T.C.

HALİÇ ÜNİVERSİTESİ

FEN BİLİMLER ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

TRİGRAM ÖZELLİK VERİ SETİ KULLANILARAK SINIFLANDIRMA YÖNTEMLERİYLE DİL TANIMA

YÜKSEK LİSANS TEZİ

Hazırlayan
Şengül BAYRAK

Danışmanlar
Prof. Dr. Mübariz EMİNLİ
Dr. Hidayet TAKÇI

İstanbul – 2011

T.C.

HALİÇ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı Tezli Yüksek Lisans öğrencisi **Şengül BAYRAK** tarafından hazırlanan **"Trigram Özellik Veri Seti Kullanılarak Sınıflandırma Yöntemleriyle Dil Tanıma"** adlı bu çalışma jürimizce Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Tarihi : 21.06.2011

(Jüri Üyesinin Ünvanı , Adı , Soyadı ve Kurumu) :	İmzası :
Jüri Üyesi: Prof.Dr.Mübariz EMİNLİ Dan.–HAL.Üniv.Bilgisayar Müh. ABD Öğr.Üyesi	.....
Jüri Üyesi : Dr.Hidayet TAKÇI 2.Dan. GYTE Öğr.Üyesi	.....
Jüri Üyesi : Yrd.Doç.Dr.Oğuz KARAN HAL.Üniv. Bilgisayar Müh.ABD Öğr.Üyesi	.....
Jüri Üyesi : Yrd.Doç.Dr.Ulviye HACIYEVA HAL.Üniv. Bilgisayar Müh.ABD Öğr.Üyesi	.....
Jüri Üyesi : Yrd.Doç.Dr.Murat BEKEN HAL.Üniv. Matematik Müh.ABD Öğr.Üyesi	.....
Jüri Üyesi : Prof.Dr.NAriman ŞERİFOĞLU HAL.Üniv. Elektrik-Elek.Müh.ABD Öğr.Üyesi (Yedek)	.....
Jüri Üyesi : Yrd.Doç.Dr.Osman ALİEFENDİOĞLU HAL.Üniv. Beykent Üniv.Öğr.Üyesi (Yedek)	.....

ÖNSÖZ

Tez çalışmalarım boyunca elinden gelen desteği ve bilgisini esirgemeyen, saygıdeğer hocalarım Prof. Dr. Mübariz Eminli'ye ve Gebze Yüksek Teknolojileri Bilgisayar Mühendisliği Bölümü'nün çok değerli hocası Öğr. Gör. Dr. Hidayet Takçı'ya;

Tez çalışmalarımda bana büyük sabırla yardım eden arkadaşım Arş. Gör. Yiğit Efe Yücesoy'a,

Manevi desteklerini ve sevgisini her zaman yanımda hissettiğim aileme ve Murat Hayta Bey'e teşekkürü bir borç bilirim.

İstanbul, 2011 ŞENGÜL BAYRAK

İÇİNDEKİLER

Sayfa No.

KISALTMALAR LİSESİ

DVM	Destek Vektör Makinaları
YSA	Yapay Sinir Ağları
BCO	Bulanık C OrtalamalarAlgoritması(Fuzzy C Means-FCM)
DVM	Destek Vektör Makinaları
NB	Naive Bayesian
CB	Centroid Based Sınıflayıcı
HMM	Hidden Markov Models
NLP	Natural Language Processing(Doğal Dil İşleme)
KTDT	Karakter Tabanlı Dil Tanıma
ANSI	American National Standards Institute
ECI	European Corpus Initiative (Dil tanıma deneylerinde kullanılan çok dilli küliyat)
$\vec{C}_i$	i. dil için centroid (merkez) değeri

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

GENEL BİLGİLER

Adı ve Soyadı : Şengül BAYRAK
Anabilim Dalı : Bilgisayar Mühendisliği
Programı : Bilgisayar Mühendisliği
Tez Danışmanları : Prof. Dr. Mübariz EMİNLİ, Dr. Hidayet TAKÇI
Tez Türü ve Tarihi : Yüksek Lisans – Haziran 2011

TRİGRAM ÖZELLİK VERİ SETİ KULLANILARAK SINIFLANDIRMA YÖNTEMLERİYLE DİL TANIMA

ÖZET

Doküman anlamanın birinci adımı doküman dilinin tanınmasıdır. Dil tanımanın amacı; dili bilinmeyen metinleri işlemek ve onları tanımlamaktır. Dokümanlar için dil bulma işlemi bir bakıma üst veri üretimi olarakta görülebilir. Dil tanıma sırasında; dokümanları sunacak sınıfları elde edebilmek için dokümandaki kelimelerin frekans değerleri kullanılır. Ayrıca dili bilinmeyen test dokümanlarının dilini bulmak için de dokümanın terim-doküman matrisi ile dil arasındaki benzerlikler bulunur. En yüksek benzerliği veren sınıf yeni dokümanın sınıfı olarak belirlenir. Böylece dil tanıma işlemi tamamlanmış olur. İstatistiksel dil tanıma olarak bilinen bu yöntem metin içeriğinden bağımsız dil tanımayı destekler. Dil tanıma, dilin ayırt edici özelliklerine sınıflandırma algoritmaları uygulanması ile gerçekleştirilmektedir. Bu kapsamda; dili tanımlayan, dilin özelliklerini sunmada ve özellikler arası ilişkilerin açığa çıkarılmasında kullanılan temel iki yöntem vardır, bunlar, dilbilimsel yöntemler ve istatistiksel yöntemlerdir (harf kombinasyonları, n-gram yöntemi, markov modelleri, bayesian ve vektör uzayı). Bunlardan istatistiksel yöntemde, dilin istatistiksel özellikleri kullanılır, dilbilimsel yöntemde ise dillere ait karakteristik özellikler kullanılır.

Sınıflandırma ve kümeleme algoritmalarıyla metin tabanlı dil tanımadaki performans analizini öneren sistemimiz eğitimi ve testi için, European Corpus Initiative (ECI) adı verilen uluslar arası kabul görmüş, çok dilli bir külliyat kullanılmıştır. Eğitim için ECI CDROM külliyatından, 1 KB ile 100 KB arasında

uzunluklarda 15 dil için (Türkçe, İngilizce, Almanca, Hollandaca, Fransızca, İtalyanca, Cezayirce, İspanyolca, Portekizce, Norveççe, Maltaca, Latince, Litvanyaca, İsveççe, Andoa Dili) alt külliyatlar kullanılmıştır.

Bu çalışmada doküman dili tanıma için n-gram tabanlı istatistiksel bir yöntem kullanılmaktadır. Yöntem; n-gram sıklıklarının dokümanın dilini tanımada kullanılabileceği temeline dayanmaktadır ve 26 harfi esas alan, trigram özellik kümesi ile çalışarak 300 öznitelik frekans değeri yöntemlere giriş olarak kullanılmıştır. Dolayısıyla Latin alfabesini kullanan diller ve Avrupa dillerinin tanınması için bir çözüm geliştirilmeye çalışılmıştır. Bu çalışmada, trigram seçimi, eğitim seti boyutu ve seçilen sınıflandırma algoritmalarının başarısı gibi parametreler esas alınarak test çalışmaları yapılmıştır. Eğitim setinin oluşturulmasında kullanılan N-Gram Özellik Seçimi Yöntemi, Profil Tabanlı Yöntem, Örnek Tabanlı Yöntem, Centroid Tabanlı Sınflayıcı, Bulanık C Ortalamalar Algoritması C# ortamında implemente edilirken, Yapay Sinir Ağları ve Destek Vektör Makinaları sınıflandırma algoritmaları ise Tanagra ve Weka veri madenciliği yazılımları kullanılarak eğitilerek test edilmiş ve sınflandırma başarıları doğruluk oranlarına göre verilmiştir.

Anahtar Kelimeler: Dil Tanıma, N-gram Özellik Çıkarım Metodu, YSA, DVM, BCO, K-Ortalamalar Algortiması.

GENERAL KNOWLEDGE

Name and Surname: Şengül BAYRAK
Field : Computer Engineering
Program : Computer Engineering
Supervisor : Prof. Dr. Mübariz EMİNLİ, Dr. Hidayet TAKÇI
Degree Awarded and Date : Master – June 2011

LANGUAGE IDENTIFICATION USING TRIGRAM FEATURE DATA SET WITH CLASSIFICATION ALGORITHMS

ABSTRACT

The first step of understanding the documents is identifying the language. The purpose of identifying the language, processing and describing unknown texts. Finding language for documents can be seen as the production of metadata. During the language identification; to obtain the During the language identification; to obtain the class which will present the documents use the frequencies. In addition, for finding unknown documents' language, obtain similarity between term-documents matrix and language. The highest similarity is as the class a new document class and so language identification process is completed. This method is known as statistical language identification, text support, regardless of content.Language identification, obtains with applying the algorithms to languages' distinctive features. In this context, describing of the language, providing the features and specifications for the removal of the basic relations between the two methods that are linguistic methods, and statistical methods (combination of letters, the n-gram method, markov models,

bayesian classifier, and vector space). In statistical method is used statistical properties of language but linguistic method is used characteristics of languages.

Our proposed method for training and testing, the European Corpus Initiative (ECI) which the internationally recognized name, used in a multilingual corpus. For training CD-ROM for the ECI corpus, lengths between 1 KB and 100 KB for the language of 15 (in Turkish, English, German, Dutch, French, Italian, Cezayirce, Spanish, Portuguese, Norwegian, Maltese, Latin, Lithuanian, Swedish, Andoa Language) sub-digests used.

In this study,using n-gram based method for language identification. Method, n-gram frequencies can be used in identifying the language of the document is based on and 26 letters is based on for working with trigram feature set. Therefore, a solution has been developed for languages using the Latin alphabet and European languages. In this study, the trigram selection, training set size and classification tests success are conducted on the basis of parameters. Tanagra and Weka's data mining software used in testing and training procedures. For preparation training set is used of N-Gram Feature Selection Method, Profile-Based Method, Example-Based Method, Centroid-Based Classify, Fuzzy C Means Algorithm is implemented C# programming language, Artificial Neural Networks and Support Vector Machines classification algorithms in the Tanagra and the Weka data mining software using the training of the classification success rates have been tested and is based on accuracy.

Keywords: Language Identification, N-Gram Based Feature Extraction Method, ANN,SVM,FCM,K-OrtalamalarAlgortihms.

1. GİRİŞ

Çağımızın en önemli ilerlemelerinden birisi iletişim alanında yaşanmaktadır. İletişimin en önemli unsuru olarak dilin bu ilerlemelerden uzak kalması düşünülemez. İnternet sayesinde daha da küçülen dünyamızda en büyük iletişim engeli maalesef dildir. Dilini bilmediğimiz bir kültürden, dili bilinmeyen bir teknolojiden faydalanmak mümkün değildir. Bu konuda önemli çabaları olarak doğal dil işleme alanı için anahtar rol ise dil tanıma alanındadır. Dolayısıyla, iletişimin başarısı için çok sayıda dili, doğru ve hızlı şekilde tanıyan sistemlere ihtiyacımız vardır. Örneğin, bilimsel bir araştırma yapmak isteyen kişi ancak bildiği dillerde yayın yapan sitelerden faydalanabilmektedir. Ayrıca, birden çok ülkeye hizmet veren firmaların da müşterilerinin dilini anlamak ve onlara hizmet verebilmek gibi bir mecburiyetleri vardır. Bu ve buna benzer çalışmalar dil tanımanın önemini artırmaktadır.

Dil tanıma problemi çözülmüş problemler arasında yer almakla birlikte, anlık mesajlaşmalarda dil tanıma, kısa metinlerde dil tanıma ve daha yüksek doğrulukta dil tanıma ihtiyacı hala vardır. Ayrıca, dil tanıma amacıyla geliştirilecek dil modelleri diğer metin madenciliği alanları için de örnek teşkil edebilmektedir.

Dil tanıma; doğal dil işleme alanından bir konudur. Doğal dil işleme ve yapılan araştırmalar, bir yanda işlenen dilin yapısal özelliklerinden bağımsız olma iddiasında kuramlar geliştirirken, bir yandan da bunların geniş kapsamlı olarak uygulanması için işlenecek dillere özel kaynakların üzerine yoğunlaşılmaktadır. Dil tanıma, konuşma dili tanıma, doküman tabanlı dil tanıma olmak üzere ikiye ayrılır: Konuşma dilinde dil tanıma yapmak için sinyal işleme teknikleri kullanılırken, metin tabanlı dil tanımada sembolik harflerle işlem yapılır. Konuşma sırasında dilin tanınması zor bir işlemken, metin tabanlı dil tanıma daha geniş alanda uygulanabilir. Çalışmamız yazılı dokümanların dilini tanıma üzerinedir. Yazılı dokümanların dilini tanımada genellikle, dile ait özellikler istatistiksel açıdan ele alınır. İstatistiksel dil tanıma adıyla da bilinen yöntem çalışmamızda da tercih edilmiştir.

Çalışmada dil özelliklerini sunmak için n-gram özellik çıkarımı yöntemi seçilmiştir. Önce eğitim verilerinden en önemli n-gramlar seçilmiş daha sonra ise bu nitelikler ile döküman ve dil modelleri kurulmuştur. Bu model sayesinde her bir doküman en küçük birimlere ayrılıp (unigram, bigram, trigram, quadram) oluşan kısa karakter frekansları bulunmuştur. Bulunan n-gramlar sıklıklarına göre sıralanıp eğitim seti oluşturulmuştur.

Eğitim setinde yer alan farklı diller için oluşturulmuş n-gramlarla dili bilinmeyen test metninde elde edilen n-gramlar karşılaştırılıp hangi sınıfa ait olduğu bulunmuş ve sınıflandırma, kümeleme algoritmaları uygulanarak performans analizi yapılmıştır. Böylece başarılı bir dil tanıma sistemi geliştirilmesi hedeflenmiştir.

N-gramdan oluşturulan 3 karakter uzunluklu (trigram) dizilerden eğitim verisinin, test verisi üzerindeki doğruluğu için sıralanan n-gram üzerine profil tabanlı metod (profile based method) ve örnek tabanlı metod (instance based method) uygulanmıştır. Profil tabanlı metot; test dokümanlarında geçen n-gramları, profil dosyalarındaki n-gramlar ile karşılaştırmış ve buradan bir skor elde etmiştir. Burada elde edilen veriler, profil tabanlı yöntemlerle eğitilerek dil tanıma başarıları test edilmiştir. Örnek tabanlı metot, test verisini bölümleme ve her bir parça için n-gram bulma şeklinde yapılır Burada elde edilen veriler YSA gibi makine öğrenimi algoritmalarında kullanılır. Dolayısıyla; profil tabanlı metotta, her bir sınıf için profil adı verilen tek bir kayıt, dile ait özellikleri tutarken, örnek tabanlı metot her bir sınıfa ait özellikleri birden fazla kayıtta tutar.

Bu çalışmada; dil tanıma iki farklı öğrenim modeli kullanılarak gerçekleştirilmiştir. Bunlardan birincisi profil tabanlı yöntemler olup; profil tabanlı yöntem olarak Centroid Sınıflayıcı, k-Ortalamalar ve Bulanık Kümeleme kullanılmıştır. Diğeri ise örnek tabanlı yöntem olup bu kapsamda; YSA ve DVM sınıflandırma algoritmaları kullanılmıştır. Dil tanıma problemi her iki tür içinde uygundur. Daha önce yapılan çalışmalarda; Centroid sınıflayıcının başarısı görüldüğü için ona benzer iki yöntem olan K-Ortalamalar ve Bulanık Kümeleme tercih edilmiştir. Ayrıca, metin madenciliği çalışmalarında sıklıkla kullanılan DVM ve YSA'da bu çalışmada iki önemli alternatif olarak yer bulmuştur. Çalışma bu yönüyle profil tabanlı ve örnek tabanlı yöntemler için bir karşılaştırma çalışması şeklinde görülebilir.

1.1. Dil Tanıma Nedir?

Dil tanıma, dili bilinmeyen bir dokümanın, dilbilimsel özellikler ve algoritmalar kullanılarak tayin edilmesi işlemidir. İstatistiksel dil tanıma, yakınlıklara veya benzerliklere dayalı olarak yapıldığından doküman sınıflandırmada bazen hatalar olabilmektedir. Örneğin, aslında Almanca olan bir metin Hollanda dilinde gibi sınıflanabilmektedir. Dil tanıma; tahminci değişkenleri dile ait özellikler, sınıf etiketi dil olan metin sınıflandırma problemi olarak görülebilir. Dolayısıyla metin sınıflandırma teknik ve yaklaşımları ile çözülebilir. Problemin çözümü için dile ait özelliklerin neler olduğu ve algoritmaların neler olabileceği önemli konulardır.

1940 yıllarından beri doğal dillerin biçimsel ve karakteristik özellikleri incelenir. Bilgi alma etki alanında vektör uzayı modeli ve olasılıksal model gibi başarılı modeller bulunmakla birlikte, dil tanıma modelleri daha çok konuşma tanıma etki alanından alınmıştır (Salton ve McGill, 1983)(Robertson ve Sparck, 1976). Dil tanımada kullanılan birçok istatistiksel model önce konuşma tanıma konusuna uygulanmıştır. Shannon (Shannon, 1948) tarafından geliştirilen kelime ve harf dizileri kullanımı bunlara bir örnektir. Shannon, İngilizce dilini araştırmış, dilin düzensizliğini ve tahmin edilebilirliğini araştırmıştır. Aynı zamanda n-gram modelleri ve Hidden Markov Modeli (HMM) sıklıkla dil tanımada kullanılmıştır. Geliştirilen dil modelleri bilgi alma etki alanında da kullanılmıştır. Sorgu sonuçlarının iyileştirilmesi bilgi almada dil modellerinin kullanımına bir örnektir. Zipf (Zipf, 1949) tüm istatistiksel yöntemlere uygulanabilecek bir yöntem önermiştir. Bilgisayar teknolojisinin hızla gelişmesiyle birlikte daha fazla bilgi toplanmış ve Zipf ile Shannon araştırmaları kullanılarak yeni teknolojiler geliştirilmiştir. Böylece dil tanıma, bilgi şifreleme işlemleri, optik karakter tanıma(OCR), konuşma tanıma, yazı doğrulama işlemlerinde kullanılmıştır.

Dil tanıma çalışmalarında n-gram yönteminin kullanılması başarılı sonuçlar üretmiştir. N-gram yöntemi metin sınıflandırma için de kullanılan bir yöntemdir. Bu yöntem dil tanıma ile metin sınıflandırma çalışmalarının aynı tekniklerle yapılabilirliğini göstermesi bakımından önemlidir. Cavnar and Trenkle (Cavnar ve Trenkle, 1994) tarafından geliştirilen n-gram tabanlı metin sınıflandırma modeli

üzerine inşa edilen Textcat isimli dil tanıma uygulaması, bugün dil tanımada n-gram tabanlı metin sınıflandırma için en çok başvurulan etkin kaynaklardan biri olmuştur.

1.2. Dil Tanıma Neden Gereklidir?

Dünya bilgi çağına girdiğinden beri, gelişmiş ülkelerde kişiler tarafından kullanılan bilginin miktarı çok hızlı bir şekilde artmaktadır. Bu bilgileri aktif bir şekilde kullanabilmek ve kısa sürede erişebilmek için birbirleri ile ilişkili olan bilgileri bulup aynı bilgi topluluğu içinde toplamak gerekir. Bu da dokümanları sınıflandırmayı gerektirir. Doküman sınıflandırmadaki amaç, bir dokümanın özelliklerine bakarak, önceden belirlenmiş belli sayıdaki kategorilerden hangisine dâhil olacağını belirlemektir. Doküman sınıflandırma bilgi alma (information retrieval), bilgi çıkarma (information extraction), doküman indeksleme, doküman filtreleme, otomatik olarak metaveri elde etme ve web sayfalarını hiyerarşik olarak düzenleme gibi pek çok alanla ilişki içindedir. Geçmiş yıllarda doküman sınıflandırma insan eliyle yapılır ve doğru kategoriye atama yapabilmek için de dokümanlar mutlaka okunurdu. Oysaki doküman sınıflandırma işlemi manuel olarak değil de elektronik ortamda, önerilen bazı yöntemler kullanılarak yapılırsa sınıflandırma için harcanan zamanda çok kısalmış olur.

Etnolojik araştırmalar sonucu şu an dünyada yaklaşık 6800 dil aktif olarak konuşulmaktadır (Xerox, 2005) Geçerli dil tanıma sistemleri ise bu dillerden ancak 250-260 tanesini tanıma yeteneğine sahiptir (Hiemstra ve Vries). Son yıllarda hızlı bir şekilde artış gösteren çok dilli dokümanların varlığı, dil tanıma konusunda yeni çalışmaları zorunlu hale getirmiştir. Hızla büyüyen doküman yığınları içinde, dokümanların içeriğinin belirlenmesi ve sorgulamaların yapılabilmesi önemlidir. İnternet üzerinde insanların birbiriyle anlaşabilmesi ve otomatik cevap sistemlerindeki elektronik ortamlarda problemin çözümü ve verinin içeriğinin anlaşılması önemlidir. Uluslararası üretim yapan firmaların, ürünlerini pazarlayacağı ülkenin dilini desteklemesi için de dil tanıma önemlidir. Askeri yazılımlarda, internetten bilgi transferinde bilgilerin farklı bir dilde şifrelenerek gönderilmesinde dil tanıma gereklidir. Çeviri sistemlerinde doğal dil işleme ve konuşma tanımada ilk adım dil tanımadır. Bu yüzden, dil tanıma, çeviri sistemleri ve konuşma tanıma ile yakından ilgilidir. Geliştirilen ilk dil tanıma sistemleri konuşma tanıma teknikleriyle

yerine getirilmiştir. Çeviri sistemlerinin dil tanıma konusunda iki temel stratejisi bulunmakta olup, bu stratejilerden birincisi; dil tanımada özel harflerin kullanımına dayalıdır (Newman, 1987). Bu tip dil tanımada özel harflerden bir tanıma listesi oluşturulmakta ve hızlıca dil tanıma işlemi yerine getirilebilmektedir. İkinci strateji, çeşitli dillerde kullanılan kısa kelimelerin bir listesi ile çalışma temeline dayalıdır. Dilin tanınabilmesi için dile ait kısa kelimeler aranır. Kimi zamanda kelimelerin hangi harf dizileriyle son bulduğu türünden bilgilere bakılır.

