. Metnin içinde ilk olarak kelimelerin kaç defa geçtiğini bulunması gerekir. Bunun için metnin kelime köklerini alfabetik olarak sıralayıp geçme sıklığını matrise çeviren CountVectorizer sınıfını kullanılmıştır. Bunun sonucunda eğitim veri seti yüklediğimizde sabit uzunlukta dizi oluşturularak her metin için ayrı ayrı hesaplanarak bir matris elde edilmiştir.

Benzerlik hesaplaması için kaç defa geçtiğinden metindeki kelimelerin baskınlık oranına göre hesaplanması daha uygundur. Bunun için “Terim Frekansı x Ters Belge Frekansı” anlamına gelen “TF x IDF” modeli kullanılmıştır. CountVectorizer ile elde ettiğimiz matrisi TfidfTransformer sınıfını kullanarak yeni bir matris elde edilmiştir.

Eğitim ve Test Veri Setlerinin Arff Olarak Kaydedilmesi

Eğitim veri setindeki metinlerin kelime kökleri ve frekansları bulunduktan sonra arff olarak kaydedilir. Test veri setinin oluşturulabilmesi için arff olarak kaydettiğimiz eğitim setinin seçilmesi gerekir. Modeli eğitimi için geçerli olan kelimeler eğitim setini esas alması gerekir. Böylelikle karşılaştırılan dizi uzunlukları aynı olacaktır. Test verisinde geçen eğitim verisinde geçmeyen kelime alınmamış olur.

Yeni test veri setinin bağlı olduğu eğitim setine göre frekans değerlerinin bulunmasının kod karşılığı aşağıdaki gibidir.

```
def newTest(name, pathTest, pathTrainArff):
    data, meta = Model.loadArff(pathTrainArff)

    #Bağlı olduğu eğitim verisini alıyoruz.Dizi uzunluğu için gerekli
    trainList = []
    attributeList = []
    attributeList = meta.names()
    for i in range(len(attributeList)-1):
        trainList.append(attributeList[i])
    print(trainList)

    vectorizer = CountVectorizer()
    trainVectorizerArray= vectorizer.fit_transform(trainList).toarray()
    print('TrainVectorizerArray', trainVectorizerArray)

    transformer = TfidfTransformer()

    testSet = []
    testTextList = []
    #read() Metinleri kelime kökleri bulunmuş şekilde getirir
    testTextList = Model.read(pathTest)
    for row in testTextList:
        testSet.append(' '.join([str(elem) for elem in row]))
    print (testSet)

    testVectorizerArray = vectorizer.transform(testSet).toarray()
    print("testVectorizerArray", testVectorizerArray)

    transformer.fit(testVectorizerArray)
    test = transformer.transform(testVectorizerArray).toarray()
    print ('Test Transform',test)

    wordList = vectorizer.get_feature_names()
    print(wordList)

    Model.arffCreate(name, pathTest, test, wordList)
```

Arff seti oluşturularak farklı sınıflandırma algoritmaları için de hazır hale getirilmiştir. Oluşturulan arff test setinin bir kısmının ekran görüntüsü aşağıdaki gibidir.

[illegible]

KNN Algoritması

Eğitim ve test veri setlerinin benzerliklerini hesaplayacak formata çevirdikten sonra sınıflandırma algoritmasına karar verilmesi gerekir. Makine öğrenmesi sınıflandırma algoritmalarından K-Nearest Neighborhood ile en yakın metin benzerliklerinin hesaplanmıştır. En yakın komşu bulunurken kosinüs benzerliğinden yararlanılmıştır.

- İlk olarak K değeri belirlenir.
- Eğitim ve test verisi yüklenir.
- Test verisine en yakın eğitim verileri kosinüs benzerliğine göre bulunur.

