T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

İNGİLİZ ALFABESİ KULLANILARAK YAZILMIŞ TÜRKÇE METİNLERİN TÜRK ALFABESİNE GÖRE YENİDEN OLUŞTURULMASI

BURAK ÇAĞRI OKUR
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2013

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

İNGİLİZ ALFABESİ KULLANILARAK YAZILMIŞ TÜRKÇE METİNLERİN TÜRK ALFABESİNE GÖRE YENİDEN OLUŞTURULMASI

BURAK ÇAĞRI OKUR
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DOÇ. DR. YUSUF SİNAN AKGÜL

GEBZE
2013

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ	YÜKSEK LİSANS JÜRİ ONAY FORMU

GYTE Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 24/06/2013 tarih ve 2013/34 sayılı kararıyla oluşturulan jüri tarafından 19/09/2013 tarihinde tez savunma sınavı yapılan Burak Çağrı OKUR'un tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE
(TEZ DANIŞMANI) : Doç. Dr. Yusuf Sinan AKGÜL

ÜYE : Doç. Dr. Fatih Erdoğan SEVİLGEN

ÜYE : Yrd. Doç. Dr. Hidayet TAKÇI

ONAY

GYTE Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ...../..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

İngilizce alfabesi ile yazılan Türkçe metinler her ne kadar insanlar tarafından kolay anlaşılsa da, bu işlemin otomatik olarak yapılması günümüzde hala tam çözülmemiş Sözcük Anlamı Belirleme (Word Sense Disambiguation) problemlerinden birisi olarak karşımıza çıkmaktadır. İngilizce alfabesi ile yazılmış olan metinlerin Türkçe alfabesi ile yeniden yazılması, Türkçe'ye özgü bir Doğal Dil İşleme çalışmasıdır. Farklı Türkçe kelime seçenekleri içinden, uygun olanın bulunması için metnin anlamsal açıdan ele alınması gerekmektedir. Bu çalışmada, metnin cümle bazlı veya tüm parça olarak incelenmesinin doğru kelime tercihi üzerindeki etkileri araştırılmıştır. İstatistiğe dayalı yöntemler ile makina öğrenmesi yöntemlerinin doğru kelime tercihi üzerindeki başarısı incelenmiştir. Bir metnin tüm parça olarak incelenmesinin, bize metin hakkında cümle bazlı yönteme göre daha fazla bilgi verdiği; ayrıca makina öğrenmesi yöntemlerinin, istatistiksel bazlı yapılan çalışmalara göre daha iyi sonuçlar sağladığı deneylerle gösterilmiştir.

Anahtar Kelimeler: Doğal Dil İşleme, Metin Madenciliği, Sözcük Anlam Belirleme, Makina Öğrenmesi.

SUMMARY

Turkish texts written by English characters are easily comprehended by people, although performing this process by machines is still one of the unsolved Word Sense Disambiguation problems. Rewriting texts in English characters using Turkish characters is a natural language processing problem special to Turkish. Choosing the right Turkish word among different alternatives requires consideration of the text semantically. In this study, the effect of examination of the text either sentence or whole text based, on the right word determination is investigated. Performance of machine learning methods and statistical methods in right word determination is examined. The study is tested on randomly selected news texts. It is shown that examination of the text as a whole provides more information compared to sentence based methods and machine learning methods provides better results compared to statistical studies.

Keywords: Natural Language Processing, Text Mining, Word Sense Disambiguation, Machine Learning.

TEŞEKKÜR

Bu tezin bütün süreçlerinde yol gösterici olan Sayın Doç. Dr. Yusuf Sinan AKGÜL hocama,

Bana her türlü olanağı sağlayıp, bilgi ve tecrübelerini benimle paylaşarak çalışmalarıma katkı sağlayan iş arkadaşlarıma,

Ve göstermiş oldukları her türlü destekten dolayı aileme ve eşim Bahriye OKUR'a en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler ve Kısaltmalar		**Açıklamalar**
ARFF	:	Attribute-Relation File Format
BOSDD	:	Belirlenimci Olmayan Sonlu Durum Dönüştürücüsü
BSDD	:	Belirlenimci Sonlu Durum Dönüştürücüsü
DDİ	:	Doğal Dil İşleme
GYTE	:	Gebze Yüksek Teknoloji Enstitüsü
POS	:	Part Of Speech
SAB	:	Sözcük Anlamı Belirleme
SDD	:	Sonlu Durum Dönüştürücüsü
WSD	:	Word Sense Disambiguation

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

1. GİRİŞ

İngilizce alfabesi ile yazılmış Türkçe metinlerin Türkçe alfabesi ile anlam karmaşıklığına neden olmadan tekrar oluşturulması, insanlar tarafından bir problem olarak algılanmamaktadır. Ancak bu işlemin otomatik olarak bilgisayarlar aracılığıyla yapılması oldukça problemlidir. Türkçe alfabesinde bulunan bazı harflerin (ç, ş, ğ, ü, ö, ı), İngilizce klavye düzeninde bulunmaması nedeniyle anlam karmaşıklıkları ortaya çıkmaktadır. Örneğin, İngilizce alfabesi ile yazılmış olan "Avrupa'dan odun aldi." cümlesi, Türkçe karakterler ile ifade edildiği zaman "Avrupa'dan odun aldı." veya "Avrupa'dan ödün aldı." cümlelerine denk gelebilmektedir.