Tablo 1.1 : UNESCO ve BMT 'de Kabul Edilen Resmi Diller

Dil	**Kullanan kişi sayısı**
İngilizce	400 milyon
Fransızca	150-200 milyon
Arapça	180 milyon
İspanyolca	300 milyon
Rusça	280 milyon
Çince	1 milyar
Türk Dili ve Lehçeleri	150 milyon

1.3.Dil Tanıma Problemleri

Doküman dili tanıma her ne kadar uzun zamandır çalışılan bir konu olsa da, dil tanıma konusunda, sınıflandırma doğruluğunu artırmak gibi hala çalışılması gereken faktörler yer almaktadır. Bu faktörlerden bazıları:

Tanımlanması gereken dokümanın boyutu: Küçük boyutlu dokümanlar, dile ait özellikleri yansıtmadıkları için onların tanınması zordur. Örneğin, 10 ile 30 kelime sınırı olan kısa mesajlarda (SMS) dil tanıma zordur.

Eğitim verisinin boyutu ve çeşitliliği: Dünya dillerindeki sözlüklerde milyonlarca kelime arasında dilleri yansıtan önemli ayırt edici kelimeleri seçmek gerekir. Bu şekilde sınıflandırmada, daha az önemli kelimeleri ayırt ederek başarı oranını artırabiliriz.

Üzerinde çalışılan sınıflandırma algoritması: Belirlenecek bir metin, her n-gram yineleme sayısını bir vektörün bileşenleri olarak düşünülebilir. Bu vektörler sınıflandırmada uygulanabilecek potansiyel verilerdir.
Birbiri ile akraba olan diller arasında ayrım yapmak zordur.

1.4. Tezdeki Çalışmalar

Dil tanıma problem bir sınıflandırma problemidir. O yüzden problem eğitim ve test adımlarından oluşur. Dil tanıma problemi için eğitim, dili sunan sınıf profillerinin doğru şekilde elde edilmesi ve sunumudur. Eğitim sırasında metinsel dokümanlar frekans yöntemiyle özetlenmiş ve böylece sınıf profilleri elde edilmiştir. Dili tanımlayan, dilin özelliklerini sunmada ve özellikler arası ilişkilerin açığa çıkarılmasında n-gram yönteminden trigram frekansları kullanılmıştır. Yani 60 harfi esas alan, trigram özellik kümesinden 60*60*60 harfin tüm kombinasyonları hesaplanarak öznitelik frekansı elde edilmiştir. Bu hesaplama neticesinde, elde edilen trigram kombinasyonlarının frekans değerleri quick sort(hızlı arama algoritması) kullanılarak büyükten küçüğe sıralanmıştır. Tanagra veri madenciliği yazılımı kullanılarak Fisher filtering yöntemiyle en önemli ilk 300 trigram seçilmiştir. Böylece eğitim verilerinden en önemli n-gramlar seçilmiş daha sonra ise bu nitelikler ile döküman ve dil modelleri kurulmuştur. Bu model sayesinde her bir doküman en küçük birimlere ayrılıp oluşan kısa karakter frekansları bulunmuştur. Bulunan n-gramlar sıklıklarına göre sıralanıp ve eğitim seti oluşturulmuştur. Eğitim setinde yer alan farklı diller için oluşturulmuş n-gramlarla dili bilinmeyen test metninde elde edilen n-gramlar karşılaştırılıp hangi sınıfa ait olduğu bulunmuş ve sınıflandırma, kümeleme algoritmaları uygulanarak performans analizi yapılmıştır. Böylece başarılı bir dil tanıma sistemi geliştirilmesi hedeflenmiştir.

Dil tanıma iki farklı öğrenim modeli kullanılarak gerçekleştirilmiştir. Bunlardan birincisi profil tabanlı yöntem olup; profil tabanlı yöntem için; Centroid Classifier, K-Ortalamalar ve Bulanık Kümelemekullanılmıştır. Diğeri ise örnek tabanlı yöntem olup bu kapsamda; YSA ve DVM algoritma kullanılmıştır. DVM algoritması olarak bu çalışmada C-SVC tercih edilmiştir. C-SVC çok sınıflı problemler için daha uygundur. Dil tanıma problemi her iki tür için de uygundur. Daha önce yapılan çalışmalarda; centroid sınıflayıcının başarısı görüldüğü için ona

benzer iki yöntem olan K-Ortalamalar ve Bulanık Kümelemetercih edilmiştir. Ayrıca, metin madenciliği çalışmalarında sıklıkla kullanılan DVM ve YSA'da bu çalışmada iki önemli alternatif olarak yer bulmuştur. Çalışma bu yönüyle profil tabanlı ve örnek tabanlı yöntemler için bir karşılaştırma çalışması şeklinde görülebilir.

Profil tabanlı metot; test dokümanlarında geçen n-gramları, profil dosyalarındaki n-gramlar ile karşılaştırmış ve buradan bir skor elde etmiştir. Burada elde edilen veriler, profil tabanlı yöntemlerle eğitilerek dil tanıma başarıları test edilmiştir. Örnek tabanlı metot, test verisini bölümleme ve her bir parça için n-gram bulma şeklinde yapılır. Dolayısıyla; profil tabanlı metotta, her bir sınıf için profil adı verilen tek bir kayıt, dile ait özellikleri tutarken, örnek tabanlı metot her bir sınıfa ait özellikleri birden fazla kayıtta tutulmuştur.

Bu çalışmada dosya boyutu 1 KB ve 100 KB lık verilere çeşitli yöntemler uygulanarak dili bilinmeyen metin üzerinde performans karşılaştırması yapılması hedeflenmiştir. Ayrıca dili bilinmeyen test dokümanlarının dilini bulmak için de dokümanın terim-doküman matrisi ile dil arasındaki benzerlikler bulunur. En yüksek benzerliği veren sınıf yeni dokümanın sınıfı olarak belirlenmiştir.

2. DOKÜMAN TABANLI DİL TANIMA

Dil tanıma işlemine geçmeden önce bazı çalışmaların yapılması gerekir. Bu tezin amacı, metin tabanlı dil tanıma yapmak ve dili ne kadar doğru tanımlayabileceğimize karar vermekte etken olan faktörleri bulmaktır. 15 resmi dil üzerinde, farklı sınıflandırma algoritmalarının dil tanıma başarıları doğruluk oranlarına göre hesap edilmiştir. Sınıflandırma işleminde, doküman verisinden hesaplanan n-gram istatistikleri kullanılarak profil tabanlı sınıflayıcı ve örnek tabanlı sınıflayıcı test edilmiştir.

Doküman dillerini tanımada takip edilmesi gereken bir yol vardır. Test metninin boyutu, eğitim verisinin doğruluğu ve çeşitliliği belirlenir. Sınıflandırma algoritması kullanılır. Benzer olan diller arasında sınıflandırma yapılır. Bu yolun şeklini, seçilen dil tanıma modeli belirler. Çalışmamızda doküman dilini tanımada izlediğimiz yol aşağıdaki gibidir:

- Eğitim ve test dokümanları için 100 KB ve 1 KB'lık doküman verisi, ECI külliyatından seçilir.
- Dokümanlar üzerinde gerekli ön işlem yapılır.
- Dokümanlar seçildikten sonra gerekli ön işlemin yapılması ve dokümanların dil tanımaya uygun hale getirilmeleri lazımdır. Bunun için ön işlem aşamasında dokümanlara temizleme ve dönüşüm uygulanır.
- Bütün dokümanlar bir vektör yardımıyla sunulur.

Sunum aşamasında özellik seçimi yapılır. Özellik seçimi dokümanı en iyi sunacak özelliklerin seçilmesi işlemidir. Dil tanıma sisteminin özellik kümesinin hangi elemanlardan oluştuğu önemli bir konudur. Özellik kümesi için genellikle harf veya kelime dizileri ile n-gramlar kullanılmaktadır. Bu aşamada özellik değerleri için eşik değeri kullanılabilir. Örneğin gramların biraraya gelme sıklığı 60'dan küçük olanlar elenmiştir.

Eğitim aşamasında her bir dili sunmak için bir vektör eğitim dokümanlarından elde edilir. Bu vektörler, örnek tabanlı ve profil tabanlı metodlar tarafından elde edilir. Eğitim aşamasında elde edilen ve dil kategorilerini ifade eden vektör, ortalama

özellik değerleriyle ilgiliyse centroid olarak isimlendirilir. Centroid değerleri her doküman kategorisi için bir tanedir. Test dokümanlarının dilini tespit için, test dokümanı vektörleri ile eğitim verisi vektörleri arasındaki benzerlikler hesaplanır. Örnek tabanlı vektörlerden elde edilen veriler, YSA, BCO, Centroid tabanlı sınıflayıcı algoritmalarına giriş olarak verilmiştir. Her dil için oluşturulan profil verileri baz alınarak dil tanıma başarı oranı tespit edilmiştir.

2.1. İlişkili Çalışmalar

Günümüze kadar dil tanıma ile ilgili çeşitli çalışmalar yapılmıştır. Beesley (Beesley,1988) İngilizce, İspanyolca, Fransızca ve Portekizce olmak üzere dört dil üzerinde çalışmıştır ve dil tanıma için 2, 3, 4 ve daha fazla harf kullanmıştır. Dunning (Dunning, 1994) İspanyolca ve İngilizce dilleri üzerinde çalışma yapmıştır ve geliştirilen dil tanıma sistemi için Bayesian sınıflandırıcı kullanmıştır ve çalışmada en ilgi çekici ve başarılı sonuçlar 50 KB eğitim verisi ve 20 byte test verisi için %92, 500 byte test verisi için %99,9 başarı elde edilmiştir. Dahası, 5 KB eğitim veri ve 500 byte test verileri için başarım oranı % 97'dir. Combrink ve Botha(Combrink ve Botha, 1995) tarafından yapılan çalışma metin sınıflandırma yöntemleri ile dil tanıma çalışmasının yapıldığı bir çalışmadır. Bu çalışmada 12 dil kullanılmış ve dilleri sınıflandırmak için Histogram Metodu kullanılmıştır. Bu çalışma ile ilgili bir sonuca rastlanmamıştır. En sık karakter dizeleri n-gram gibi kullanılır. Sistem, her dilin geçiş vektörler için birçok histogram kurar. Grafenstette (Grafenstette, 1995), çalışmasında trigram ve shortterm kullanmıştır ve bu iki yöntem için en büyük problem yüksek boyutlu özellik setidir. Trigram yöntemiyle yapılan testlerde 100 byte uzunluğundaki test verisi için %98.96 oranında başarı elde edilmiştir. Short term metodu ile yapılan testlerde ise 100 byte uzunluğundaki test metinleri için sınıflandırma doğruluğu %98.68'dir. N-Gram tabanlı bir başka çalışma Adams ve Resnik(Adams ve Resnik ,1997), dinamik tüm belgeler için World Wide Web sayfalarının dil etiketleri ekleyen bir sistem önermiştir. Bu nedenle, 5-gram 220 KB eğitim verisinden çıkarılmıştır ve her bir 100-500 byte test verisi için doğruluk değeri %98.68 elde edilmiştir. Aynı eğitim verisi 3-gram için uygulandığında %98.32 test başarısı elde edilmiştir. Prager (Prager, 1999), vektör uzayı modeli kullanarak Linguini sistemi önermiştir. Linguini eğitim textinden sözlük üretmiştir. Verilen özellik vektöründen dilin karar verilmesi için kosinüs benzerliği kullanır.

Özellik olarak karakter seviyesinde n-gram, kelimeler ve kombinasyonları kullanılmıştır. Sistem çalıştırıldığında sadece karakter n-gramlarında, 4-gram da en iyi sonuç üretmiştir. Kelimelerin kullanılması durumunda, uzun kelimeler kısa kelimeler daha iyi sonuç üretir. Xafopoulous ve ekibi dil tanımada, karakter dizileri için HMM(Hidden Markov Model) önermiştir. Önerilen bu sistem, İngilizce, Almanca, Fransızca, İspanyolca ve İtalyanca dillerine ait web üzerindeki dokümanlarını otomatik tanımlar. 140 byte test verisi için %99 doğruluk oranına erişilmiştir. Diğer bir önerilen sistem ise Takçı ve Soğukpınar (Takçı ve Soğukpınar , 2004)'ın İngilizce, Fransızca, Almanca ve Türkçe olmak üzere 4 farklı dil için 22 harf sisteminden oluşan centroid-based algoritması kullanarak tanımlama yapmasıdır. 500 KB eğitim verisi için %98 başarı oranına ulaşılmıştır. Test verisinın boyu küçüldükçe doğruluk oranı da düşmektedir. 2004 yılından sonra da bazı çalışmalar yapılmıştır. Fakat bu çalışmalar sözlük tabanlı yöntemler gibi daha çok dilbilimsel yöntemlerdir. Hâlbuki çalışmamızda aynı etki alanında dil tanıma işlemi yapan sistemler karşılaştırılmıştır. Daha doğru karşılaştırmalar için böyle çalışmalar yapılmıştır.

Tablo 2.1: Metinsel Tabanlı Dil Tanıma ile İlgili Yapılmış Çalışmalar

Deney Grubu	**Yıl**	**Çalışılan Diller**	**Kullanılan Yöntem**	**Başarım Oranı**
Dunning	1994	2 farklı dil	Bayesian Sınıflayıcı	%97
Combrink ve Botha	1995	12 farklı dil	Histogram Metodu	Sonuca ulaşılamamış
Grefenstte	1995	9 farklı dil	Trigram Metod, Short Term Metod	%98.96
Adams ve Resnik	1997		N-gram Based Metod	%98.32
Prager	1999	13 farklı dil	Vector Space Base Model	%98.2
Xafopoulous ve ekibi	2004	5 farklı dil	HMM Based Model	%99
Takçı ve Soğukpınar	2004	4 farklı dil	Centroid Based Model	%99

2.2. Metinsel Tabanlı Dil Tanıma Yöntemleri

Metin tabanlı dil tanıma yaklaşımları, dilbilimsel yöntem ve istatistiksel yöntem olarak ikiye ayrılır. Metin tabanlı dil tanıma yaklaşımların bir spektrum kullanılmaktadır ve dilsel derinliği olan en önemli ayırt edici faktör ile öne sürülerek tanınması yaygın bir çalışma haline gelmiştir. Metin tabanlı dil tanımadaki bir yaklaşım olan dilbilimsel yöntem, dillere ait dilbilgisi kurallarına göre dokümanlardaki dili tahmin eder. Doküman içinde yer alan kelimeleri arar ve bunlara sıklıklarına göre puanlama yapar. Türkçede büyük ünlü uyumuna göre sözcük dizilimi dilbilimsel dil tanıma için örnek olarak verilebilir. Metinsel tabanlı dilin dil bilimsel yönteminde yazının sadece parçalara ayrılması yetmez, onun syntax yapısı da metinsel yapısı da göz önüne alınır. Bu da dilin tanınmasında karmaşıklık boyutunu artırır. Bu dilbilimsel yaklaşımlar dilin tanınmasında çok iyi doğruluk verir. Fakat eğer dile ait büyük bir kaynak veri setine ihtiyaç olursa, bu durum hesaplamada pahalıya mal olabilir. Yazılmış bir metinden dil tanıma yapabilmek için en güzel yol, metnin dilin özelliklerini yansıtacak şekilde parçalara ayrılarak dili istatistiksel olarak ifade etmektir. Dili istatistiksel olarak ifade edebilmek için,

- harflerin sıralanışı
- belirli anahtar kelimelerin varlığı
- kısa kelimelerin frekansları(bir arada buluma durumu)Tekil ve geniş ayırt edici yazılar veya kısa karakter dizileri ile elde edilebilir.

Dil tanıma çalışmalarının önemli bir kısmında istatistiksel yöntemler kullanılmaktadır. İstatistiksel yaklaşıma, örüntü tanımadaki geleneksel algoritmalar kullanılmıştır. İstatistiksel yaklaşımda kullanılan iyi bir seçenek olan n-gram ve diğer yöntemler ileriki bölümlerde anlatılmıştır.

2.3. İstatistiksel Dil Tanıma Yöntemleri

İstatistiksel yöntemler, istatistiksel analiz kavramına dayanır, nesneleri bölümlemek için benzerlik ölçülerini kullanırlar ve sayısal veriler ile sınırlandırılırlar. İstatistiksel yöntemler, dil bilimci olmadan dil tanımayı mümkün hale getirmektedir fakat sistemin eğitimi söz konusudur. Dil tanımanın bu

yönteminde, dilleri ayırt etmede kullanılacak matematiksel prensiplerin neler olması gerektiği önemlidir (Dunning, 1994). İstatistiksel dil modelleri ile ilgili çalışmalar Andrei Markov'a kadar dayanır. Markov çalışmasında metinsel verilerdeki harf dizilerini modellemiştir (Manning, Schutze, 1999) . Bir diğer ünlü çalışma Claude Shannon tarafından harf ve kelime dizilerinin modellenmesi çalışmasıdır (Shannon, 1948).Yazılan metinden dil tanıma; yazılmış bir metinden dil tanıma yapabilmek için en güzel yol, metnin dili özelliklerini yansıtacak şekilde parçalara ayrılmasıdır.

İstatistiksel dil modeli terimlerin ağırlıkları veya olasılıkları üzerine kurulur. Bir sözlükte yer alan terimlerden bazısı diğer bazılarına göre daha önemlidir. Terimlerin önemi kimi zaman ağırlığı ile kimi zaman da olasılığı ile verilebilir. Önemi yüksek olan kelime ağırlığı veya olasılığı yüksek olan kelimedir. İstatistiksel dil modellemede terimlerin sırası önemli değildir, önemli olan terimlerin değeridir. Olasılığa dayalı istatistiksel dil modellerinde bazı notasyonlar kullanılmaktadır. Bu notasyonlar kısaca şöyledir. T dil içindeki bir terimi göstermek üzere P(T), T teriminin, P(T|D) ise D dokümanı içindeki T teriminin olasılığıdır. Dillerde hangi terimlerin önemli hangilerinin önemsiz olduğu bilinemez. O yüzden bir terimin önem derecesi λ ile verilebilir, λ için aralık $(0 \geq \lambda \geq 1)$ şeklindedir.

2.4. Özellik Vektörleri

Dili istatistiksel olarak ifade edebilmek için; harflerin sıralanışı, belirli anahtar kelimelerin varlığı, kısa kelimelerin frekansları(bir arada buluma durumu) belirleyicidir. Bu durum belirlenirken n-gram özellik çıkarımı yöntemi kullanılmıştır. Dokümanların birer özellik vektörü şeklinde gösterimi en çok kullanılan metotlardan biridir(Salton ve McGill, 1983).

2.4.1. N-Gram Özellik Çıkarımı Yöntemi

N-gram, bir karakter katarının n adet karakter dilimidir. N-gram tabanlı sınıflandırma
yöntemi, doküman içerisindeki n-gram karakterlerin kullanım sıklığına dayalı bir işlemdir. Bu
çalışmada, n-gram'ın 3-gram uzunluğu kullanılmıştır.