Knn algoritmasına göre sınıflandırma kod karşılığı aşağıdaki gibidir.

```
def knn(pathTrain, pathTest, n):  
    dataTrain, metaTrain = Model.loadArff(pathTrain)  
    dataTest, metaTest=Model.loadArff(pathTest)  
  
    result = []  
    knn = NearestNeighbors(n_neighbors=n, metric='cosine')  
    knn.fit(dataTrain)  
    kneighbors = knn.kneighbors(dataTest)  
    print(kneighbors)  
  
    for i, item in enumerate(kneighbors[0]):  
        for j, item in enumerate(kneighbors[0][i]):  
            result.append([(metaTest['class'][1][i]),(metaTrain['class'][1][kneighbors[1][i][j]]) ,round((1-kneighbors[0][i][j])*100, 3)])  
    return (result)
```

K =5 iken benzerlik sonuçları:

Aynı kitabın farklı özetlerini ilk sıralarda bulması programımızın gayet doğru şekilde çalıştığını gösterir.

```
Test ask.txt Eğitim ask.txt Benzerlik Oranı: 99.313  
Test ask.txt Eğitim ask2.txt Benzerlik Oranı: 26.319  
Test ask.txt Eğitim ask3.txt Benzerlik Oranı: 23.561  
Test ask.txt Eğitim kurkMantoluMadonna3.txt Benzerlik Oranı: 20.855  
Test ask.txt Eğitim kuyucaklıYusuf.txt Benzerlik Oranı: 19.686  
Test calikusu.txt Eğitim calikusu.txt Benzerlik Oranı: 99.378  
Test calikusu.txt Eğitim calikusu3.txt Benzerlik Oranı: 48.681  
Test calikusu.txt Eğitim calikusu2.txt Benzerlik Oranı: 43.418  
Test calikusu.txt Eğitim ucurtmaAvcisi2.txt Benzerlik Oranı: 21.471  
Test calikusu.txt Eğitim yaban.txt Benzerlik Oranı: 20.427  
Test dijitalKale.txt Eğitim dijitalKale.txt Benzerlik Oranı: 99.391  
Test dijitalKale.txt Eğitim dijitalKale3.txt Benzerlik Oranı: 35.189  
Test dijitalKale.txt Eğitim dijitalKale2.txt Benzerlik Oranı: 30.872  
Test dijitalKale.txt Eğitim kayipSembol.txt Benzerlik Oranı: 23.049  
Test dijitalKale.txt Eğitim ask.txt Benzerlik Oranı: 15.316  
Test dokuzuncuHariciyeKogusu.txt Eğitim dokuzuncuHariciyeKogusu.txt Benzerlik Oranı: 99.171  
Test dokuzuncuHariciyeKogusu.txt Eğitim dokuzuncuHariciyeKogusu3.txt Benzerlik Oranı: 40.145  
Test dokuzuncuHariciyeKogusu.txt Eğitim dokuzuncuHariciyeKogusu2.txt Benzerlik Oranı: 34.979  
Test dokuzuncuHariciyeKogusu.txt Eğitim calikusu3.txt Benzerlik Oranı: 23.127  
Test dokuzuncuHariciyeKogusu.txt Eğitim eylul.txt Benzerlik Oranı: 21.867  
Test eylul3.txt Eğitim eylul3.txt Benzerlik Oranı: 99.359  
Test eylul3.txt Eğitim eylul2.txt Benzerlik Oranı: 33.524  
Test eylul3.txt Eğitim ask2.txt Benzerlik Oranı: 25.922  
Test eylul3.txt Eğitim eylul.txt Benzerlik Oranı: 25.919  
Test eylul3.txt Eğitim calikusu3.txt Benzerlik Oranı: 20.792  
Test kayipSembol.txt Eğitim kayipSembol.txt Benzerlik Oranı: 99.31  
Test kayipSembol.txt Eğitim kayipSembol2.txt Benzerlik Oranı: 67.735  
Test kayipSembol.txt Eğitim dijitalKale.txt Benzerlik Oranı: 23.076  
Test kayipSembol.txt Eğitim dijitalKale3.txt Benzerlik Oranı: 17.448  
Test kayipSembol.txt Eğitim olasiliksiz2.txt Benzerlik Oranı: 15.406
```


Program Arayüzü

Arayüz oluşturulurken PyQt5 kütüphanesi kullanılmıştır.