Metin tabanlı dil tanıma, elektronik ortamda yayınların tanınması doküman işlemede temel görevdir. OCR gibi, metinsel hatalar, telaffuz hatalar, gramer hatalarının, telaffuz hatalarının tanımındaki hataların bulunması da text categorization'ın işidir. Belirlenen n değerine göre oluşmuş ardışık harfler dizisidir. Kelime içerisinde, birbirine komşu harflerden oluşmuş bir dizi harfler topluluğudur. N-gram, Trenkle ve Cavnar(Trenkle ve Cavnar, 1994) çalışmalarında karakterler üzerinde n-gram özellik çıkarımı yöntemini bulmuştur. Literatürde n-gram tanımı terimlerin birbiri ardı sıra sıralanması anlamına gelse de bu çalışmada birbiri ile bitişik dizilerden meydana gelmiş dizilerdir. Bir kelimedeki n-gram, örtüşen kelime grubu olarak da temsil edilebilir. Kelime dizisindeki boşluk, noktalama işaretleri ve sayısal karakterler gözönünde bulundurulmamıştır. n-gram bir kelimenin, belirli bir değere göre, belirli bir kural doğrultusunda hecelenmesidir tanımı da yapılabilir. n-gram'ın farklı birkaç uzunluğu olarak 2-gram, 3-gram ve quad-gram'lar kullanılmıştır. "öğrenci" kelimesinin n-gramlarına ayrılmış hali Şekil 2.1. gibidir:

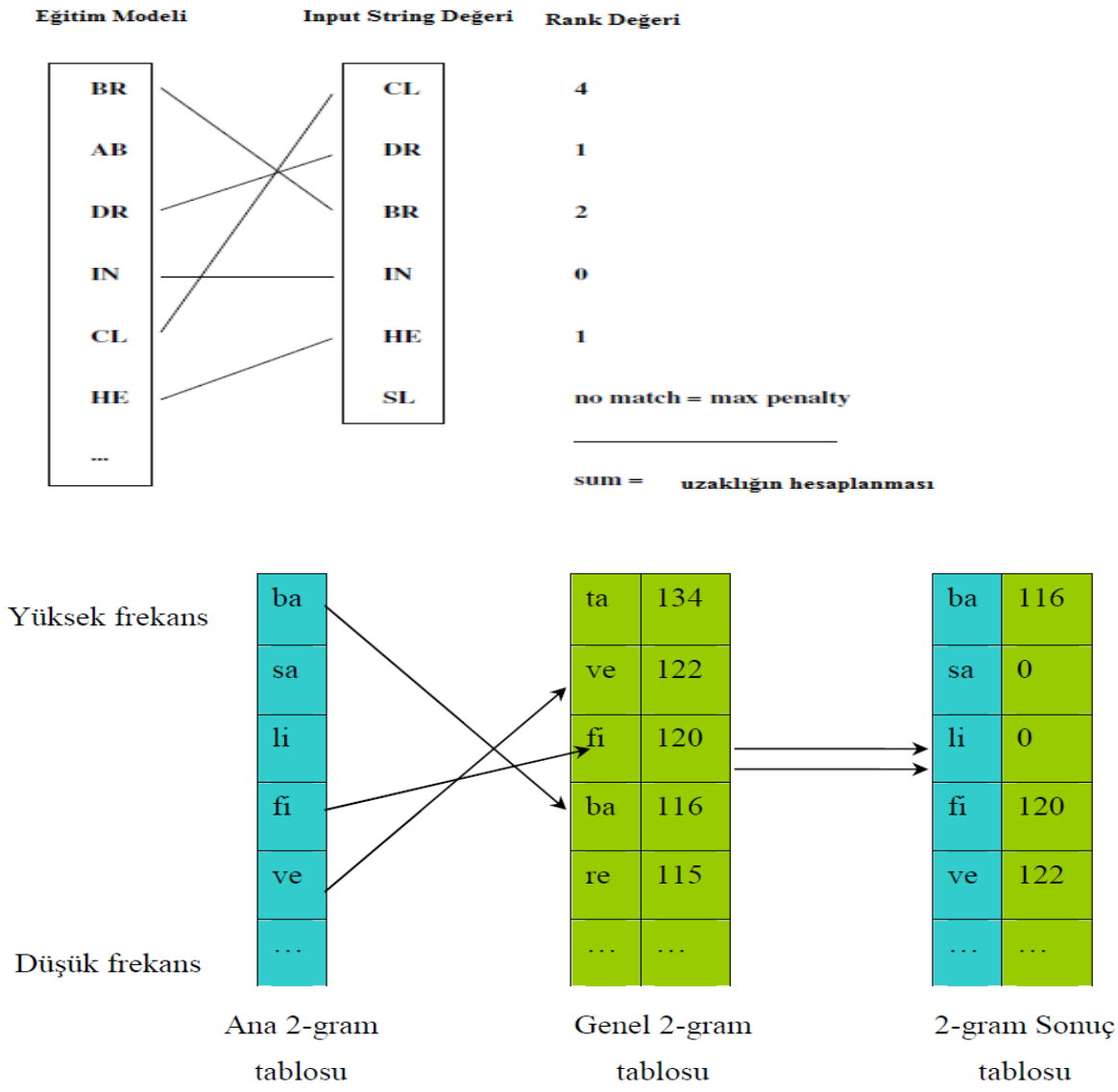
bi-grams	:	öğ	ğr	re	en	nc ci
tri-grams	:	öğr	ğre	ren	enc	nci
quad-gram	:	öğre	ğren	renc	enci	

Şekil 2.1: Karakter Dizisi Olan "öğrenci" İçin N-Gram Kombinasyonları

Her kelime farklı bir frekans ile gerçekleşir. Kelimedeki gramların geçme sıklığı o kelimenin frekansını ifade eder. N-gram ile küçük boyutlu verilerle hızlı bir sonuç alınır. Sistem ilk önce, n-gram frekanslarındaki profilleri karşılaştırıp hesaplama yapar. Çeşitli kategorilerde temsil edilen eğitim setini hesaplar. Sonra sistem dokümanda bulunan parçaları sınıflandırır. En sonunda sistem doküman profili ve her bir kategori profili arasında uzaklık ölçümü yapar. Hangi kategorinin profiline uzaklık olarak daha yakınsa ona atanır.

N-gram yöntemi, dokümanları sınıflandırmak için kullanılan basit ve güvenilebilir bir yöntemdir. Temel düşünce, bir doküman içerisindeki n-gram oluşumlarının tanımlanmasıdır. N-gram frekans yaklaşımı dilden bağımsız çalışır. Yani belirli bir dil hakkında detaylı bir dilbilgisine veya bir sözlük yapısına ihtiyaç yoktur. Tüm harflerin veya hecelerin istatistiklerini kullanarak benzer sonuçlara

ulaşmak mümkündür. Bununla beraber, bu düşünce ile birlikte çok kısa paragraflardan oluşan dokümanlar, konu tabanlı kelime istatistiğinde çok yetersiz kalmaktadır. Sonuç olarak eşleme için önemli olabilecek n-gram'ların yeterli şekilde toplanabilmesi için daha uzun paragraflara ihtiyaç vardır.



Şekil 2.2: Örnek 2-Gram Tablosunun Oluşturulması

N-gramlar kelimenin başından ve sonundan değişik uzunluklarda n-gramlar alınarak yapılmıştır. N-gram, yüksek verimli sınıflandırma sağlamıştır. Örneğe göre sınıflandırma yapılması önemlidir. Toplanılan örnekler OCR gibi, üzerinde otomatik yolla sınıflandırma yapılmıştır. Günümüzde birbirinden farklı diller üzerinde tanıma yapılması için n-gram yöntemi etkin olarak kullanılmaktadır.

2.5. Metin Sınıflandırma Yöntemleri

Metin tabanlı sınıflandırma başarısının ölçülmesi için kullanılan önemli sınıflandırma ve kümeleme algoritmaları vardır. Bunlar Markov Modeller, DVM, Naive Bayesian, Centroid Sınıflayıcı, Karar Ağaçları, YSA, K-Ortalamalar algoritmasıdır ve algoritmaların işleyişi ile ilgili bilgi verilmiştir.

2.5.1. Markov Modeller

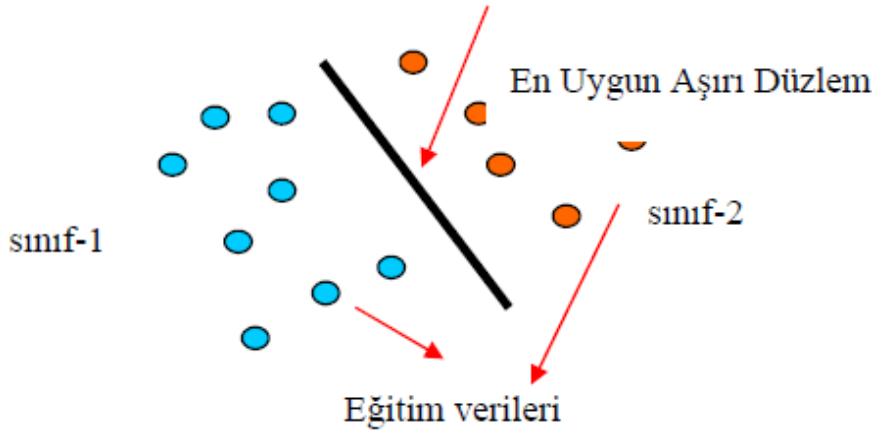
Belli bir zaman diliminde çeşitli durumlarda bulunmanın ve bir durumdan diğer duruma geçişin olasılıklarının göz önüne alınır. Bir durumdan diğer duruma geçişte, sistemin bir önceki durumuna bağlıdır ve bu durumda şartlı olarak ifade edilir. Buradaki amaç, incelenen sorunun beklenen sonucuna ilişkin optimum yapıyı belirlemektir(Satish, Gururaj, 2003).

2.5.2. Destek Vektör Makinaları (Support Vector Machine, DVM)

Destek vektör makinası, sınıflandırma ve regresyon analizi için kullanılan, veri analiz ve desenlerini tanımayı sağlayan denetimli öğrenme için kullanılan bir yöntemdir. Destek Vektör Makinesi, makine öğrenmesi yöntemlerinden biri olup V.Vapnik tarafından ortaya atılmıştır. Çekirdek tabanlı doğrusal olmayan sınıflandırıcıların sinyal işleme, yapay öğrenme ve veri madenciliği alanındaki pratik problemlerde iyi sonuçlar verdiği kanıtlanmıştır(Vapnik)(Dunning, 1994). Temelde DVM iki sınıflı problemlerle ilgilenir. Girişte alınan veriler destek vektörler ile tanımlanabilen bir aşırıdüzlem (hyperplane) tarafından Şekil 2.4.'de gösterildiği gibi ikiye ayrılır. Amaç 2 sınıfı birbirinden ayırabilecek en uygun aşırıdüzlemi bulabilmektir.

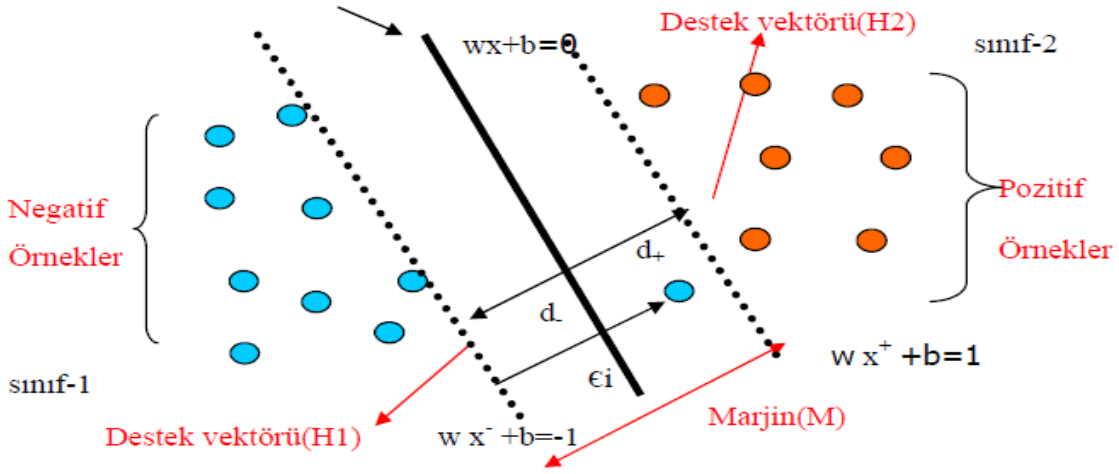
Standart DVM giriş veri kümesi alır ve olası sınıfların giriş olarak tahminde bulunur.

Eğitim örnekleri kümesi verildiğinde, her ikikategoriden birine aitolarak işaretlenmiş bir DVM eğitim algoritması bir kategori ya da diğer örnekler için yeni örnekler atar ve bir model oluşturur. DVM, ayrı kategoriden örneklerin aralığını mümkün olduğunca geniş bir boşluk ayırarak açar. Böylece eşlenmiş uzayda sınıfları temsil eden örnek bir temsil bulunur. Test verisinden gelen verilerin hangi örneğe denk geldiğine ait öngörü oluşturur.



Şekil 2.3: Gelen Verileri Ayıran Aşırı Düzlem

En uygun aşırıdüzlemi bulabilmek için, her iki sınıfın en uygun aşırıdüzlemine en yakın veri noktalarından geçen aşırıdüzlemler çizilir ve bu iki düzlem birbirine paraleldir. Bu düzlemler arasındaki mesafe en uygun aşırıdüzleminin kalitesini belirler. DVM iki sınıf arasındaki sınırı ayırt etme yüzeyini belirlemekte, yani eğitim kümesi ile ayırt etme yüzeyine en yakın noktaların arasındaki mesafeyi maksimumlaştırmaktadır. Destek vektörleri, aşırıdüzlemler ve marjin kavramları şekil Şekil 2.4.'de gösterilmiştir.



Şeki 2.4: İki Sınıfı Ayıran Aşırıdüzlem ve Sınır(Marjin)

İki sınıfın örneklerini birbirinden ayıran bir aşırıdüzlem vardır, bu düzlem üzerindeki noktalar

w.x+b=0 eşitliğini sağlayacaktır, burada w aşırıdüzleme olan normal ve |b|/||w|| aşırıdüzlemden orijine olan dik uzaklıktır. Destek vektör yöntemi aşırıdüzleme en yakın

pozitif ve negatif örnekler arasındaki mesafenin (marjin genişliğinin) en yüksek olduğu bir

aşırıdüzlem bulmaya çalışır. Marjin (M) genişliği aşağıdaki gibidir.

$$w.\ x+ + b = +1 \qquad (2.1.)$$
$$w.\ x- + b = -1$$

"w" değeri ne kadar küçülürse marjin genişliği o kadar artar. Her nokta xi olarak gösterilir. İki sınıfa ait eğitim verisi x = (x1,x2,…,xn) ve etiket değeri y = (-1, +1) ise en uygun aşırıdüzlem aşağıdaki gibidir:

$$\text{eğer } y = +1 \text{ ise } w\ .\ xi + b >= 1 \qquad (2.2.)$$
$$\text{eğer } y = -1 \text{ ise } w\ .\ xi + b < 1$$

Verilen X örneğini sınıflandırmak için öncelikle en uygun aşırıdüzlem bulunur. Bu

aşırıdüzlem taraflarından biri negatif sınıfı, diğeri ise pozitif sınıfı temsil eder.

$$f(x) = \text{sign}(w\ x + b)$$

f(x) >= 0 pozitif sınıfı temsil etmektedir.

f(x) < 0 negatif sınıfı temsil etmektedir.

Çoklu-sınıf verilerinin DVM ile sınıflandırılması için Kneer ve arkadaşları bire-karşı-bir (oneagainst-one) yöntemini önermişlerdir(Boser ve diğ., 1992). Bu yöntemde n adet sınıf için n(n−1)/2 sınıflandırıcı oluşturulur. Her biri sadece iki sınıftan oluşan veriler ile eğitilir. Bu yöntem sayesinde çoklu sınıf problemi iki sınıf problemine çevrilmiş olur. Ayrıca eğitim için her sınıflandırıcıda sadece iki sınıfa ait verilerin kullanılması toplam eğitim zamanını azaltacaktır. Çoklu-sınıf verilerinin DVM ile sınıflandırılması için bir diğer yöntem ise bire-karşı-hepsi(one-against-all) adı verilen yöntemdir (Suykens ve Vandewalle, 1999). Bu yöntemde n adet sınıf için n adet DVM kurulur ve i.nci DVM, i sınıfındaki verileri kendi sınıf verileri olarak kullanırken, diğer sınıflardan gelen verilerin hepsini sanki 2.sınıfa ait veriymiş gibi kabul eder. Yani kendi verilerine +1 etiketi verirken, diğer sınıflara ait olan tüm verilere -1 etiketini verir ve eğitimi bu şekilde n adet DVM için yapar. Dikkat edilirse eğitim zamanı, eğitim için fazla sayıda örüntünün kullanıldığı sınıflandırma problemleri için, bire-karşı-bir (Müller ve diğ. ,2001) yöntemine göre oldukça büyük olacaktır.

DVM yöntemi cinsiyet belirleme, yüz tanıma, karakter tanıma vb. çalışmalarda kullanılmıştır. Kısaca DVM, doğrusal olmayan bir şekilde ayrılabilen öbekler için en uygun aşırıdüzlem bulmaya çalışır. Bu yüzden DVM'nin VM'deki uygulamaları özellikle sınıflama tekniğinde ortaya çıkmıştır. Elde edilen sonuçlar bu yöntemin sınıflama tekniğinde oldukça başarılı olduğunu göstermiştir (Fung ve Mangasarian, 2002).

2.5.3. Naive Bayesian Sınıflayıcı Modeli

Naive Bayes yöntemi doküman sınıflandırma işlemlerinde en sık kullanılan, pratik, olasılığa dayanan bir sınıflayıcıdır. Diğer bütün sınıflandırıcılarla karşılaştırıldıklarında en düşük hata oranına sahiptirler. Elimizde n adet sınıf olduğunu farz edelim, $S_1,S_2,\ldots,S_n$. Herhangi bir sınıfa ait olmayan bir veri örneği X'in, hangi sınıfa ait olduğu Naive Bayes sınıflandırıcı tarafından belirlenir. Veri örneği X, verilen sınıflara ait olma olasılığı en yüksek değere sahip sınıfa atanır.

Sonuç olarak, Naive Bayes sınıflandırıcı bilinmeyen örnek X'i, S_i sınıfına atar. Her veri örneği, m boyutlu özellik vektörleri ile gösterilir, X = (X_1,X_2,…,X_m). Özelliklerin hepsi aynı derecede önemlidir ve birbirinden bağımsızdır. Bir özelliğin değeri başka bir özellik değeri hakkında bilgi içermez(Domingos ve diğ., 1997).

Verilen dokümanda kategori sınıflandırılması ve kelimenin joint olasılıksal sınıflandırılmasında kullanılır. Bu çalışmada kelimenin frekans özellikleri yoktur. Buradakiler sürekli yapıya sahiptir. WEKA implemantasyonu kullanılmıştır.

L değeri dilleri, D değeri test dökümanını belirtir. Payda değeri bütün diller için aynıdır. Buradaki amaç test dökümanının hangi dile ait olduğunu olasılıksal olarak en büyüğünü veren değeri bulmaktır.

Tablo 2.2.'deki eğitim verisinden yararlanarak bilinmeyen bir X örneğinin hangi sınıfa ait olduğunu Naive Bayes sınıflandırma kullanılarak tahmin etmek isteyelim. Eğitim verisi çizelge 2.2.'deki veri olup, örnekleri vites, renk, yakıt, ve kapı özellikleri ile tanımlanır. Elimizde A ve B olmak üzere 2 sınıf mevcuttur. Sınıflandırmak istediğimiz bilinmeyen örnek,

X = (vites = "otomatik", renk = "gri", yakıt = "benzinli", kapı = "4") olsun.

Tablo 2.2.Naive Bayesian İçin Örnek A ve B eğitim seti

Sınıf		Vites	Renk	Yakıt	Kapı
1	A	Otomatik	Gri	Dizel	2
2	B	Normal	Gri	Benzinli	4
3	A	Otomatik	Kırmızı	Benzinli	2
4	A	Otomatik	Gri	Benzinli	4
5	A	Otomatik	Beyaz	Dizel	2
6	B	Otomatik	Kırmızı	Benzinli	4
7	A	Normal	Gri	Dizel	4
8	B	Otomatik	Gri	Benzinli	4
9	A	Normal	Beyaz	Benzinli	4
10	A	Otomatik	Kırmızı	Benzinli	4
11	B	Normal	Kırmızı	Dizel	4
12	A	Normal	Beyaz	Dizel	4
13	A	Normal	Gri	Benzinli	2
14	B	Otomatik	Beyaz	Benzinli	4
15	B	Otomatik	Beyaz	Benzinli	4

X bilinmeyen verisinin hangi sınıfa ait olduğunu bulabilmek için $P(X|S_i)P(S_i)$ değerini

maksimize etmemiz gerekmektedir. A sınıfı 9 elemandan, B sınıfı da 6 elemandan oluşmuş

$P(S_i)$ için, her sınıf için olasılık değerleri eğitim örneklerinden hesaplanabilir:

P(A) = 9/15 =0.600

P(B) = 6/15 =0.400

P(X|S_i), i =1,2 değerlerinin koşullu olasılıkları hesaplarsak,

P(vites = “otomatik” | A) = 5/9 = 0.555

P(renk = “gri” | A) = 4/9 = 0.444

P(yakıt = “benzinli” | A) = 5/9 = 0.555

P(kapı = “4” | A) = 5/9 = 0.555

P(vites = “otomatik” | B) = 4/6 = 0.667

P(renk = “Gri” | B) = 4/6 = 0.667

P(yakıt = “benzinli” | B) = 5/6 = 0.833

P(kapı = “4” | B) = 6/6 = 1

elde ederiz. Hesaplanan bu olasılık değerleri kullanılarak

P(X|A) = 0.555*0.444*0.555*0.555 = 0.076

P(X|B) = 0. 667*0. 667*0. 833*1 = 0.370

P(A) P(X|A) = 0.600*0.076= 0.046

P(B) P(X|B) = 0.400*0.370= 0.14

2.5.4. Centroid Tabanlı Sınıflayıcı

Centroid tabanlı sınıflayıcı vektör uzayı modeli tabanlı, etkili bir metin sınıflandırma yöntemidir. Doküman sunumunda vektör uzayı modelinin sağladığı imkânları kullanır. Model her bir dokümanı d terim uzayında bir vektör olarak ele alır. Vektörün her bir boyutu dokümanda geçen bir terimin ağırlıklandırılmış frekansını tutar. Centroid tabanlı sınıflandırmada, sınıflar centroid adı verilen vektörlerle sunulur. Centroid, sınıf elemanlarını sunan ortalama bir değerdir ve bu orta değerin bütün sınıfı temsil ettiği kabul edilir. Eğitim seti k farklı kategori içeriyorsa bu eğitim verilerinden k adet centroid vektörü elde edilir.

$$\vec{C} = \frac{1}{|S|}\sum_{d \in S}\vec{d} \qquad (2.3.)$$

Doküman değerinin centroid değeri için eldeki dökümanın ortalaması alınır.

$$sim(\vec{x}, \overrightarrow{C_{j,j\in k}}) = \frac{\vec{x}.\overrightarrow{C_j}}{\left\|\vec{x}\right\|_2 * \left\|\overrightarrow{C_j}\right\|_2} \quad (2.4.)$$

Test metninde gelen dökümanın centroid değeri hesaplanır ve önceden hesaplanmış centroid değerine en yakın değerin diline atanır.

Centroid sınıflayıcı ile kategorisi bilinmeyen bir metni sınıflandırmada benzerlik yönteminden yararlanılır. En sık kullanılan benzerlik yöntemi cosine benzerlik yöntemidir. Bir test dokümanının hangi kategoriye ait olduğunu bulmak için öncelikle test dokümanı ile centroid vektörleri arasındaki benzerlikler hesap edilir ve test dokümanı kendine en yakın kategoriye atanır. En yakın kategorinin elde edilmesinde maksimum benzerlik formülünden faydalanılır. Aşağıdaki iki denklemde benzerliğin bulunması ve maksimum benzerliğin seçimi sunulmaktadır.

İki vektör arasındaki açının kosinüsü	cos(d1,d2)=d1*d2/||d1||||d2|	(2.5.)

di*dj: iki dokümanın vektör çarpımı

||di||: di dokümanının uzunluğu

Benzerlik değerleri elde edildikten sonra maksimum olanın seçimi bize dokümanın sınıfını vermektedir. Centroid tabanlı sınıflayıcıların performansı genellikle centroid vektörlerinin kalitesine bağlıdır. Performans artımı için centroid vektörlerinin güncellendiği birçok çalışma yapılmıştır. Birkaç tane centroid yönteminin birleştirildiği bir çalışmada sonuçların iyileştiği gözlenmiştir (Lertnattee ve Theeramunkong).

Bu çalışmada da centroid değerlerinin elde edilmesi ile ilgili yeni bir yöntem ortaya konularak sınıflandırma başarısı artırılmaya çalışılacaktır. Sınıflandırma farklı boyuttaki test metinleriyle karşılaştırmalı olarak performasnları değerlendirilmiştir.

2.5.5. Karar Ağaçları

Verileri sınıflandırma yöntemlerinden biri "karar ağaçları" ile sınıflandırma adını taşır. Karar ağaçları akış şemalarına benzeyen yapılardır. Her bir nitelik bir düğüm tarafından temsil edilir. Dallar ve yapraklar ağaç yapısının elemanlarıdır. En son yapı yaprak; en üst yapı kök ve bunlar arasında kalan yapılar ise dal olarak isimlendirilir. Karar ağaçları sınıflandırma algoritmalarını uygulayabilmek için uygun bir alt yapı sağlamaktadır. Karar ağaçlarında en önemli sorunlardan birisi herhangi bir kökten itibaren bölümlemenin ve ya bir başka deyişle dallanmanın hangi kıstasa göre yapılacağıdır. Aslında her farklı ölçüt için bir karar ağacı algoritması karşılık gelmektedir. Bu algoritmalar;

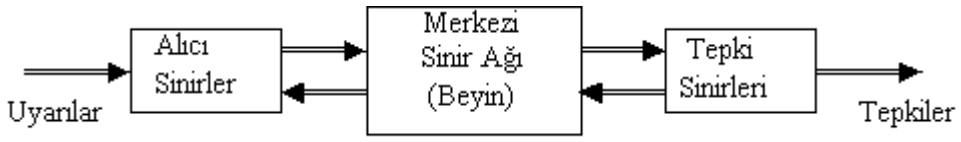
*entropiye dayalı algoritmalar
*sınıflandırma ve regresyon ağaçları
*bellek tabanlı sınıflandırma algoritmalar şeklinde gruplayabiliriz. Entropiye dayalı bölümlemeyi kullanan algoritmalara örnek olarak ID3 ve onun gelişmiş biçimi olan C4.5 algoritmaları verilebilir(Yuan ve Shaw, 1995).

Bu yöntemlerde karar ağacında hangi niteliğe göre dallanmanın yapılacağını belirlemek üzere entropiye başvurulur. Hedef niteliğini ifade eden T,hedef niteliği olmayan bir X niteliğinin değerine bağlı olarak T1,T2,….Tn alt kümelerine ayrılırsa T nin bir elemanının sınıfını belirlemek için gerekli bilgi, Ti nin bir elemanının sınıfının belirlenmesinde gerekli olan bilginin ağırlıklı ortalaması olarak kabul edilir.

2.5.6. Yapay Sinir Ağları

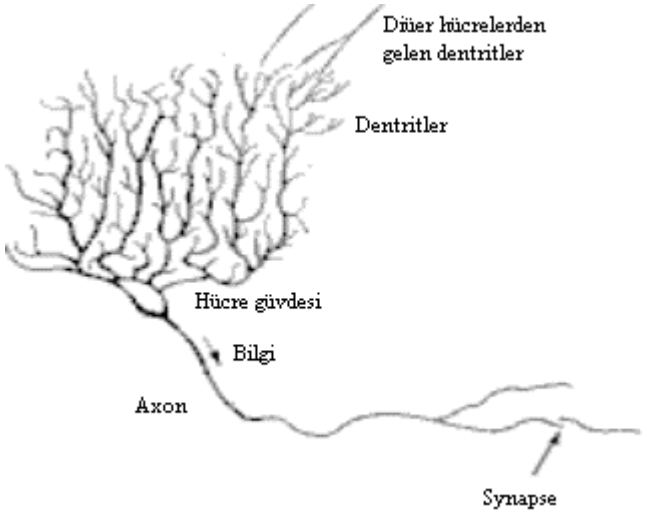
Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin (merkezi sinir ağı) bulunduğu 3 katmanlı bir sistem olarak açıklanır. Alıcı sinirler (receptor) organizma içerisinden ya da dış ortamlardan algıladıkları uyarıları, beyine bilgi ileten elektriksel sinyallere dönüştürür. Tepki sinirleri (effector) ise, beyinin ürettiği elektriksel darbeleri organizma çıktısı olarak

uygun tepkilere dönüştürür. Şekil 2.6.' de bir sinir sisteminin blok gösterimi verilmiştir.



Şekil 2.5. Biyolojik Sinir Sisteminin Blok Gösterimi

Merkezi sinir ağında bilgiler, alıcı ve tepki sinirleri arasında ileri ve geri besleme yönünde değerlendirilerek uygun tepkiler üretilir. Bu yönüyle biyolojik sinir sistemi, kapalı çevrim denetim sisteminin karakteristiklerini taşır. Merkezi sinir sisteminin temel işlem elemanı, sinir hücresidir (nöron) ve insan beyninde yaklaşık 10 milyar sinir hücresi olduğu tahmin edilmektedir. Sinir hücresi; hücre gövdesi, dendriteler ve axonlar olmak üzere 3 bileşenden meydana gelir. Dendriteler, diğer hücrelerden aldığı bilgileri hücre gövdesine bir ağaç yapısı şeklinde ince yollarla iletir. Aksonlar ise elektriksel darbeler şeklindeki bilgiyi hücreden dışarı taşıyan daha uzun bir yoldur. Aksonların bitimi, ince yollara ayrılabilir ve bu yollar, diğer hücreler için dendriteleri oluşturur. Şekil 2.7.' de görüldüğü gibi axon-dendrite bağlantı elemanı synapse olarak söylenir(Deng ve diğ., 2011).



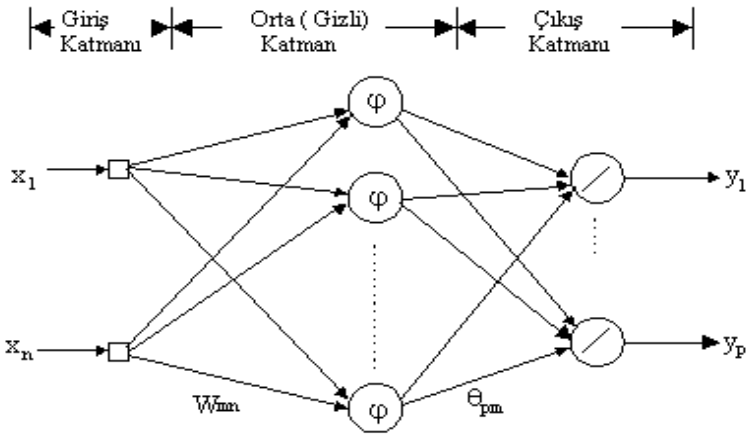
Şekil 2.6. Biyolojik Sinir Hücresi ve Bileşenleri

Synapse gelen ve dendriteler tarafından alınan bilgiler genellikle elektriksel darbelerdir ancak, synapsedeki kimyasal ileticilerden etkilenir. Belirli bir sürede bir hücreye gelen girişlerin değeri, belirli bir eşik değerine ulaştığında hücre bir tepki üretir. Hücrenin tepkisini artırıcı yöndeki girişler uyarıcı, azaltıcı yöndeki girişler ise önleyici girişler olarak söylenir ve bu etkiyi synapse belirler.

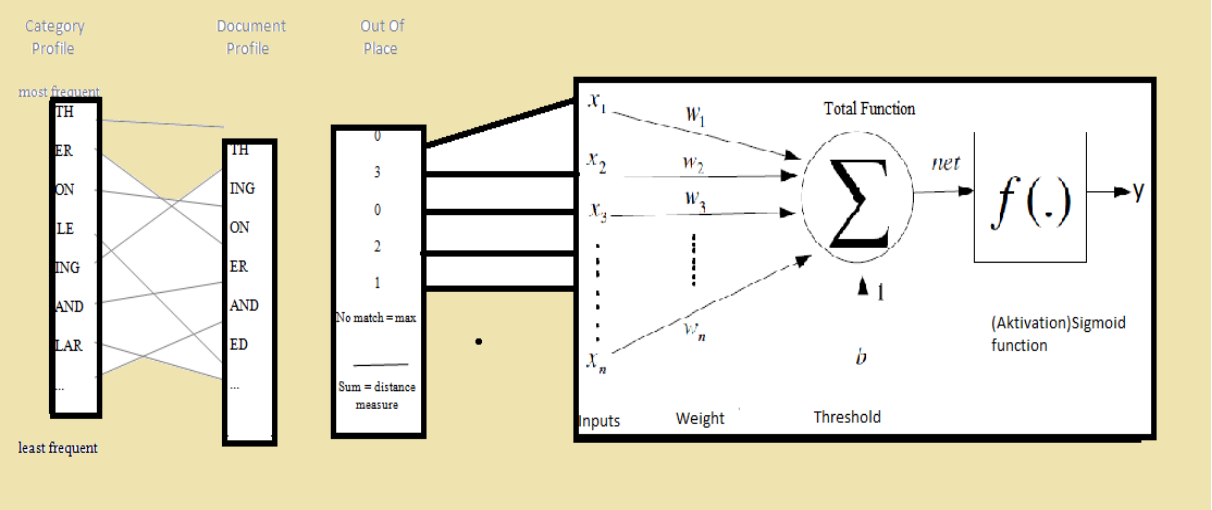
İnsan beyninin 10 milyar sinir hücresinden ve 60 trilyon synapse bağlantısından oluştuğu düşünülürse son derece karmaşık ve etkin bir yapı olduğu anlaşılır. Diğer taraftan bir sinir hücresinin tepki hızı, günümüz bilgisayarlarına göre oldukça yavaş olmakla birlikte duyusal bilgileri son derecede hızlı değerlendirebilmektedir. Bu nedenle insan beyni; öğrenme, birleştirme, uyarlama ve genelleştirme yeteneği nedeniyle son derece karmaşık, doğrusal olmayan ve paralel dağılmış bir bilgi işleme sistemi olarak tanımlanabilir. Beynin üstün özellikleri, bilim adamlarını üzerinde çalışmaya zorlamış ve beynin nörofiziksel yapısından esinlenerek matematiksel modeli çıkarılmaya çalışılmıştır. Beynin bütün davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Böylece Yapay Sinir Ağları denen yeni ve günümüz bilgisayarlarının algoritmik hesaplama yönteminden farklı bir bilim alanı ortaya çıkmıştır. Yapay sinir ağları; yapısı, bilgi işleme yöntemindeki farklılık ve uygulama alanları nedeniyle çeşitli bilim dallarının da kapsam alanına girmektedir.

Genel anlamda YSA, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Donanım olarak elektronik devrelerle ya da bilgisayarlarda yazılım olarak gerçeklenebilir. Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir. Öğrenme süreci, arzu edilen amaca ulaşmak için YSA ağırlıklarının yenilenmesini sağlayan öğrenme algoritmalarını ihtiva eder. YSA' nın hesaplama ve bilgi işleme gücünü, paralel dağılmış yapısından, öğrenebilme ve genelleme yeteneğinden aldığı söylenebilir. Genelleme, eğitim ya da öğrenme sürecinde karşılaşılmayan girişler için de YSA' nın uygun tepkileri üretmesi olarak tanımlanır. Bu üstün özellikleri, YSA' nın karmaşık problemleri çözebilme yeteneğini gösterir.

Çalışmamızda çeşitli boyutlardan elde edilmiş eğitim verilerinin yapay sinir ağlarına giriş olarak verilmesiyle başarı oranı elde edilmiştir. Bu uygulama geliştirilirke YSA'nın ileri beslemeli geriye yayılımlı yapay sinir ağları algoritması kullanılmıştır. İleri beslemeli YSA' da, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan orta (gizli) katmandaki hücrelere iletir. Bilgi, orta ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı YSA' nın, orta katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir. En çok bilinen geriye yayılım öğrenme algoritması, bu tip YSA ların eğitiminde etkin olarak kullanılmakta ve bazen bu ağlara geriye yayılım ağları da denmektedir. Şekil 2.8. de giriş, orta ve çıkış katmanı olmak üzere 3 katmanlı ileri beslemeli YSA yapısı verilmiştir.



Şekil 2.7. İleri beslemeli 3 katmanlı YSA

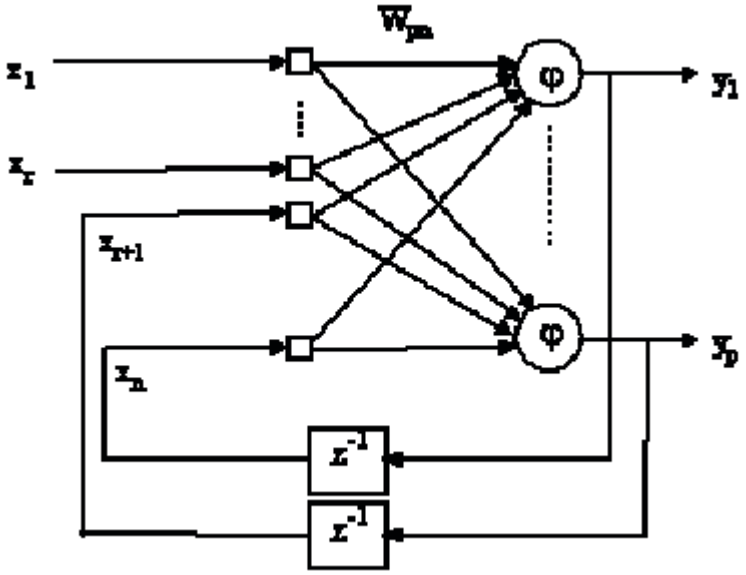


Şekil 2.8. N-Gram Değerleriyle YSA Blok Diyagramı

İleri beslemeli 3 katmanlı ve çıkış katmanı doğrusal olan YSA' nın matematiksel modeli, x- giriş vektörünü, o- orta katman çıkış vektörün ü, y- ağ çıkış vektörünü göstermek üzere x_0 ve o_0 girişleri, polarma girişleri olarak alınmıştır.

Herhangi bir problemi çözmek amacıyla kullanılan YSA da, katman sayısı ve orta katmandaki hücre sayısı gibi kesin belirlenememiş bilgilere rağmen nesne tanıma ve sinyal işleme gibi alanların yanı sıra ileri beslemeli YSA, sistemlerin tanılanması ve denetiminde de yaygın olarak kullanılmaktadır.

Geri beslemeli YSA' da ise, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli YSA , doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli YSA yapıları elde edilebilir. Bu nedenle, bu bölümde bazı geri beslemeli YSA yapılarında örnekler verilecektir. Şekil 2.10. de iki katmanlı ve çıkışlarından giriş katmanına geri beslemeli bir YSA yapısı görülmektedir(MacNamara ve diğ., 1998).



Şekil 2.9. Geri Beslemeli İki Katmanlı YSA

2.5.7. K-OrtalamalarAlgoritması

K-ortalamalar algoritması yöntemi, kümeleme problemini çözen en basit denetimsiz öğrenme yöntemleri arasında yer alır. Algoritmanın genel mantığı n adet veri nesnesinden oluşan bir veri kümesini(X), giriş parametresi olarak verilen k (k * n) adet kümeye bölümlemektir. Amaç, gerçekleştirilen bölümleme işlemi sonunda elde edilen kümelerin, küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerinin minimum olmasını sağlamaktır. Yöntemin performansını k küme sayısı, başlangıç olarak seçilen küme merkezlerinin değerleri ve benzerlik ölçümü kriterleri etkilemektedir.

Yöntemin adımları aşağıda gösterilmiştir.

Adım 1: Test kümesi k alt kümeye bölünür.

Adım 2: Bölme işleminde k küme için ilk küme merkezleri belirlenir. C = {c1, c2, c3…ck} Bunun için nesneler arasından k adet rastgele nokta seçilebilir ya da tüm nesnelerin ortalaması ile de belirlenebilir.

Adım 3: Test kümesindeki her verinin X = {x1, x2,x3…xn} seçilen merkez noktalara yakınlığı kosinüs benzerliği ile hesaplanır. Her veri kendine en yakın merkez noktanın olduğu kümeye dahil edilir.

$$cos\left(x_i, merkez\left(c_j\right)\right) = \frac{x_i}{||x_i||} * \frac{merkez(c_j)}{||merkez(c_j)||} \qquad (2.6.)$$

(i = {1, 2, 3...n}, j = {1, 2, 3...k})

Adım 4: Oluşan kümelerin merkez noktaları o kümedeki tüm nesnelerin ortalama değerleri ile değiştirilir.

$$merkez(c\_j) = \frac{\sum_{i=1}^{n_j}(X_i)}{n_j} \quad (2.7.)$$

($x_i \in c_j$) ve $n_j = c_j$ kümesindeki veri sayısı

Adım 5: Merkez noktalar değişmeyene kadar 3. ve 4. adımları tekrarlanır.

K-ortalamalar algoritmasının trigram veri seti üzerinde dil tanıma başarı oranı Tanagra veri madenciliği yazılımında test edilmiştir(MacQueen, 1967).

2.5.8. Bulanık C Ortalamalar Algoritması

Bulanık mantık (Fuzzy Logic) kavramı ilk kez 1965 yılında California Berkeley Üniversitesinden Prof. Lotfi A.Zadeh'in(Seising ve Rudolf, 2007) bu konu üzerinde ilk makalelerini yayınlamasıyla duyulmuştur. O tarihten sonra önemi gittikçe artarak günümüze kadar gelen bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematik düzen olarak tanımlanabilir. Bilindiği gibi istatistikte ve olasılık kuramında, belirsizliklerle değil kesinliklerle çalışılır ama insanın yaşadığı ortam daha çok belirsizliklerle doludur. Bu yüzden insanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışmak gereklidir(Bezdek, 1981).

Bulanık mantık ile matematik arasındaki temel fark bilinen anlamda matematiğin sadece aşırı uç değerlerine izin vermesidir. Klasik matematiksel yöntemlerle karmaşık sistemleri modellemek ve kontrol etmek işte bu yüzden zordur, çünkü veriler tam olmalıdır. Bulanık mantık kişiyi bu zorunluluktan kurtarır ve daha niteliksel bir tanımlama olanağı sağlar. Bir kişi için 38,5 yaşında demektense sadece orta yaşlı demek birçok uygulama için yeterli bir veridir. Böylece azımsanamayacak ölçüde bir bilgi indirgenmesi söz konusu olacak ve matematiksel bir tanımlama yerine daha kolay anlaşılabilen niteliksel bir tanımlama yapılabilecektir.

Bulanık mantıkta bulanık kümeler kadar önemli bir diğer kavramda dilsel değişken(Linguisic Variable) kavramıdır. Dilsel değişken "sıcak" veya "soğuk" gibi

kelimeler ve ifadelerle tanımlanabilen değişkenlerdir. Bir dilsel değişkenin değerleri bulanık kümeler ile ifade edilir. Örneğin oda sıcaklığı dilsel değişken için "sıcak", "soğuk" ve "çok sıcak" ifadelerini alabilir. Bu üç ifadenin her biri ayrı ayrı bulanık kümeleri ile modellenir.

Bulanık mantığın uygulama alanları çok geniştir. Sağladığı en büyük fayda ise "insana özgü tecrübe ile öğrenme" olayının kolayca modellenebilmesi ve belirsiz kavramların bile matematiksel olarak ifade edilebilmesine olanak tanımasıdır. Bu nedenle lineer olmayan sistemlere yaklaşım yapabilmek için özellikle uygundur.

Bulanık kümeleme analizi, desen tanıma, görüntü işleme ve bulanık modelleme gibi uygulamalar için etkili bir şekilde kullanılmaktadır. En iyi bilinen bulanık kümeleme yaklaşımı ilk olarak Dunn (Dunn, 1974) tarafından önerilen ve Bezdek tarafından geliştirilen Bulanık C-Ortalamalar(BCO) algoritmasıdır.

Bulanık C- Ortalamalar algoritması, amaç fonksiyonuna dayanan bütün kümeleme tekniklerinin temelini oluşturmaktadır. İlk olarak Duda ve Hart (Duda ve Hart, 1973) sert küme bölünmesini(Hard Cluster Partition) hesaplamıştır. Dunn ise (Dunn, 1974) bir bireyin birden fazla kümeye girebilmesini sağlamak için bu algoritmanın bulanık versiyonunu tanıtmıştır. Son olarak, Bezdek bulanıklık indeksini(m) dahil ederek algoritmanın son halini geliştirmiştir. BCO algoritması sonuçlandığında, p- boyutlu uzaydaki noktalar küresel bir şekil alır. Bu kümelerin yaklaşık olarak aynı boyutta olduğu varsayılır. Her bir kümeyi, küme merkezleri(centre) temsil eder ve bunlara prototip denir. Uzaklık ölçüsü olarak, veriler ile küme merkezi arasındaki Euclidian uzaklığını kullanır.

$$d^2(v_i, x_k) = D_{FCM} = \|x_k - v_i\|^2 = \sum_{v=1}^{p} (x_k{}^{(v)} - v_i{}^{(v)})^2$$

(2.8.)