“Yeni Eğitim Seti Oluştur” ile farklı eğitim setinin kelime kökleri, frekansları bulunarak yeni arff uzantılı veri seti oluşturur.

“Yeni Test Seti Oluştur” ile farklı test setleri arff uzantılı olarak oluşturulur.

“Eğitim Seti Seç” ve “Test Setini Seç” arff uzantılı veri seçilip k değeri ile benzerlik hesaplanır.

Text Similarity

Arff uzantılı yeni eğitim setinin adı:

Yeni Eğitim Seti Oluştur

Arff uzantılı yeni test setinin adı:

Bağlı Eğitim Setini Seç

Yeni Test Seti Oluştur

Benzerlik Hesaplama İşlemleri:

Eğitim Setini Seç

Test Setini Seç

K sayısı:

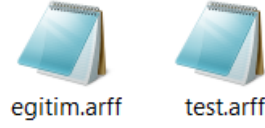
Benzerlik Hesapla

C:/Users/saruhan/Desktop/egitim.arff
C:/Users/saruhan/Desktop/test.arff

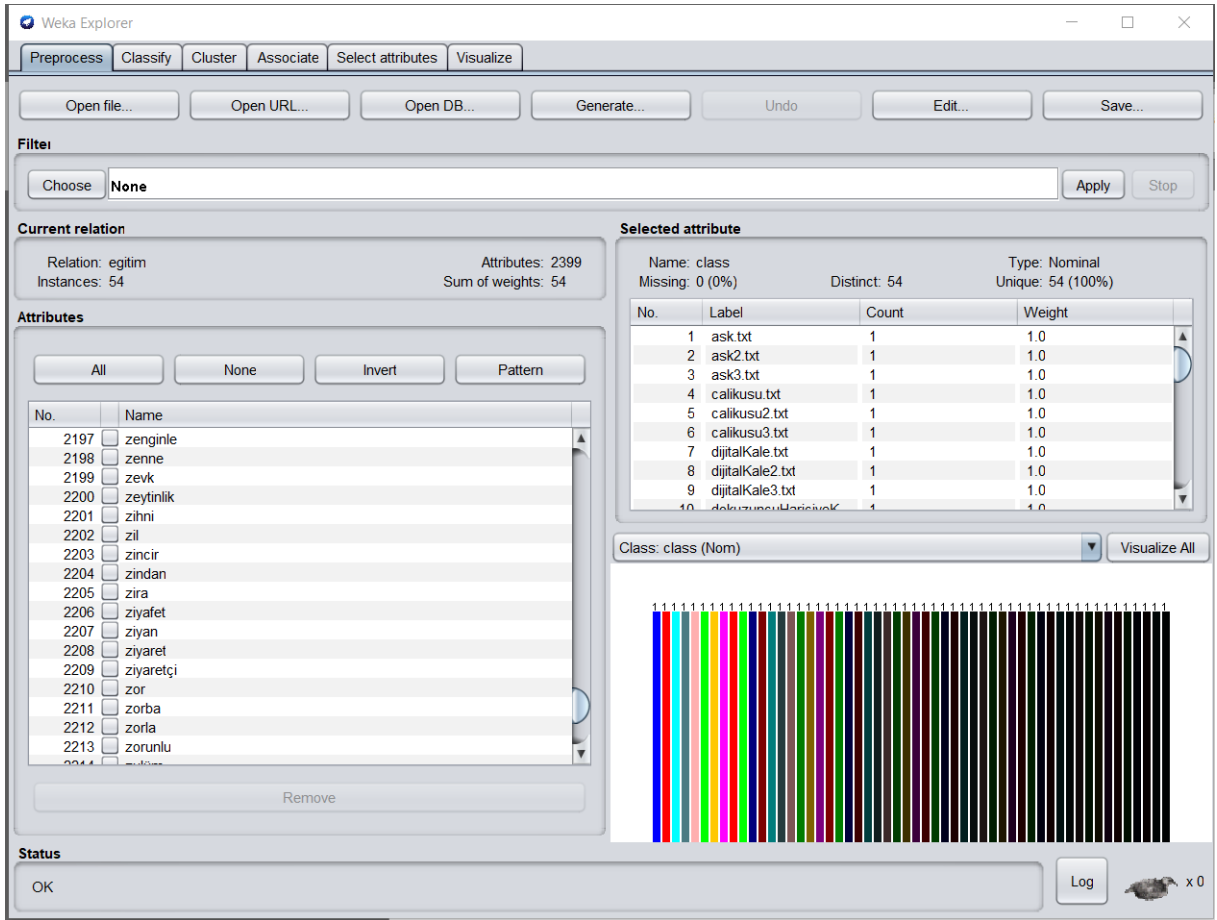
['ask.txt', 'ask.txt', 99.313]
['ask.txt', 'ask2.txt', 26.319]
['ask.txt', 'ask3.txt', 23.561]
['ask.txt', 'kurkMantoluMadonna3.txt', 20.855]
['ask.txt', 'kuyucakliYusuf.txt', 19.686]
['calikusu.txt', 'calikusu.txt', 99.378]
['calikusu.txt', 'calikusu3.txt', 48.681]
['calikusu.txt', 'calikusu2.txt', 43.418]
['calikusu.txt', 'ucurtmaAvcisi2.txt', 21.471]
['calikusu.txt', 'yaban.txt', 20.427]
['dijitalKale.txt', 'dijitalKale.txt', 99.391]
['dijitalKale.txt', 'dijitalKale3.txt', 35.189]
['dijitalKale.txt', 'dijitalKale2.txt', 30.872]
['dijitalKale.txt', 'kayipSembol.txt', 23.049]
['dijitalKale.txt', 'ask.txt', 15.316]
['dokuzuncuHariciyeKogusu.txt', 'dokuzuncuHariciy
['dokuzuncuHariciyeKogusu.txt', 'dokuzuncuHariciy