Burada $p \in N, D = R^p$, **x={** $x_1, x_2, ..., x_n\} \subseteq D, C = R^p, c \in N$, R= $P_c(C)$, m$\in R_{>1}$ ve

d:D x C $\rightarrow R, (x, p) \rightarrow \|x - p\|$ V={ $v_1, v_2, ..., v_c\} \in R$

Bu tekniğin uygulanabilmesi için, küme sayısının ve bireylerin kümeye üyelik derecelerinin önceden bilinmesi gerekmektedir. Bu tür parametrelerin önceden bilinmesi zor olduğundan, bu değerler deneme yanılma yoluyla ya da geliştirilen bazı

teknikler ile bulunabilir. Bu kümeleme yöntemi için kullanılan amaç fonksiyonu şu şekildedir:

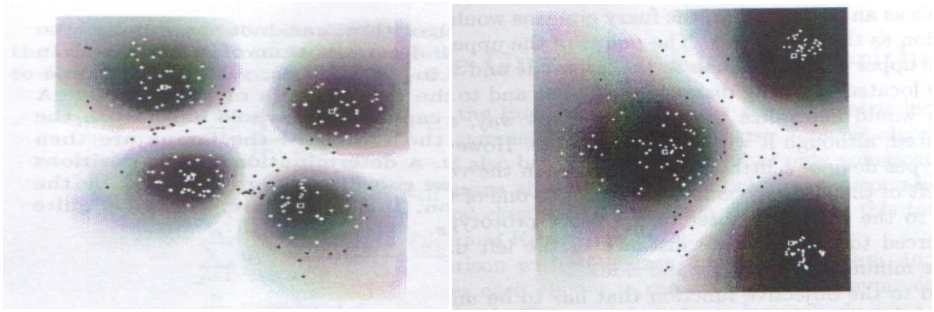
$$J(u,v) = \sum_{j=1}^{n}\sum_{t=1}^{c} {u_{t,j}}^{m} \left\| x_j - v_t \right\|^2 \qquad (2.9.)$$

Bu fonksiyon en küçük kareler fonksiyonudur. n parametresi gözlem sayısını, c ise küme sayısını gösterir. ${u_{tj}}^{m}$ t. kümedeki x_j 'nin üyeliği, J(u,v) değeri ise tüm ağırlıklandırılmış kare hatalarının toplamının bir ölçüsüdür.

Eğer J(u,v) fonksiyonu c'nin her değeri için minimize edilecek olursa, diğer bir deyişle v_i 'lere göre 1.dereceden türevi alınıp 0'a eşitlenirse BCO Algoritmasının Prototipi şu şekilde olacaktır;

$$v_i = \frac{\sum_{j=}^{n} {u_{ij}}^{m} x_j}{\sum_{j=1}^{n} {u_{ij}}^{m}} \qquad (2.10.)$$

Burada bulanıklaştırıcı m BCO algoritması için önemli bir parametredir. Kaç tane kümenin üst üste geleceğini kontrol eder. m=1(bulanıklık indeksi) seçildiği zaman BCO algoritması, Sert C-Ortalamalar(SCO) algoritmasının genelleştirilmiş bir biçimi olacaktır. Burada küme prototipleri yine aynı formül ile hesaplanır ve üyelikler ya 0 olacaktır ya 1 olacaktır. Veriler en küçük uzaklığa sahip olduğu kümeye girer. Ancak BCO algoritması 0'a bölüm hatasına yol açabileceğinden m değeri olarak 1 kullanılmasına izin vermez.(Höppner ve diğ.,). Eğer m değeri çok büyük seçilirse bireylerin kümelere etkileri çok küçük olacaktır ve bireylerin kümelere üyelik dereceleri yaklaşık olarak 1/c olacaktır.BCO algoritmasında, küme prototipleri ile bireyler arasındaki uzaklık Şekil (2.10)'da verilen, sadece küresel şekle sahip kümeler için uygun olan Euclidian uzaklık ölçüsü ile hesaplanır. Ancak oval, hat, dörtgen gibi farklı şekillere sahip küme çeşitleri bulunmaktadır ve hatta veri seti içindeki kümeler farklı boyuta ve yoğunluğa sahip olabilirler. BCO algoritması, küme boyutları ve yoğunluklarının farklı olduğu durumlarda da iyi çalışamamaktadır. Ayrıca küçük boyutlara sahip kümeleri de teşhis etmek zor olmaktadır. Bu tip problemleri çözmek için, Gustafson-Kessel, Bulanık C-Hatlar gibi bir takım algoritmalar geliştirilmiştir(Zhang, 2005).



Şekil 2.10. BCO Algoritması Sonucunda Elde Edilen Kümeler

BCO algoritmasının uygulanması sonucunda oluşan kümeler Şekil 2.11'den görüldüğü gibi küresel yapıdadır.

BCO Algoritması için Gerekli Adımlar**:**

Adım1: Başlangıç Değerlerinin Girilmesi: Küme sayısı c, bulanıklık indeksi, m ve Üyelik dereceleri matrisi U veya V küme prototiplerini rasgele üret, işlem bitirme kriteri ε gibi,

Adım 2: V küme prototiplerinin rasgele üretildiği varsayılırsa bu değerler kullanılarak üyelik dereceleri matrisini hesapla

$$u_{ik} = \sum_{l=1}^{c} \left(\frac{\left\| v^i, x_k \right\|}{\left\| v^l, x_k \right\|} \right)^{-(2)/(m-1)} \qquad (2.11.)$$

Adım 3: 2.11. eşitliğine göre V küme prototiplerini güncelle

Adım 4: $\left\| V^{(t)} - V^{(t-1)} \right\| <$ ε ise iterasyon durdurulur aksi takdirde Adım 2'ye geri dönülür.BCO algoritması uygulandıktan sonra hangi bireyin hangi kümeye gireceğine karar vermek için üyelik dereceleri kullanılır. Her bir bireyin hangi kümeye olan üyeliğinin en büyük olduğuna bakılır ve bu bireyler o kümeye dâhil edilir. Ancak her bir birey diğer kümelere de belli bir üyelikle girerler.BCO algoritmasının sonucu başlangıçta rastgele üretilen değerlere oldukça bağlıdır. Bu yüzden bu rastgelelikten kaynaklanan problemleri ortadan kaldırmak için bir takım algoritmalar geliştirilmiştir.Uygulamamızda farklı boyutlardan elde edilmiş eğitim verileriyle test metinlerinin Fuzzy C Means algoritmasına göre kümeleme başarı sınaması yapılmıştır.

3. DENEYLER VE SONUÇLAR

3.1.Ön İşlemler ve Öznitelik Vektörlerinin Çıkarılması

Trigram tabanlı dil tanıma yöntemini test etmek için ECI (ECI/MCI, 2005) isimli, çok dilli bir külliyat kullanılmıştır. Bu külliyat, dilbilimsel çalışmalarda sıklıkla tercih edildiğinden ve Avrupa dilleri ile Türkçe için örnek verilere sahip olduğundan tercih edilmiştir. Külliyatın kaynağını ağırlıklı olarak Almanca (34 milyon kelime), Fransızca (4.1 milyon kelime) ve Hollanda dilinde (5.5 milyon kelime) yayın yapan basın yayın organları oluşturmaktadır. Ayrıca; İngilizce, Fransızca ve İspanyolca dillerinde paralel metinlerin yer aldığı bir organizasyondan da 5 milyon civarında kelime içeren bir külliyat ECI içinde bulunmaktadır. Diğer kaynaklar ise bütün Avrupa dilleri, Türkçe, Rusça ve Japonca'dır.

Testlerde kullanılmak üzere ECI corpustan yukarıdaki on beş dili içeren 4,5 Milyon karakter seçilmiştir. Seçilen verinin %70'i eğitim, %30'u ise test verisi olarak ayrılmıştır. Bu veriler kullanılarak farklı eğitim ve test kombinasyonları yapılmıştır. Özellik setimiz karakterlerden meydana geldiği için her bir eğitim ve test dokümanı karakter frekanslarına çevrilmiştir. Bu çevrim sırasında her bir metin 60 civarında karakterin (harfler ve bazı özel karakterler) sıklığı ile sunulmuştur. Avrupa Sözlük Girişimi (European Corpus Initiative (ECI)) mümkün olduğunca düşük bir maliyetle bilimsel araştırma için dijital olarak kullanılabilen büyük dilli sözlüktür. Korpus(sözlük) 1994 yılından itibaren CD-ROM ortamında temin edilmiştir ve ELSNET tarafından dağıtılır. Temmuz 1992 Frankfurter Rundschau Alman gazete metinleri - Mart 1993. Yaklaşık 34 milyon kelime. Eylül 1989, Ekim 1989 ve Ocak 1990 tarihinden itibaren malzemeden oluşan Le Monde'dan, Fransız gazetesi metinlerinden yaklaşık 4.1 milyon kelime vardır.

Tezde 15 dil üzerinde çalışma öngörülmüş olup bu diller; Türkçe, İngilizce, Almanca, Hollandaca, Fransızca, İtalyanca, Cezayirce, İspanyolca, Portekizce, Norveççe, Maltaca, Latince, Litvanyaca, İsveççe, Andoa Dilidir. ECI verilerini kullanmadan önce alt külliyatlar oluşturulmuştur. Alt külliyatlar oluşturulurken

külliyatların boyutu önemli bir konudur. 15 alt külliyatının hangi boyutta olması gerektiğini tespit için 100 KB ve 1 KB lık veriler seçilerek trigramlarına ayrılarak eğitim verisi oluşturulmuştur. Oluşturulan eğitim verileriyle sınıflandırma yapılarak sonuçların performans analizi yapılmıştır.

Tablo 3.1. Eğitim İçin Külliyat Boyutları

Dil	Külliyat Boyutu
İngilizce	1 KB ve 100 KB
Fransızca	1 KB ve 100 KB
Almanca	1 KB ve 100 KB
Hollanda	
Dili	1 KB ve 100 KB
İtalyanca	1 KB ve 100 KB
Portekizce	1 KB ve 100 KB
Türkçe	1 KB ve 100 KB
İspanyolca	1 KB ve 100 KB
İsveççe	
	1 KB ve 100 KB
Norveççe	
Latince	
Litvanya	1 KB ve 100 KB
Dili	1 KB ve 100 KB
Cezayir	1 KB ve 100 KB
Dili	1 KB ve 100 KB
Andoa	1 KB ve 100 KB
Dili	1 KB ve 100 KB
Malta Dili	

Deneysel çalışmamaızda kullanılan test metinleri ECI külliyatından seçilmiştir. Deneysel sonuçlar, 1 KB ve 100 KB veriler halinde incelenmiştir.

Klasik trigram yönteminde 26 harfli alfabeden türeyen trigram özellik seti kullanılır. Halbu ki çalışmamızda 15 dilde kullanılan toplam 60 harften oluşan karma bir alfabenin harfleri kullanılmıştır. Harf özellik setini oluşturan elemanlar İngiliz alfabesindeki 26 harf ve dil tanıma sistemi tarafından tanınan dillerin alfabesinde yer alan diğer özel harflerdir. Bir diğer ifadeyle özellik setimiz 15 adet dilin alfabesinin bir birleşimidir. Bu birleşimde yer almayan harfler özellik çıkarma esnasında özellik setinden çıkarılan, ayırt edici değeri düşük özelliklerdir. Bu harf özellik setinden oluşabilecek maksimum trigram sayısı (60*60*60 = 216000) değer söz konusudur.

Karma Özellik Seti

a,à,á,â,ã,ä,å,æ,b,c,ç,d,e,è,é,ê,ë,f,g,ğ,h,ı,i,ì,í,î,ï,j,k,l,m,n,ñ,o,ò,ó,ô,õ,ö,p,q,r,s,ş,t,u,ù,ú,û, ü,v,w,x,y,ÿ,ß,ø,z

3.2. Deneysel Tasarım

Deneysel çalışmalar sırasında, farklı eğitim seti boyutları, farklı algoritmalar ve farklı test seti boyutları için deneyler yapılacaktır. Deneylerden beklenti en uygun kesişimlerin bulunmasıdır.

3.2.1. Metinsel Verinin Sayısal Hale Dönüşümü

ECI'den elde edilen veriler ham halde bulunmaktadır. Ham metinsel verilere sınıflandırma tipinde sayısal algoritmalar uygulamak mümkün olmadığından metinsel dokümanların özetleme gibi işlemler yardımıyla sayısallaştırılmaları lazımdır.

Karakter tabanlı ön işlemin en temel iki aşaması temizleme ve dönüşümdür. Veri temizleme aşamasında veriler filtre edilir. Filtreleme, doküman bazında ve karakter bazında olabilir. Doküman bazında filtreleme yapılırken çok fazla bilgi taşımayan, içinde sık sık tekrarlar bulunan dokümanlar elenir. Çünkü tekrarların fazla olması dokümandaki kelime ve karakter sıklıklarını olumsuz yönde etkilemektedir.

Dokümanlar temizleme işleminden geçirildikten sonra ön işlemin ikinci aşaması olan dönüşüme hazır hale gelirler. Dönüşüm işlemi, ham bilgilerin karakter sıklık bilgilerine dönüştürülmesi işlevini yerine getirmektedir.

3.2.2. Dokümandan Trigram Veri Elde Edilmesi

Her dil için 100 Kb eğitim verisi kullanılmış ve bu veriler 1 Kb uzunluklu parçalara ayrılmıştır. Böylece bütün külliyat için 1500 adet kayıt elde edilmiştir. Kayıtlar vektör formatında tutulmakta olup aynı kayıtlar hem profil tabanlı

işlemlerde hem de örnek tabanlı işlemlerde kullanılacaktır. Ayrıca, külliyattan 100 Byte boyutunda 1500 kayıt elde edilmiş ve test işleminde kullanılmıştır. Gerek eğitim amaçlı veriler gerekse test amaçlı veriler özellik setine dayalı olarak özetlenmiştir. İşlem adımları şöyledir:

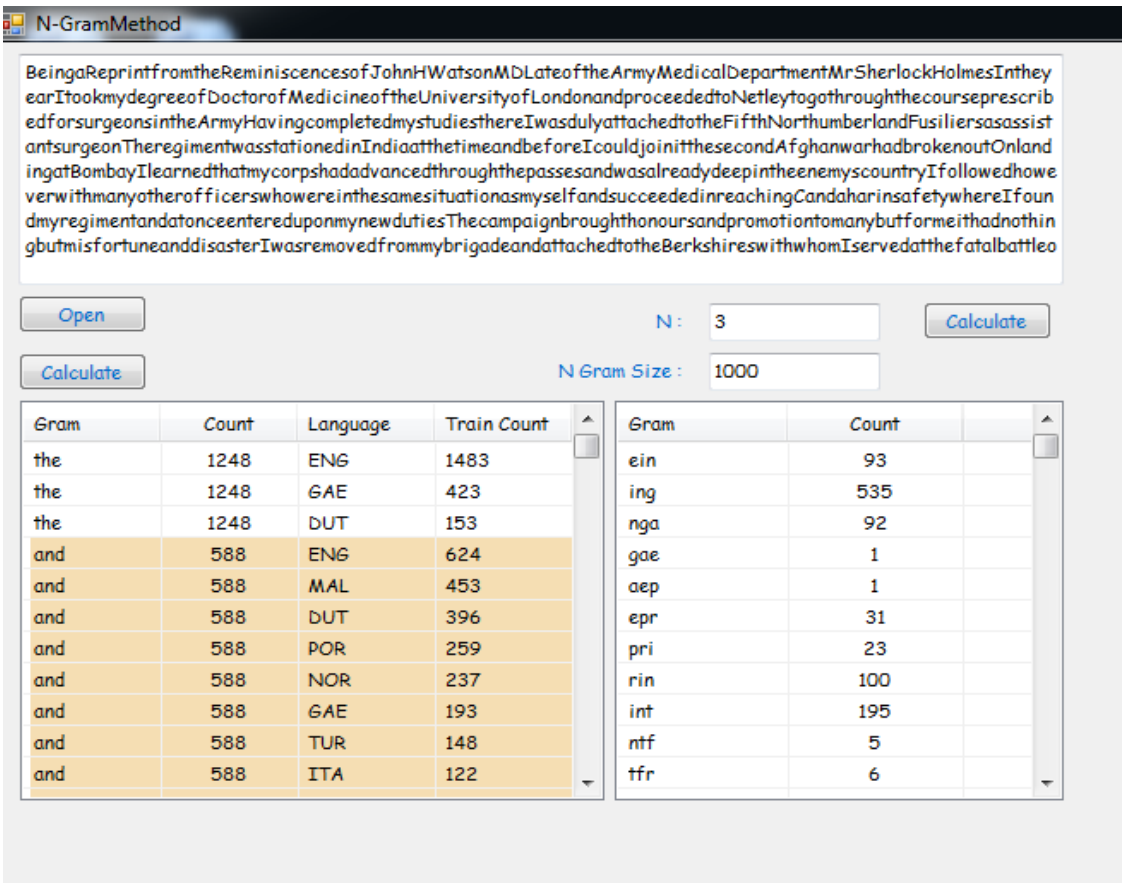
- Eğitim ve test verileri için 3-gram özetleri çıkarılır. Bu değerler işlem kolaylığı açısından büyükten küçüğe doğru Quick Sort sıralama algoritması ile sıralanmıştır.
- Profil tabanlı yöntemler için, her dile ait dil profil değerleri elde edilecektir. Bu işlem sonrasında 15 adet dil için 15 adet dil profili bulunacaktır. Yine en belirleyici n-gramlar en yukarıda bulunacak ve en belirleyici 300 adet n-gram seçilecektir.
- Test dokümanlarına ait n-gram profilleri ile dillere ait profiller karşılaştırılarak her bir test dokümanının dili, maksimum benzerlik veya minimum uzaklık formülü ile bulunacaktır.

Deneyler sırasında her bir dilde en önemli 3-gramların neler olduğu da tespit edilmiştir. Bu bilgi de dil tanıma çalışmalarında zaman zaman kullanılabilecek bir bilgidir.

Tablo 3.2. 100 KB Dillere Ait Verilerden En Önemli Trigram Seti

the	1483	ENG
kan	1168	MAL
zio	212	ITA
ikk	417	NOR
die	212	DUT
lar	760	TUR
ada	196	SPA
fro	343	GAE
som	71	SWE
jeg	535	NOR
ndo	302	POR
ost	320	CZE
que	314	LAT
ent	663	FRE
usi	329	LIT

100 KB ile sınırlandırılmış İngilizce test metninin programımızda trigramlarına ayrılması örneği Tablo 3.2.'de verilmiştir. Programımız nesneye yönelik proragramlama dillerinden C# dili ile implemente edilmiştir.



Şekil 3.1. 100 KB lik İngilizce Test Metninin Trigramlarına Ayrılması

Şekil 3.1.'de görüldüğü gibi eğitim dosyasından elde edilen trigram frekans değerleriyle (TrainCount), test metninde geçen trigram frekans değerleri (Count) birarada gösterilmiştir ve frekans sıklıklarına göre büyükten küçüğe sıralanmıştır. Çıkartılan trigramlardan görülmüştür ki, gerçekte her dile özgü belirleyici trigramlar vardır. Örneğin ‘lar’ Türkçe için, ‘the’ İngilizce için belirleyicidir. Deneylerde aynı trigramların farklı diller için de belirleyici olabildiği görülmüştür.

Her bir dokümanın özel bir şekilde ifade edilmesi amacıyla çıkarılmış özeliklerden bazıları dokümanları ayırt edici nitelik taşımamaktadır. Ayırt edici nitelikte olmayan özelliklere sahip özellik vektörleri kullanıldığı durumda sınıflandırma başarısı düşebilir.

Çalışmamızda bütün dillere ait trigramlar bir araya getirilmiş ve bunlar arasında en önemli olan 300 tanesi ortak trigram özellik seti olarak seçilmiştir. 15 adet dilden elde edilen toplam trigram sayısı 3000 adet olup onda bir(1/10) oranında azaltma yapılmıştır. Bu çalışmada Tanagra Tool içerisinde bulunan özellik setinde Fisher Filtering yöntemi özellik azaltma amacıyla kullanılmıştır. Fisher filtering, öğrenme algoritmaları gibi sınıflandırma algoritmalarında kullanılabilen özellik seçimi algoritmasıdır.

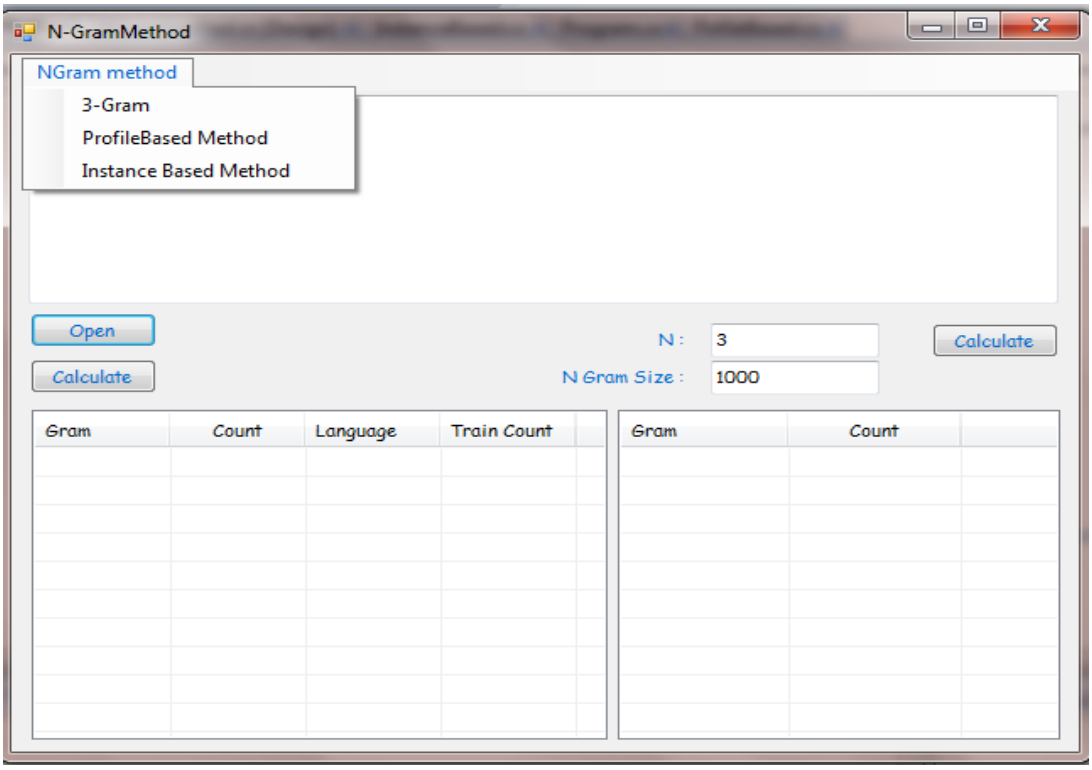
3.3.Trigram Eğitim Verileriyle Profil Tabanlı Dil Tanıma Metodu

Dil tanıma, başta doğal dil işleme olmak üzere birçok uygulamada anahtar vazife görmektedir. Dil tanıma, dilin ayırt edici özelliklerine bazı tekniklerin uygulanması ile gerçekleştirilmektedir. Bu kapsamda; terimler (kelime veya kelime öbekleri), harf dizileri veya n-gramlar bugüne kadar dilin ayırt edici özellikleri olarak dil tanımada kullanılmıştır. Bag-of-words çözümleme bir metin içerisinde yer alan bütün terimlerin sıklık bilgilerinin kullanılmasını ifade etmekte iken n-gramlar sıklık bilgisine ek olarak terimlerin sırası ile de ilgilenmektedir. Profil tabanlı yöntemde her dilin profillerini bularak doğruluk oranı elde edilir. Örneğin Türkçe'ye ait bir metnin profilini oluşturmak istersek:

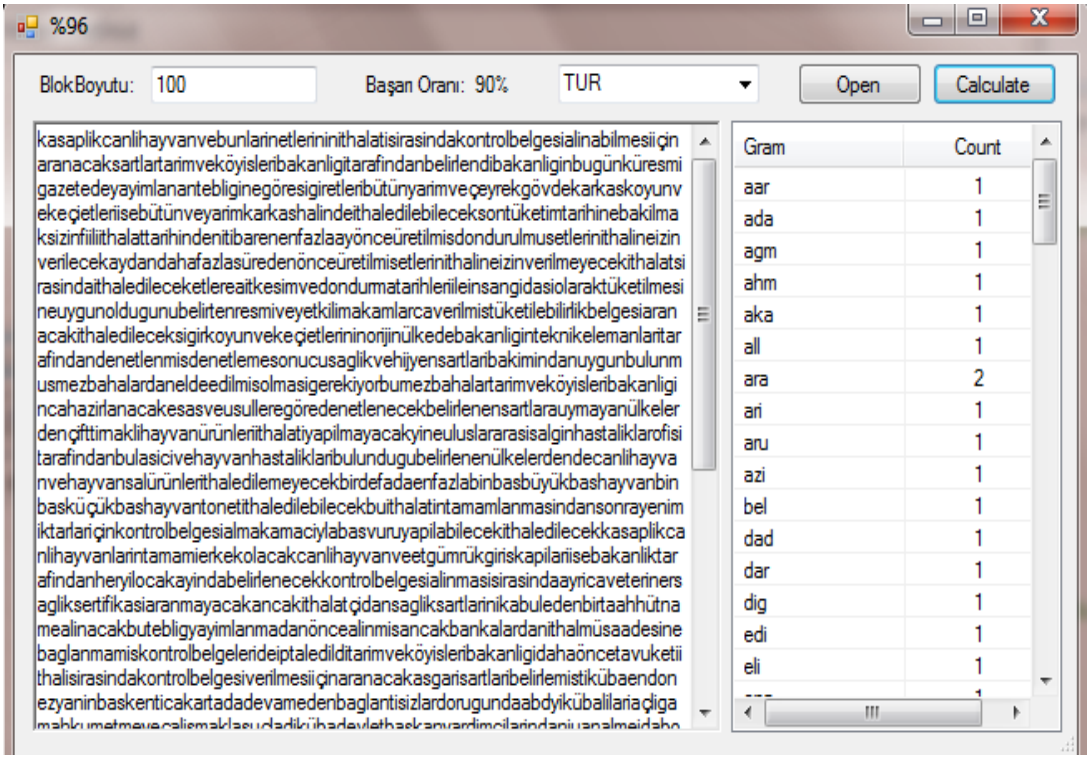
'Ali ata bak', cümlesinde oluşan trigram seti sırasıyla 'ali', 'lia', 'iat', 'ata', 'tab', 'aba', 'bak' dır. Burada her defasında sıradaki trigram bir önceki trigramla karşılaştırıp saydırılır. Böylece metnin bag of words değerleri bulunmuş olur. Algoritma akışı aşağıdaki gibidir.

- 15 dil içerisinden istenilen dil seçilir ve dosyadan bir metin açılır.
- Metin istenilen boyuta göre bloklara ayrılır.(1 KB lık dosyalara)
- Test metninde ayırdığımız her sample için geçen trigram değerleri hesaplanır.
- Benzerlik için metin puanlama yöntemi uygulandığında doküman vektörü ile train veri vektörlerinin nokta çarpımlarından test dokümanının dil puanları elde edilir ve bu puanlardan maksimum olanın test dokümanının dilini bulmada kullanılır.
- Test metnindeki trigram değeriyle, dili bilinen eğitim setindeki trigram değeri aynı olan değerler toplanır. Bu işlem metin puanlandırma olarak adlandırılır.
- Her dil için, dil bazında ayrı ayrı toplam alınır.
- Bu işlem test metninde her blok için yapılır.

- Her dile ait toplam en büyükse o toplam değeri dosya boyutuna bölünüp, yüz değeri ile çarpılarak o dile atanır ve doğruluk oranı yüzdelik değeri ile ifade edilmiş olur.



Şekil 3.2. N-Gram Özellik Çıkarım Yöntemi Program Arayüzü



Şekil 3.3. 'Türkçe' Metin İçin Profil Tabanlı Metodun Başarı Oranı Hesaplanması

Şekil 3.3.'de dili Türkçe seçilen ve dosyadan okunan Türkçe metin 100 byte'lık bloklara ayrılmıştır. Her bir blok, bütün diller için ayrı ayrı önceden belli olan trigram değerleriyle karşılaştırılmıştır. Test metninde hesaplanan trigram değerleriyle, önceden hesaplanmış trigram değerlerinden benzerliği en büyük olan değer o dile yakınsamıştır ve başarı oranı yüzdelik dilim halinde elde edilmiştir. Şekildeki çalışmamızda, test metninin metin puanı Türkçe için en yüksek çıkmış ve ekran görüntüsünde görüldüğü gibi %90 başarı oranı elde edilmiştir. Bu oran diğer diller için de başarılı bir şekilde elde edilmiştir.

3.3.1. Centroid Tabanlı Sınıflayıcı ile Dil Tanıma

Trigram tabanlı dil tanımanın daha yüksek doğruluğu verip vermeyeceğini anlamak için bazı deneyler yapılmıştır. Bu deneylerde dillerin hangi sayısal değerlerle sunulması gerektiğinden, hangi benzerlik yönteminin tercih edilmesi gerektiğine kadar bazı testler yerine getirilmiş ve sonuçlar yorumlanmıştır.

Bu kapsamda yöntemin dil tanıma doğruluğunu etkileyebileceği düşünülen faktörlerden bazıları şunlardır;

- Centroid değeri
- Benzerlik/uzaklık yöntemleri

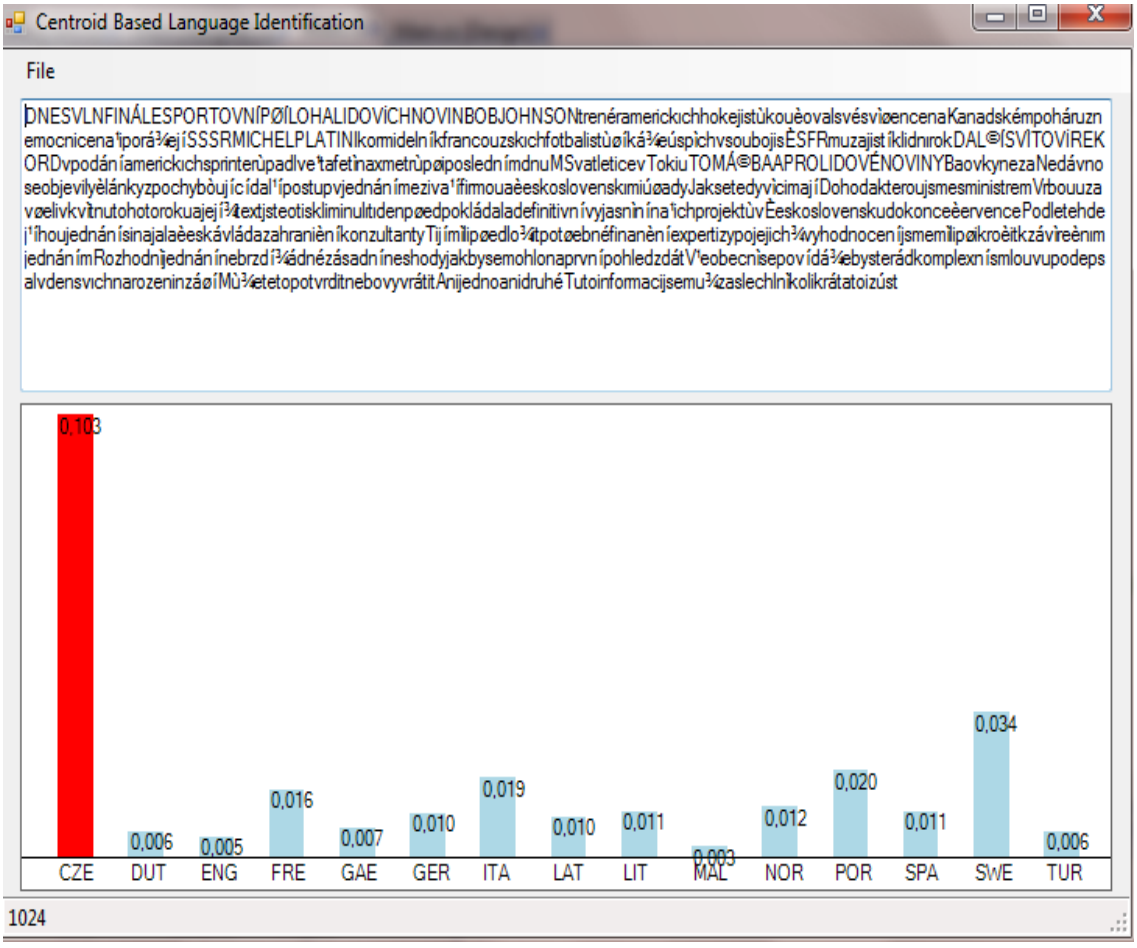
Dil tanımanın başarısına etki edebilecek faktörler sırayla test edilerek dil tanıma için en uygun form bulunmaya çalışılmıştır. Deneyler sonrasında; benzerlik için kosinüs fonksiyonu kullanılması gerektiği anlaşılmıştır. Ayrıca deneyler sırasında sisteme yapılabilecek ekler test edilmiştir.

Dilleri karakterize etmede kullanılabilecek iki seçenekten birisi dillerin ham ortalama sıklıkları diğeri ise dillerin centroid değerleridir. Dillerin ortalama karakter sıklıkları da dilleri bir miktar karakterize etmektedir.

Centroid tabanlı sınıflayıcı da peformans analizi yapmak için iş akışı aşağıdaki gibidir:

- Eğitim seti 1 KB' lik dosyaların 100 byte şeklinde ayrılarak elde edilmesiyle oluşturulmuştur ve her dil için 100 ayrı kayıt değeri vardır. Her sınıf için 10 kayıt tutulmuştur ve 15 adet centroid vektörü elde edilmiştir.

- Döküman değerinin centroid değeri için eldeki dökümanın ortalaması alınır.
- Test metninde gelen dökümanın centroid değeri hesaplanır ve önceden hesaplanmış centroid değerine en yakın değerin diline atanır.
- Centroid sınıflayıcı ile kategorisi bilinmeyen bir metni sınıflandırmada cosine benzerlik yönteminden yararlanılmıştır. Bir test dokümanının hangi kategoriye ait olduğunu bulmak için öncelikle test dokümanı ile centroid vektörleri arasındaki benzerlikler hesap edilir ve test dokümanı kendine en yakın kategoriye atanır. En yakın kategorinin elde edilmesinde maksimum benzerlik formülünden faydalanılır.



Şekil 3.4 : Centroid Tabanlı Sınıflayıcı İçin Dil Tanıma Sonuçları

Şekil 3.4. 'de görüldüğü gibi dosyadan okunan 1 KB'lık metnin 100 bytelık dosyalara ayrılarak, centroid tabanlı sınıflayıcı ile dili tanıma başarı oranı yüksektir. Girilen metnin cosine benzerliğinden 0.103 oranında Cezayirce diline başarıyla atandığı görülmektedir. Sistemde 15 dilden istenilen bir dili seçebileceğimiz dosya

menüsünde Centroid Tabanlı Sınıflayıcı butonuna basıldığında, sistemde algoritma basamak basamak uygulanır. Dosya menüsünde seçilen, dili bilinmeyen metni kosinüs benzerliğini kullanarak her dil için değer elde eder. Oran olarak en büyük değer elde ettiği dile kendisini atar. Çalışma sonucunda sistemin test metinlerini başarıyla tanıdığı görülmüştür.

Görüldüğü gibi 1 KB ile sınırlandırılmış test metninin tüm dillere olana kosinüs uzaklığı hesaplanmıştır. Seçilen metni 0.103 değeriyle Cezayirce diline atamıştır.

3.3.2. K-Ortalamalar Kümeleme Algoritması ile Dil Tanıma

K-Ortalamalar'da sürekli giriş değer verisi olan trigram frekans değerlerinin, Tanagra veri madenciliği yazılımı kullanılarak kümeleme performans analizi yapılmıştır. Verilerin Tanagra ortamında çalışması için Weka dosya formatında hazırlanmıştır.10 deneme yapılarak kümeleme yapması beklenen deneyimiz neticesinde maximum 5 iterasyon yapılarak kümeleme analizi bitmiştir. Bunun sebebi ise 5 iterasyon sonunda R-Square değerinin artık sabitlenmesidir. Yani verilerin kümeleme yapılırken, küme merkezlerine olan uzaklık değeri 5 iterasyondan sonra değişmemektedir. R-Square değeri kayıtların, küme merkezlerine olan uzaklıkların temsilidir. Elde edilen kümeleme analizi Şekil 3.6. ve Şekil 3.7.'de verilmiştir ve doğruluk oranı 0.26 olarak hesaplanmıştır.

Number of trials	5
Trial	R-square
1	0,275268
2	0,263174
3	0,267989
4	0,260289
5	0,278867

Şekil 3.5. 150 Kayıt İçin K-Ortalamalar Algoritmasına Göre Kümelemede Hata Değerleri

Clusters	15		
Cluster	Description	Size	WSS
cluster n°1	c_kmeans_1	84	20572,7791
cluster n°2	c_kmeans_2	1	0,0000
cluster n°3	c_kmeans_3	32	6475,5257
cluster n°4	c_kmeans_4	74	28580,7260
cluster n°5	c_kmeans_5	174	42539,1945
cluster n°6	c_kmeans_6	90	30849,5200
cluster n°7	c_kmeans_7	19	6766,4547
cluster n°8	c_kmeans_8	98	20223,3375
cluster n°9	c_kmeans_9	91	34938,3294
cluster n°10	c_kmeans_10	13	1505,6140
cluster n°11	c_kmeans_11	82	35595,6027
cluster n°12	c_kmeans_12	69	16624,0211
cluster n°13	c_kmeans_13	12	2605,1201
cluster n°14	c_kmeans_14	573	107265,1352
cluster n°15	c_kmeans_15	88	26428,8145

Şekil 3.6. 1500 Kayıtın Kümeleme Analiz Sonuçları

Clusters	15		
Cluster	Description	Size	WSS
cluster n°1	c_kmeans_1	121	29421,2738
cluster n°2	c_kmeans_2	1	0,0000
cluster n°3	c_kmeans_3	6	466,3031
cluster n°4	c_kmeans_4	1	0,0000
cluster n°5	c_kmeans_5	1	0,0000
cluster n°6	c_kmeans_6	1	0,0000
cluster n°7	c_kmeans_7	6	1081,1604
cluster n°8	c_kmeans_8	3	364,1557
cluster n°9	c_kmeans_9	1	0,0000
cluster n°10	c_kmeans_10	4	1118,0969
cluster n°11	c_kmeans_11	1	0,0000
cluster n°12	c_kmeans_12	1	0,0000
cluster n°13	c_kmeans_13	1	0,0000
cluster n°14	c_kmeans_14	1	0,0000
cluster n°15	c_kmeans_15	1	0,0000

Şekil 3.7. 150 Kayıtın Kümeleme Sonuçları

Yapılan çalışmalar sonucunda görülmüştür ki; k-ortalamalar kümeleme algoritması homojen bir şekilde kümeleme yapamamıştır. Bu yüzden hangi kümenin hangi kümeye girdiği tam olarak bilinemediği için k-ortalamalar algoritması istenilen sonucu verememiştir. Bunun sebebi olarak dillere ait trigram değerlerinin birbirlerine yakınlığı sebebiyle diller iç içe girerek birbirlerinden ayırt edilememiştir. Metinsel tabanlı dil tanıma sistemimizdeki 300 öznitelik vektörüyle k-ortalamalar algoritması üzerinde eğitimi sırasında iyi bir referans elde edilemediği için test etmeye gerek görülmemiştir. Sonuç kısmında kümeleme ve sınıflandırma sonuçları bu durum göz önüne alınarak yorumlanmıştır.

3.3.3. Bulanık C Ortalamalar Kümeleme Algoritması ile Dil Tanıma

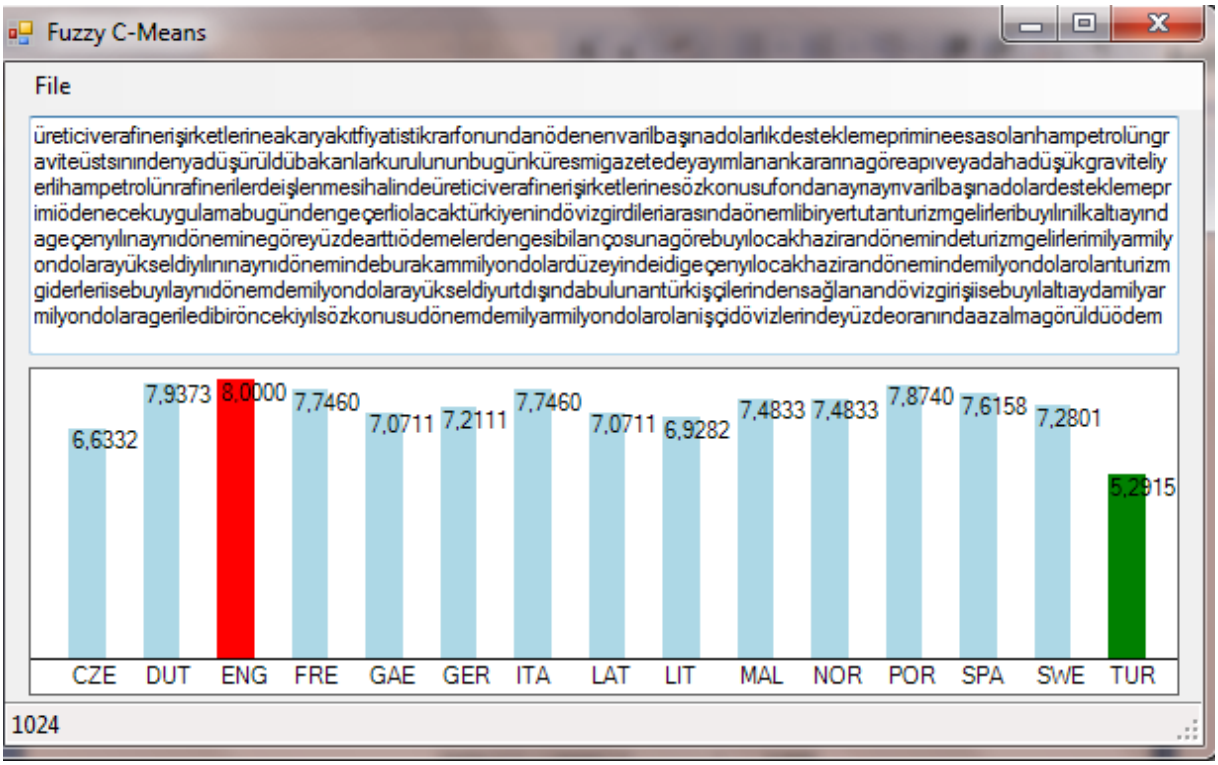
Trigram veri seti kullanılarak elde edilen 300 öznitelik değeri bulanık kümeleme yöntemiyle dil tanıma yapılması hedeflenmiştir. 100 KB lık dosyalardan oluşturulmuş eğitim veri seti 1KB'lık dosyalara ayrılmıştır. 1500 kayıtın ve 15 sınıfın bulunduğu dil tanıma sisteminin bulanık kümeleme yöntemi kullanılarak izlenen adımlar aşağıdaki gibidir.

Adım1: Küme sayısı c = 15, bulanıklık indeksi m(fuzzication değeri) değeri 2 seçilmiştir. Üyelik dereceleri matrisini gösteren U matrisi için başlangıç değerleri belirlenir. Bunun için her bir kayıttan dili temsil eden 15 ayrı centroid değeri üretilmişitr.

Adım 2: Centroid (V) küme prototiplerinin rasgele üretildiği varsayılırsa bu değerler kullanılarak üyelik dereceleri matrisini aşağıdaki formüle gore hesaplanmıştır.

$$u_{ik} = \sum_{l=1}^{c} \left(\left\| v^i, x_k \right\| \Big/ \left\| v^l, x_k \right\| \right)^{-(2)/m-1} \qquad (3.1.)$$

Adım 4: Eğer elde edilen centroid değeri önceki elde edilen centroid değerinden farkı epsilon(0.1) dan küçük değilse 2. Adıma gidilmiştir.