Sonuç

İlk defa çalıştırılan program 5 dakika gibi bir sürede metin analizini yapıp arff veri setini oluşturarak benzerlik oranı hesaplarken oluşturulan veri setleri tekrar kullanıldığında 2 saniye gibi bir sürede benzerlik oranını bulur.



Egitim.arff verisinin Weka da görüntüsü:



Current relation

Relation: egitim
Instances: 54
Attributes: 2399
Sum of weights: 54

Attributes

All None Invert Pattern

No.	Name
2197	zenginle
2198	zenne
2199	zevk
2200	zeytinlik
2201	zihni
2202	zil
2203	zincir
2204	zindan
2205	zira
2206	ziyafet
2207	ziyan
2208	ziyaret
2209	ziyaretçi
2210	zor
2211	zorba
2212	zorla
2213	zorunlu

Selected attribute

Name: class
Missing: 0 (0%)
Distinct: 54
Type: Nominal
Unique: 54 (100%)

No.	Label	Count	Weight
1	ask.txt	1	1.0
2	ask2.txt	1	1.0
3	ask3.txt	1	1.0
4	calikusu.txt	1	1.0
5	calikusu2.txt	1	1.0
6	calikusu3.txt	1	1.0
7	dijitalKale.txt	1	1.0
8	dijitalKale2.txt	1	1.0
9	dijitalKale3.txt	1	1.0
10	delusoryHaricK	1	1.0

Class: class (Nom) Visualize All

Status

OK Log x 0