Şekil 3.8. Bulanık KümelemeAlgoritması İle Dil Tanıma Performans Sonuçları

Şekil 3.8. ‘de görüldüğü gibi dosyadan okunan test metni BCO algoritması kullanılarak doğru bir şekilde sınıflandırma yapması hedeflenmiştir. Sistemde 15 dilden istenilen bir dili seçebileceğimiz dosya menüsünde öncelikle **CMeans** butonuna basılarak, eğitim verisi üzerinde bulanık kümeleme algoritmasına göre kümeleme yapılması yapılmıştır. Kümeleme işleminden sonra metnin dilinin belirlenmesi için yani, onun sınıfını tespit etmek için tüm kümelere üyelik değerlerinin maksimum üyelik değerinin dikkate alınması gerekir. Sistem kümeleme işlemini Şekil 3.8. deki sonuç maksimum 8 iterasyon yaparak bitirmiştir. Daha sonra test aşamasında **Execute** butonuna kullandığımızda dosya menüsünde seçilen, dili bilinmeyen metni sınıfının belirlenmesi sağlanmıştır. Test metninde oluşturulan öznitelik değerlerinin, bütün kümeler için önceden hesaplanmış centroid değererine olan uzaklığı öklit uzaklığına göre hesaplanmıştır. Her bir dil için hesaplanan bu uzaklık değerlerinin arasında en düşük değerin bulunduğu dil o metnin dilidir. Değer olarak elde edilen en küçük değeri en yakın kümeye atar ve o kümenin elemanı olur. Yani küme merkezleri birbirine en yakın olan değere kümeleme yapılır. Çalışma sonucunda sistemin test metinlerini başarıyla tanıdığı görülmüştür. Şekil 3.8.'de örnek test metninin tüm diller için sınıflandırma analizi yapılmıştır ve verilen test

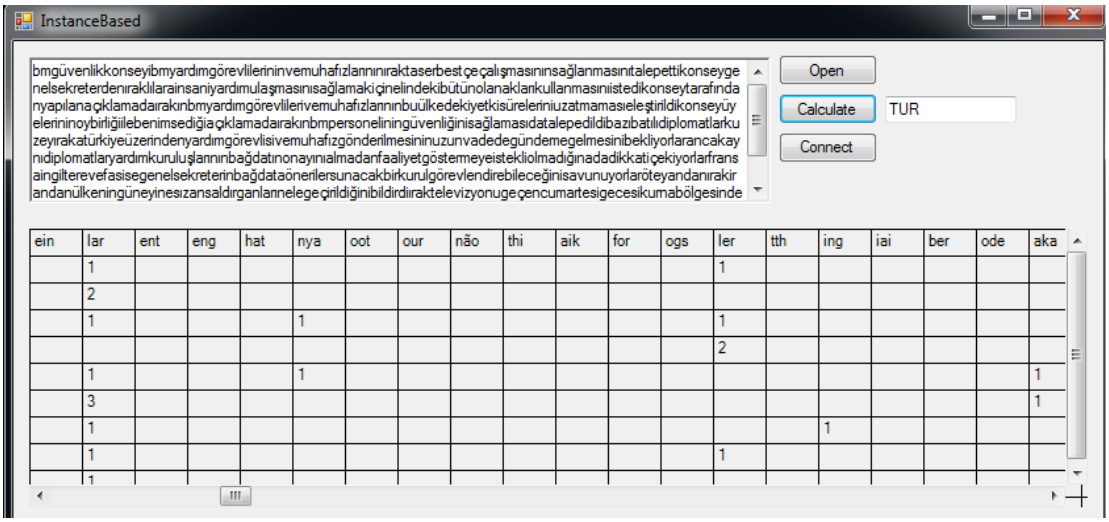
metnini 5.2915 minimum uzaklık değeriyle Türkçe dili olduğu tespit edilmiştir. Kendisine en uzak küme ise 8.000 değeriyle İngilizce dili olduğu görülmüştür. Programımızda girilen test metnine en uzak dil kırmızı, en yakın yani doğru olarak belirlediği sınıf yeşil ile boyanmıştır. Uzaklık değerleri virgülden sonra üç basamak hassasiyetinde tutulmuştur. Bulanık kümeleme için elde edilen centroid değerleri ve üyelik matris değerleri program klasörü içerisinde dinamik olarak tutulmaktadır.

3.3. Trigram Eğitim Verileriyle Örnek Tabanlı Dil Tanıma Metodu

Elde edilen trigram frekans değerlerinin sınıflandırma algoritmalarına giriş verisi olarak verilip performans analizinin yapılabilmesi için her dile ait örnek elde edilmesi gerekmektedir. Bunun için örnek tabanlı yöntemi kullanılmıştır ve algoritma akışı aşağıdaki gibidir.

- Dillere ait 100 KB ve 1KB ile sınırlandırılarak seçilmiş doküman verileri seçilir.
- Bu verilerden elde edilen trigram frekanslarının sınıflandırma algoritmalarına giriş olarak uygulamamız gerekmektedir. Bunun gerçekleştirilebilmesi için de her dil için aynı boyutlu test dokümanının trigram frekans değerleri hesaplanır.
- Trigram frekans değerleri hesaplanmadan önce, dosya boyutu 100 KB ise test metni 100'er byte olarak ayrılır ve her 100 byte için test dokümanının frekans değerleri önceden hesaplanmış en önemli eğitim trigram frekanslarına göre hesaplanmıştır.
- Dosya boyutu 100 KB ile sınırlandırılmış ise dosya 1 KB 1 KB ayrılır ve her 1 KB için test dokumanının frekans değerleri önceden hesaplanmış en önemli eğitim trigram frekanslarına göre hesaplanmıştır.

  Şekil 3.9. da programımızdan 'Türkçe' için örnek tabanlı metod kullanılarak trigram frekans değerlerinin hesap edilişi gösterilmiştir.

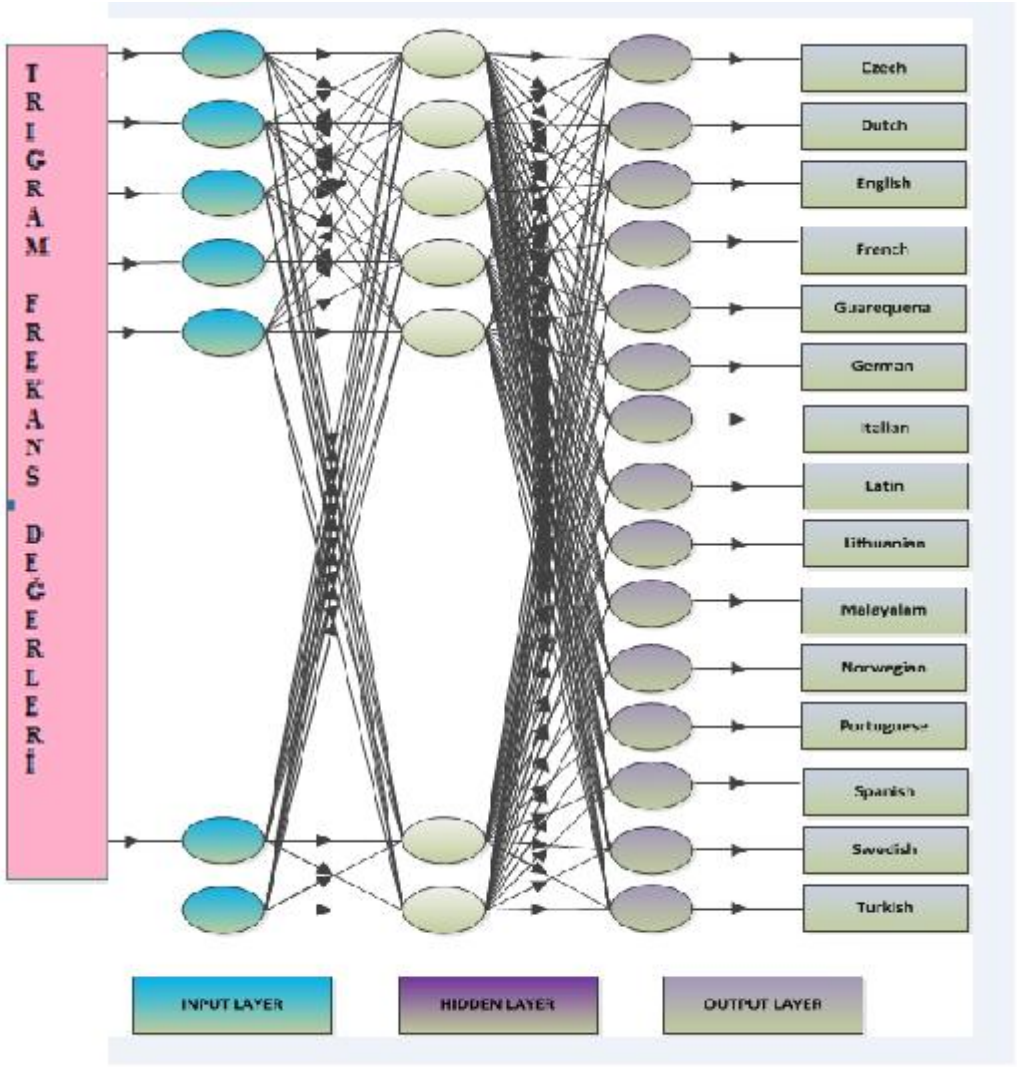


Şekil 3.9. 1 KB 'Türkçe' İçin Örnek Tabanlı ile Trigram Frekans Hesabı

Örnek tabanlı yöntemden elde edilen gramlar, makina öğrenmesi algoritmalarının kullanımına uygundur. Bunun için diller için oluşturulan her bir veri dosya boyutuna göre çalışmamızda kullanacağımız Yapay Sinir Ağları(YSA), Destek Vektör Makinaları sınıflandırma algoritmalarına giriş verisi olarak uygulanacaktır ve performans analizi yapılacaktır.

3.4.1.Yapay Sinir Ağları Sınıflandırma Algoritması ile Dil Tanıma

Metin tabanlı dil tanıma yaparken yapay sinir ağları sınıflandırma algoritmalarından feed forward backpropagation öğrenme algoritması kullanılmıştır. Şekil 3.10.'da çok katmanlı öğrenme mimarisi verilmiştir. Sistem için trigram frekans değeri eğitim metninden elde edilen 300 öznitelik vektörü giriş düğüm, sınıflandırma yapmasını istediğimiz dillerden oluşan 15 çıkış düğüm olarak ele alınmıştır. Sistemin başarı sınaması Tanagra hazır veri madenciliği yazılımı kullanılarak yapılmıştır.



Şekil 3.10. Metin Tabanlı Sınıflandırmada MLP Yapısı

Örüntü tanımada, algoritmaların doğruluk oranı ' doğruluk(precision) ve yeniden çağırım(recall)' değerleriyle verilir. Yeniden çağırım(recall) değeri metinde tahmin edilen ile gerçek olan verilerin kesişiminin ilgili veriye bölümüdür. Precision ise gerçekte tahmin edilen ile gerçek olan verilerin olası tüm verilere bölümüdür.

$$precision = \frac{\left|\{relavent\_document\} \cap \{retrieved\_documents\}\right|}{\left|\{retrieved\_documents\}\right|} \quad (3.2.)$$

$$recall = \frac{\left|\{relavent\_document\} \cap \{retrieved\_documents\}\right|}{\left|\{relavent\_documents\}\right|} \quad (3.3.)$$

		correct result / classification	
		E1	E2
obtained result / classification	E1	tp (true positive)	fp (false positive)
	E2	fn (false negative)	tn (true negative)

(3.4.)

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn} \quad (3.5.)$$

Bu bilgilerden yola çıkılarak, Şekil 3.10'da dillere ait Karıştırma(Confusion) matrix'de yeniden çağırım(yeniden çağırım(recall)) ve 1-doğruluk(precision) değerleri verilmiştir. Cezayirce dili için toplamda 10 metnin 10'uda Cezayircedir. Bu durumda yeniden çağırım(yeniden çağırım(recall)) değeri 1 , 1-doğruluk(precision) değeri 0 bulunmuştur. Bu parametrelere göre MLP ağının tek bir ara katmanı olup bu katmadaki nöron sayısı 10 olarak belirlenmiştir. Ayrıca ağın öğrenme oranı 0.15 ve durma kriteri de 100 iterasyon olarak belirlenmiştir. 100 iterasyon sonunda ağın hata oranı 0.08 olarak elde edilmiştir.

Error rate			0,0800													
Values prediction			Confusion matrix													
Value	Recall	1-Precision		CZE	DUT	ENG	FRE	GAE	GER	ITA	LAT	LIT	MAL	NOR	POR	SPA	SWE	TUR
CZE	1,0000	0,0000	CZE	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DUT	1,0000	0,0000	DUT	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
ENG	0,6000	0,0000	ENG	0	0	6	0	3	1	0	0	0	0	0	0	0	0	0
FRE	1,0000	0,0000	FRE	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
GAE	1,0000	0,2857	GAE	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
GER	1,0000	0,2308	GER	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
ITA	0,8000	0,1111	ITA	0	0	0	0	0	0	8	0	0	0	0	1	1	0	0
LAT	0,9000	0,0000	LAT	0	0	0	0	0	0	0	9	1	0	0	0	0	0	0
LIT	1,0000	0,0909	LIT	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
MAL	1,0000	0,0000	MAL	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
NOR	0,9000	0,0000	NOR	0	0	0	0	0	1	0	0	0	0	9	0	0	0	0
POR	1,0000	0,0909	POR	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
SPA	0,7000	0,1250	SPA	0	0	0	0	1	0	1	0	0	0	0	0	7	1	0
SWE	1,0000	0,0909	SWE	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
TUR	0,9000	0,0000	TUR	0	0	0	0	0	1	0	0	0	0	0	0	0	0	9
			Sum	10	10	6	10	14	13	9	9	11	10	9	11	8	11	9

Şekil 3.11. MLP Ağında 150 Kayıt İçin Dillerin Ait Olduğu Sınıf Matrisi

Şekil 3.11. Tanagra ile elde edilen sınıflandırma sonuçları sunulmaktadır. Sonuçlar doğruluk(precision) ve yeniden çağırım (recall) ölçümleri esas alınarak elde edilmiştir. Ayrıca, model değerlendirme için 15 x 15 boyutunda bir karıştırma (Confusion) matrisi ile sonuçlar sunulmuştur. Elde edilen değerler, eğitim verisi için doğruluğu sunmakta olup, test verisi için doğru tanımayı elde etmede cross validation tekniği kullanılmıştır. Cross validation için tercih edilen n değeri 10'dur. Her seferinde verinin yüzde doksanı (n-1 fold) eğitim için, yüzde 10'u ise test için ayrılmıştır. Sonuçlar şekil 3.12. de verilmiştir.

150 kayıtın bulunduğu veri 10-fold cross validation uygulanarak hata oranı 0.54 olarak hesaplanmıştır.

Values prediction		
Value	Recall	1-Precision
CZE	1,0000	0,6154
DUT	0,6000	0,1429
ENG	0,4000	0,5000
FRE	0,3000	0,5714
GAE	0,5000	0,5455
GER	0,6000	0,4545
ITA	0,5000	0,4444
LAT	0,4000	0,5556
LIT	0,3000	0,2500
MAL	0,4000	0,3333
NOR	0,8000	0,4667
POR	0,3000	0,7000
SPA	0,3000	0,7692
SWE	0,2000	0,7143
TUR	0,3000	0,5714

Şekil 3.12. 150 Kayıt İçin Dillerin Test Başarı Sonuçları

1500 kayıtın bulunduğu sistem analizi için ise MLP yapısında ara katmandaki nöron sayısı 10 olan, öğrenme oranı 0.15 olan bir ağ oluşturulmuştur. Şekil 3.13'de maksimum 100 iterasyonda 0.0940 hata oranı elde edilmiştir.

0,2180														
Confusion matrix														
	CZE	DUT	ENG	FRE	GAE	GER	ITA	LAT	LIT	MAL	NOR	POR	SWE	TUR	Sum
CZE	93	0	0	1	1	0	1	1	0	0	0	0	3	0	100
DUT	0	88	0	0	2	4	0	1	0	0	4	0	0	1	100
ENG	0	0	92	1	3	0	0	0	0	0	1	0	1	2	100
FRE	0	1	0	88	0	1	0	6	0	0	1	1	1	1	100
GAE	5	0	6	0	69	1	0	11	0	0	2	1	4	1	100
GER	0	2	0	0	2	94	1	0	0	0	1	0	0	0	100
ITA	1	2	0	0	0	2	89	0	0	0	2	1	0	3	100
LAT	5	0	0	4	3	0	1	172	1	2	1	7	2	2	200
LIT	0	0	0	0	1	0	0	0	24	75	0	0	0	0	100
MAL	0	0	0	0	0	1	0	0	83	16	0	0	0	0	100
NOR	0	2	0	0	0	1	0	1	0	0	88	1	7	0	100
POR	1	0	0	0	0	0	3	5	0	0	0	91	0	0	100
SWE	7	0	0	1	3	0	0	1	0	1	5	0	79	3	100
TUR	4	0	0	0	0	1	1	0	0	0	0	1	3	90	100
Sum	116	95	98	95	84	105	96	198	108	94	105	103	100	103	1500

3.13. MLP Ağında 1500 Kayıt İçin Dillerin Ait Olduğu Sınıf Matrisi

1500 kayıtın bulunduğu veri setine cross validation uygulanmıştır.Sistemin test etmedeki hata oranı 0.218 olarak elde edilmiştir. Sistemin dillere ait test

etmedeki 1-doğruluk(precision) ve yeniden çağırım(yeniden çağırım(recall)) değerleri Şekil 3.14.'de verilmiştir.

Values prediction

Value	Recall	1-Precision
CZE	0,9300	0,1983
DUT	0,8800	0,0737
ENG	0,9200	0,0612
FRE	0,8800	0,0737
GAE	0,6900	0,1786
GER	0,9400	0,1048
ITA	0,8900	0,0729
LAT	0,8600	0,1313
LIT	0,2400	0,7778
MAL	0,1600	0,8298
NOR	0,8800	0,1619
POR	0,9100	0,1165
SWE	0,7900	0,2100
TUR	0,9000	0,1262

Şekil 3.14. 1500 Kayıt İçin Dillerin Test Başarı Sonuçları

3.4.2. Destek Vektör Makinası Sınıflandırma Algoritmasi ile Dil Tanıma

DVM algoritmasının, ikiden fazla sınıflı uygulamalarda kullanılan C-SCV algoritması üzerinde performans analizi yapılmıştır ve LibDVM kütüphanesi kullanılmıştır. C-SCV algoritması, ikiden fazla sınıflayıcısı olan uygulamalarında kullanılan DVM 'in özel algortimalarındandır. Çalışmamızda 15 sınıf olduğu için C-SCV algoritması kullanılmıştır.

150 kayıdın bulunduğu destek vektör makinası algoritması için Lineer kernel kullanılmıştır. Sistemin, Şekil 3.15.'de görüldüğü gibi 150 kayıt için hiç hata vermediği görülürken, Şekil 3.15.'de 1500 kayıt için 0,0673 oranında hata vermiştir.

Error rate			0,0000													
Values prediction			Confusion matrix													
Value	Recall	1-Precision		CZE	DUT	ENG	FRE	GAE	GER	ITA	LAT	LIT	MAL	NOR	POR	SPA	SWE	TUR
CZE	1,0000	0,0000	CZE	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DUT	1,0000	0,0000	DUT	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
ENG	1,0000	0,0000	ENG	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
FRE	1,0000	0,0000	FRE	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
GAE	1,0000	0,0000	GAE	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
GER	1,0000	0,0000	GER	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
ITA	1,0000	0,0000	ITA	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
LAT	1,0000	0,0000	LAT	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
LIT	1,0000	0,0000	LIT	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
MAL	1,0000	0,0000	MAL	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
NOR	1,0000	0,0000	NOR	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
POR	1,0000	0,0000	POR	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
SPA	1,0000	0,0000	SPA	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
SWE	1,0000	0,0000	SWE	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
TUR	1,0000	0,0000	TUR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
			Sum	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Şekil 3.15. 150 Kayıt İçin DVM Karıştırma(Confusion) Matrisi

0,0673														
Confusion matrix														
	CZE	DUT	ENG	FRE	GAE	GER	ITA	LAT	LIT	MAL	NOR	POR	SWE	TUR	Sum
CZE	100	0	0	0	0	0	0	0	0	0	0	0	0	0	100
DUT	0	100	0	0	0	0	0	0	0	0	0	0	0	0	100
ENG	0	0	100	0	0	0	0	0	0	0	0	0	0	0	100
FRE	0	0	0	100	0	0	0	0	0	0	0	0	0	0	100
GAE	0	0	0	0	100	0	0	0	0	0	0	0	0	0	100
GER	0	0	0	0	0	100	0	0	0	0	0	0	0	0	100
ITA	0	0	0	0	0	0	100	0	0	0	0	0	0	0	100
LAT	0	0	0	0	0	0	0	200	0	0	0	0	0	0	200
LIT	0	0	0	0	0	0	0	0	59	41	0	0	0	0	100
MAL	0	0	0	0	0	0	0	0	59	41	0	0	0	0	100
NOR	0	0	0	0	0	0	0	0	0	0	100	0	0	0	100
POR	0	0	0	0	0	0	0	0	0	0	0	100	0	0	100
SWE	1	0	0	0	0	0	0	0	0	0	0	0	99	0	100
TUR	0	0	0	0	0	0	0	0	0	0	0	0	0	100	100
Sum	101	100	100	100	100	100	100	200	118	82	100	100	99	100	1500

Şekil 3.16. 1500 Kayıt İçin DVM Karıştırma(Confusion) Matrisi

Şekil 3.16. Tanagra ile elde edilen sonuçları sunmaktadır. Sonuçlar doğruluk(precision) ve yeniden çağırım(recall) ölçümleri esas alınarak elde edilmiştir. Ayrıca, model değerlendirme için 15 x 15 boyutunda bir karıştırma (Confusion) matrisi ile sonuçlar sunulmuştur. Elde edilen değerler, eğitim verisi için doğruluğu sunmakta olup, test verisi için doğru tanımayı elde etmede cross validation tekniği kullanılmıştır. Cross validation için tercih edilen n değeri 10'dur. Her seferinde verinin yüzde doksanı (n-1 fold) eğitim için yüzde 10'u ise test için ayrılmıştır. 150 kayıtın bulunduğu veri 10-fold cross validation uygulanarak hata oranı 0.00 olarak hesaplanmıştır. Test başarı Sonuçlar Şekil 3.17.'de verilmiştir.

Values prediction		
Value	Recall	1-Precision
CZE	1,0000	0,8333
DUT	0,6667	0,0000
ENG	0,6000	0,0000
FRE	0,6667	0,0000
GAE	0,0000	1,0000
GER	1,0000	0,7500
ITA	1,0000	0,3333
LAT	1,0000	0,5000
LIT	1,0000	0,0000
MAL	1,0000	0,0000
NOR	1,0000	0,0000
POR	0,6000	0,0000
SPA	0,2500	0,7500
SWE	0,6667	0,0000
TUR	0,7500	0,0000

Şekil 3.17. 150 Kayıtın Bulunduğu Veri Seti İçin DVM Test Başarı Oranı

1500 kayıtın bulunduğu veri seti üzerinde cross validation (n-1) uygulanarak test edilmişrir.Test verisi olarak ayrılarak hata oranı 0.2253 olarak hesaplanmıştır. Sistemin dillere ait yeniden çağırım(recall) ve 1-doğruluk(precision) sonuçları Şekil 3.18.'de verilmiştir.

Error rate			0,2253														
Values prediction			Confusion matrix														
Value	Recall	1-Precision		CZE	DUT	ENG	FRE	GAE	GER	ITA	LAT	LIT	MAL	NOR	POR	SWE	TUR	Sum
CZE	0,9100	0,2835	CZE	91	0	0	0	0	1	0	1	0	0	0	0	5	2	100
DUT	0,9400	0,0309	DUT	0	94	0	1	0	1	0	1	0	0	2	0	1	0	100
ENG	0,9300	0,0700	ENG	1	1	93	0	4	0	0	0	0	0	0	0	0	1	100
FRE	0,8800	0,0833	FRE	1	1	0	88	1	1	0	4	0	0	1	2	1	0	100
GAE	0,6800	0,2093	GAE	10	0	7	2	68	1	0	4	0	0	0	0	6	2	100
GER	0,9400	0,0408	GER	1	1	0	0	1	94	0	0	0	0	1	0	0	2	100
ITA	0,9200	0,0213	ITA	2	0	0	0	0	0	92	3	0	0	0	2	0	1	100
LAT	0,9050	0,0995	LAT	5	0	0	3	4	0	1	181	0	0	1	0	2	3	200
LIT	0,0700	0,9314	LIT	0	0	0	0	0	0	0	0	7	93	0	0	0	0	100
MAL	0,0600	0,9394	MAL	0	0	0	0	0	0	0	0	94	6	0	0	0	0	100
NOR	0,8900	0,1010	NOR	1	0	0	1	1	0	0	1	0	0	89	0	6	1	100
POR	0,9400	0,0408	POR	0	0	0	0	0	0	1	3	1	0	0	94	0	1	100
SWE	0,7500	0,2424	SWE	12	0	0	1	4	0	0	2	0	0	5	0	75	1	100
TUR	0,9000	0,1346	TUR	3	0	0	0	2	1	0	1	0	0	0	0	3	90	100
			Sum	127	97	100	96	86	98	94	201	102	99	99	98	99	104	1500

Şekil 3.18. 1500 Kayıtın Bulunduğu Veri Seti İçin DVM Karıştırma(Confusion) Matrisi

3.4.Test Analiz Sonuçları

Çalışmamızın test analiz sonuçları 150 kayıt ve 1500 kayıt olmak üzere dosya boyutuna göre verilmiştir. 150 kayıtlık dosya 1 Kb'lik metin dosyalarının 100 byte'lara ayrılmasıyla, 1500 kayıtlık dosyalar ise 100 Kb'lık metin dosyalarının 1 Kbl'lık dosyalara bölümlenerek oluşturulmuşturve bu dosyalara sınıflandırma ve kümeleme algoritmaları uygulanarak dil tanıma gerçekleştirilmiştir.

Sınıflandırma algoritmaları olarak Centroid sınıflayıcı, MLP, DVM ve K-Ortalamalar ve BCO Kümeleme algoritması kullanılmış ve başarı sonuçları Tablo 3.3. ve Tablo 3.4.'te verilmiştir. Görüldüğü gibi iki farklı kayıt türü olan 150 ve 1500 kayıt için elde edilen başarı sonuçları sınıflandırma algoritmaları ile her dil için başarı sonucu olarak verilmiştir. Dil tanımada sınıflandırma ve kümeleme algoritmalarıyla yönteme bağlı olarak farklı ölçüm kriterleri kullanımıştır. Nitekim, centroid tabanlı sınıflayıcı için kosinüs benzerliğinin maksimum değeri K-Ortalamalar ve BCO algoritmasına göre dillere ait öznitelik vektörkleri ile kümelerin centroid değerleri arasındaki minimum öklit uzaklığı kullanılmıştır. .MLP ve DVM için doğruluk(precision) ve yeniden çağırım(recall) değerlerine göre sonuçlar verilmiştir. Tanagra veri madenciliği yazılımı üzerinden eğitim işleminde yapılan kümeleme homojen bir kümeleme sağlayamadığı için iyi bir referans olmadığı görülmüştür.

Tablo 3.3. Dillerin Sınıflandırma ve Kümeleme Algoritmalarına Göre Tset Başarı Sonuçları(150 Kayıt)

	CentroidTabanlıSınıflayıcı	MLP	BCO	C-CSV
CZE	**2.328**	**1.00**	**5.385**	**1.00**
DUT	**0.764**	**1.00**	**9.380**	**1.00**
ENG	**0.882**	**1.00**	**9.591**	**1.00**
FRE	**0.874**	**1.00**	**8.366**	**1.00**
GAE	**1.227**	**1.00**	**8.185**	**1.00**
GER	**0.616**	**1.00**	**8.00**	**1.00**
ITA	**0.676**	**0.85**	**8.660**	**1.00**
LAT	**0.855**	**0.88**	**8.426**	**1.00**
LIT	**1.463**	**0.75**	**6.633**	**1.00**
MAL	**1.201**	**0.90**	**8.214**	**1.00**
NOR	**0.929**	**0.78**	**9.380**	**1.00**
POR	**0.804**	**0.92**	**8.811**	**1.00**
SPA	**0.767**	**0.97**	**9.273**	**1.00**
SWE	**0.966**	**0.66**	**8.000**	**1.00**
TUR	**0.766**	**0.88**	**6.245**	**1.00**

Tablo 3.4. Dillerin Sınıflandırma ve Kümeleme Algoritmalarına Göre Test Başarı Sonuçları(1500 Kayıt)

	CentroidTabanlıSınıflayıcı	MLP	BCO	C-CSV
CZE	**0.373**	**0.838**	**5.385**	**1.000**
DUT	**0.396**	**0.933**	**9.380**	**0.666**
ENG	**0.396**	**0.963**	**9.591**	**0.600**
FRE	**0.376**	**0.961**	**8.062**	**0.666**
GAE	**0.374**	**0.592**	**8.185**	**0.000**
GER	**0.400**	**0.966**	**8.000**	**1.000**
ITA	**0.373**	**0.933**	**8.660**	**1.000**
LAT	**0.332**	**0.925**	**8.426**	**1.000**
LIT	**0.335**	**0.130**	**6.633**	**1.000**
MAL	**0.305**	**0.266**	**8.124**	**1.000**
NOR	**0.389**	**0.777**	**9.380**	**1.000**
POR	**0.370**	**0.848**	**8.00**	**0.600**
SPA	**0.374**	**0.821**	**9.273**	**0.250**
SWE	**0.357**	**0.718**	**7.000**	**0.666**
TUR	**0.375**	**0.911**	**6.245**	**0.750**

Tablo 3.5.'de 1 KB boyutlu dosyada her bir dil için test metninin sınıflandırma ve kümeleme algoritmalarına göre,100 farklıdeneme sonucundan elde edilen ortalama test başarı oranları karşılaştırmalı olarak verilmiştir. Test verisi üzerinden sınıflandırma başarı sonucu k-ortalamalar algoritması için verilmemiştir. Çünkü, Tanagra veri madenciliği yazılımı üzerinden eğitim verisi için yapılan kümeleme iyi bir referans olmamıştır. Çalışmada sadece Cezayir dili için yapılan test işleminde, Centroid Sınıflayıcı'da %90,YSA'da %100, BCO'da %90, DVM'de algoritmasında

ise %90 başarı elde edildiği gözlemlenmiştir. Bu tür değerlendirmeler diğer diller için de yapılabilir..

Tablo 3.5. Sınıflandırma ve Kümeleme Algoritmalarının Dil Tanıma Başarı Sonuçları

	Centroid Sınıflayıcı	MLP	BCO	DVM
Cezayirce	%90	%100	%90	%90
Almanca	%90	%80	%90	%90
Ingilizce	%80	%50	%90	%90
Fransızca	%90	%40	%90	%80
Andoa Dili	%90	%40	%90	%60
Almanca	%99	%50	%90	%90
Italyanca	%80	%50	%90	%90
Latince	%90	%40	%90	%90
Litvanyaca	%90	%70	%80	%70
Maltaca	%90	%60	%60	%60
Norveççe	%90	%50	%80	%80
Portekizce	%90	%30	%90	%90
Ispanyolca	%60	%30	%90	%70
Isveççe	%90	%30	%80	%70
Türkçe	%80	%40	%90	%90

4. SONUÇ ve ÖNERİLER

Metin tabanlı dil tanıma gerçek zamanlı sistemler gibi, performansın önem kazandığı durumlarda idealdir. Çünkü bilinen dil tanıma yöntemlerine oranla oldukça hızlı şekilde dil tespitini yerine getirmesi gerekmektedir. Metin tabanlı dil tanıma yazar tanıma, çeviri sistemleri, otomatik cevap sistemleri, spam engelleme ve atak tespiti gibi konularda kullanılabilir. Çünkü bu işlemler de metinlerin fiziksel özelliklerine dayalı olarak yerine getirilebilir. Daha geniş bir bakış açısıyla anormallik tespiti tarzı işlemlerde metin tabanlı yöntem dil tanımada kullanılabilir. Dil tanıma deneyleri bigram ve unigram özellik setleriyle de uygulama yapılmıştır. Fakat trigram özellik setiyle gerçekleştirilmiş uygulamalara göre başarı performansı daha düşük olduğu için bu çalışmaya dahil edilmemiştir. Bu yüzden çalışmamız trigram özellik seti kullanılarak sınıflandırma ve kümelemealgoritmaları ile metin tabanlı dil tanıma sistemlerine uygulamak için temel oluşturmuştur. Dile ait farklı kayıtlar sınıflandırılarak dili bilinmeyen bir metnin önceden belirlenmiş dilin tanınması sağlanmıştır. Sınıflandırma test başarı performansında genel olarak dili bilinmeyen metinlerin tespitinde Centroid sınıflayıcı, BCO kümeleme algoritması, DVM sınıflandırma algoritması MLP'ye göre daha başarılı olduğu tespit edilmiştir. Bu durumda dilin tanınmasında sınıflandırma algoritmasının seçimi dil tanımada önem arz eder. Ayrıca özellik uzayının seçilmesi de performansı etkileyen diğer bir faktördü. Düşük boyutlu özellik uzayı ise ön işlemlerden geçilerek özellik uzayının 300 boyutlu öznitelik vektörü üzerinde durulmuştur ve her bir dile ait metin 60 harften oluşan trigram kombinasyonları tüm diller için ortak olan 300 boyutlu öznitelik frekans değerleriyle temsil edilmiştir. Böylece çalışmamızda önceden yapılmış çalışmalara göre daha yüksek performansta sonuç alınmıştır.

Metin tabanlı dil tanıma sisteminin, dili bilinmeyen metine göre sınıflandırma yapabilmsei için performansı yüksek bir sistem tasarlanması hedeflenmiştir. Bunun için performansı artıran iki önemli iş yapılmıştır. Bunlardan ilki düşük boyutlu özellik uzayı ile iş yapılmıştır. Bunu da harf özellik seti ile trigram kombinasyonları oluşturularak Düşük boyutlu özellik uzayı ise ön işlemlerden geçilerek özellik

uzayının 300 boyutlu öznitelik vektörü üzerinde durulmuştur ve her bir dile ait metin 60 harften oluşan trigram kombinasyonları tüm diller için ortak olan 300 boyutlu öznitelik frekans değerleriyle temsil edilmiştir. Böylece önceden yapılmış çalışmalara göre daha yüksek performansta sonuç alınmıştır. Bunun sebebi ise dili temsil eden niteleyicilerin öneminin artırılmasını sağlamaktır. İkinci problem öznitelik vektörünün sınıflandırılması için uygun sınıflandırma algoritmalarının seçilmesidir. Etkili sınıflandırma yöntemi ise YSA, Centroid Tabanlı Sınıflayıcı, DVM algoritmaları, K-Ortalamalar ve BCO kümeleme algoritmalarıdır. Deneysel çalışmaların sonuçlarına bakıldığında aşağıdaki şu sonuçlara varılabilir:

- Centroid tabanlı sınıflayıcı ile 1 KB metin dosyaları için eğitim ve test işlemleri yapılarak dili bilinmyen metnin sınıflandırması başarıyla gerçekleştiği görülmüştür.
- Yapay sinir ağları eğitimi 0.08 hata oranıyla sağlanmıştır. 150 (1 KB metin) kayıtlık dosyanın hata oranı 0.54, 1500 (100 KB metin)kayıtlık dosyada ise 0.0940 hata oranıyla tanınması sağlanmıştır.
- DVM algoritmasının çoklu sınıflayıcı algoritmalarından C-CSV algoritmasının 150 kayıt için(1 KB metin) eğitimi 0.00 hata oranıyla, 1500 kayıtlık(100 KB metin) dosyada 0.0673 hata oranıyla eğitim yapılmıştır. 150 kayıtlık dosyada 0.00, 1500 kayıtlık dosyada ise 0.8847 doğruluk oranıyla test işlemi gerçekleşmiştir.
- Bulanık C Ortalamalar algoritmasının ise 1 KB ve 100 KB dosyada eğitim işlemi başarıyla elde edilerek öklit uzaklıklarına göre sonuç olarak verilmiştir.

Metin tabanlı dil tanımayı hedefleyen sistemimiz dosya boyutu olan 1 KB ve 100 KB uzunluklu dosyaları temel almıştır. Gelecekte yapılacak çalışmalarda mümkün olan en düşük boyutlu metin dosyaları ile dil tanıma sistemi gerçekleştirilmesi yapılacaktır.Ayrıca trigram özellik seti kullanılarak sınıflandırma ve kümeleme algoritmalarıyla geliştirmiş olduğumuz metin tabanlı dil tanıma sistemi ileriki çalışmalarda mobil tabanlı uygulamaya dönüştürülmesi hedeflenmektedir.

KAYNAKLAR

B. E. Boser, İ. M. Guyon, and V. N. Vapnik(1992). *A training algorithm for optimal margin classifiers*. In D. Haussler, editor, 5th Annual ACM Workshop on COLT. Pittsburgh, PA,. ACM Press 144–152.

Buckley, C., Singhal, A., and Mitra (1996). M. *New retrieval approaches using SMART*. In Proc. Of the 4th Text Retrieval conference (TREC-4), Gaithersburg.

Cavnar, W and Trenkle, J. (1994). *N-gram-based text categorization*. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp.161—175.

Cavnar, W.B., Trenkle, J.M(1994).: *N-gram-based text categorization*. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US 161–175.

Deng,H.; Runger, G.; Tuv, E. (2011). *Bias of importance measures for multi-valued attributes and solutions"*. Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN).

Djoerd Hiemstra and Arjen P. De Vries (2000), *Relating the new language models of information retrieval to the traditional retrieval models*, Published as CTIT technical report TR-CTIT-00-09, May, http://www.ctit.utwente.nl1.

Domingos, Pedro & Michael Pazzani (1997) .*On The Optimality Of The Simple Bayesian Classifier Under Zero-One Loss*. Machine Learning, 29:103–137.

Duda, R.,; Hart, P., (1973) . Pattern Classification and Scene Analysis, Wiley, New York.

Dunn, J.C., (1974). *A Fuzyy Relative of ISODATA Process and Its Use in Detecting Compact*, Well Separated Clusters, Journ., Cybern., 3, 95-104.

Dunning, T. (1994). *Statistical identification of language*. Technical Report CRL Technical Memo MCCS-94-273, University of New Mexico ESANN.99, Belgium.

EuropeanCorpusInitiativeMultilingualCorpusI(ECI/MCI)(2005),http://www.elsnet.org/resources/ecicorpus.html, Page last modified 29-03-2005.

Fung, G. Ve Mangasarian, O. L., (2002), *Incremental Support Vector Machine Classification* Second SIAM International Conference on Data Mining.

George K. Zipf (1949).*Human Behavior and the Principle of Least Effort*. Addison-Wesley.

Grefenstette, G. (1995). *Comparing two language identification schemes*. In: Proceedings of JADT 1995, 3rd International Conference on Statistical Analysis of Textual Data.

Höppner, F.;Klawonn, F.;Rudolf, K.; Runkler, T.(1999). *Fuzzy Cluster Analysis*, Wiley.

J. C. Bezdek (1981): Pattern Recognition with Fuzzy Objective Function Algoritms, Plenum Press, New York.

Joachims, T., *Text categorization with support vector machines: learning with many relevant features.* Proceedings of ECML-98, 10th European Conference on Machine Learning, eds. C. N´ dellec & C. Rouveirol, Springer Verlag, Heidelberg, DE: Chemnitz, DE, pp. 137–142, 1998. Published in the Lecture Notes in Computer

Science series, number 1398.

Joel W. Reed, Yu Jiao, Thomas E. Potok, Brian A. Klump, Mark T. Elmore, and Ali R. Hurson, TF-ICF*: A New Term Weighting Scheme for Clustering Dynamic Data Streams*

Jones, K.S. and Willett, P.(1997). *Readings in Information Retrieval*, Chap. 3. Morgan Kaufmann Publishers, San Francisco, CA,305-312.

MacQueen, J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations"*. **1**. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. Pp. 281–297. MR0214227

Manning, C. And H. Schutze (1999*). Foundations of Statistical Natural Language Processing*. MIT Press, Boston.

Müller K. R., Mika S., Ratsch G., Tsuda K., Schölkopf B., (2001), *An Introduction to Kernel Based Learning Algorithms*.IEEE Trans. On Neural Networks.12:2.

Patricia Newman. (1987). *Foreign language identification: First step in the translation process.* In Proceedings of the 28th Annual Conference of the American Translators Association, pages 509–516.

S.E. Robertson and K. Sparck Jones (1976). *Relevance weighting of search terms*. Journal of the American Society for Information Science, 27:129–146.

Salton, G. and Mcgill (1983), M. *Introduction to Modern Information Retrieval*. Mcgraw Hill.

Satish L, Gururaj BI (April 2003). *Use of hidden Markov models for partial discharge pattern classification*. IEEE Transactions on Dielectrics and Electrical Insulation.

Seising, Rudolf (2007). *The Fuzzification of Systems: The Genesis of Fuzzy Set Theory and Its Initial Applications* - Developments Up to the 1970s (İngiliz dilinde), 32, Springer.

Shannon, C.E. (1948). *A mathematical theory of communication*. Bell System Technical Journal 27, 379–423, 623–656.

Suykens J.A.K., Vandewalle J.,(1999). *Least squares support vector machine classifiers*, Neural Processing Letters, vol. 9, no. pp. 293–300.

Takçı, H., Soğukpınar, İ.,(2004) . *Centroid-Based Language Identification Using Letter Feature Set*, Lecture Notes in Computer Science, (CICLING 2004) Springer-Verlag, Vol. 2945/2004, pages 635-645.

Tanagra Tool- http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html.

Vapnik, V. N., *An overview of statistical learning theory*, IEEE Transactions on Neural Network.

Verayuth Lertnattee and Thanaruk Theeramunkong, *Class normalization in centroid-based text categorization.*

Weka Tool- http://www.cs.waikato.ac.nz/ml/weka/ Erişim Tarihi: 12 Mart 2011.

Weston, J., Watkins, C., 1999), *Support vector machines for multiclass*, Proceedings of ESAAN:99, Belgium.

Xerox Language Identification System (2005), http://www.languageidentifier.com/

Y. Yuan and M.J. Shaw(1995*). Induction of fuzzy decision trees*. Fuzzy Sets and Systems .69, pp. 125–139.

Zhang, H., (2005). *A Note Fuzzy Clustering*, Department of Computer Science and Engineering University Connecticut.

Z. Macnamara, P.Cunningham, J. Byrne(1998). *Neural Network for Language Identification* A comparative study.Information Processing and Management, Vol.34, No.4,pp.395-403.

ÖZGEÇMİŞ

1986 yılında İstanbul/Fatih'de doğdu. 2004 yılında Küçükçekmece (YDA)Lisesi'nden mezun oldu. 2004 yılında Haliç Üniversitesi Mühendislik Fakültesi'nde eğitime başlayıp 2009 yılında Bilgisayar Mühendisi olarak mezun oldu. 2009 yılında Haliç Üniversitesi Bilgisayar Mühendisliği Bölümü'nde yüksek lisansa başlayıp eğitimi devam etmektedir. Aynı zamanda 2009 yılında, Araştırma Görevlisi olarak göreve başladığı Haliç Üniversitesi Bilgisayar Mühendisliği Bölümü'nde görevine hala devam etmektedir.