Türkçe klavyeler ancak 2000'li yıllarda sıklıkla kullanılmaya başlandığı için, bu tarihten önceki Türkçe metinler İngilizce klavyeler kullanılarak yazılmıştır. Bu nedenle, sayısal dünyada İngilizce alfabesi kullanılarak yazılmış Türkçe metin oldukça fazladır. Ayrıca, bu tarihten önce bilgisayar kullanmaya başlamış olan kişiler de, alışkanlıklarını devam ettirerek Türkçe klavye üzerinde ancak İngilizce alfabesini kullanarak yazı yazmaya devam etmişlerdir. Son olarak, Türkçe klavyeye erişimi bulunmayan kullanıcılar da, bu tür metinlerden üretmektedirler. Yukarıda sayılan sebeplerden dolayı ortaya çıkan İngilizce alfabe ile yazılmış metinlerin düzeltilmesi hem metinlerin anlaşılabilmesi ve oluşabilecek muğlaklıkların giderilmesi, hem de Türkçe'nin doğru kullanılması açısından önemlidir.

Problemi matematiksel olarak tanımlamak gerekirse;

$$T = \{a, b, c, ç, \dots, z, A, B, \dots, X, Y, Z\} \tag{1.1}$$

Türkçe alfabesi harflerini içeren küme olsun.

$$E = \{a, b, c, , \dots, x, y, z, A, B, \dots, X, Y, Z\} \tag{1.2}$$

İngilizce alfabesi harflerini içeren küme olsun.

$$f = T \to E \tag{1.3}$$

f, T kümesinden E kümesine bir N-1 fonksiyon olsun. Bu fonksiyon, kartezyen çarpımı şeklinde ifade edilebilir.

$$f = \{(a,a),(b,b),(c,c),(ç,c),\dots,(s,s),(ş,s),\dots\} \quad (1.4)$$

Türkçe alfabesi kullanılarak yazılmış bir metin, kolay bir şekilde sadece İngilizce alfabesini kullanan bir metne fonksiyonu ile dönüştürülebilir.

$$g = E \rightarrow T \quad (1.5)$$

g, E kümesinden T kümesine bir 1-N bağıntı olsun. Bu bağıntı, kartezyen çarpımı şeklinde ifade edilebilir.

$$g = \{(a,a),(b,b),(c,c),(c,ç),\dots,(s,s),(s,ş),\dots\} \quad (1.6)$$

Dikkat edilirse, g bağıntısı bir fonksiyon değildir. Çünkü aynı anda hem $g(c) = c$, hem de $g(c) =$ ç sağlanmak zorundadır. Bu nedenle İngilizce alfabesini kullanarak yazılmış bir Türkçe metnin Türkçe alfabesi ile otomatik olarak çevrilmesi kolay bir şekilde yapılamaz. Bu problemin çözülebilmesi için çok üst seviyede anlam ve bağlam bilgisi kullanmak gerekmektedir.

Bu konuda daha önceden yapılmış farklı çalışmalar (Asciifier & Deasciifier) mevcuttur [1]-[3]. Bu çalışmalar temel olarak, metni incelerken sentaks açısından konuya yaklaşmışlar, fakat anlam ve bağlam açısından konuyu ele almamışlardır. Bu şekilde yapılan metin düzenlemeleri, parçanın anlam bütünlüğünün bozulmasına neden olmaktadır. Çalışmamızda, istatistik ve makina öğrenmesi yöntemlerini devreye sokarak, anlam ve bağlam açısından konuya yaklaşmanın daha başarılı sonuçlar verdiği gözlemlenmiştir.

Bu çalışmamızda, bahsedilen problemin çözümü için Sözcük Anlamı Belirleme (Word Sense Disambiguation) tekniklerini kullanmayı önermekteyiz. SAB teknikleri genel olarak cümle içinde geçen farklı anlamlara gelebilecek kelimelerin, hangi anlama denk geldiğini bulmak için kullanılmaktadır. İngilizce alfabesi kullanılarak yazılmış metnin Türkçe alfabesi kullanan şekle getirilmesi sırasında ortaya özgün bir

SAB problem çıkmaktadır ve bu problemin çözümü için bilinen SAB tekniklerinde faydalanılacaktır.

Bu çalışmada, bahsedilen problemin çözümünde, istatistiksel yöntemler (Ngram'lar), metin madenciliği uygulamaları (kelime ön işleme, kök bulma, gürültü ayıklama...) ve makina öğrenmesi algoritmaları (AdaBoost, J48 tree, Naive Bayes Algoritmaları) kullanılmış; bunların başarım dereceleri incelenmiştir. İncelenen metinlerdeki kelimeler metin madenciliği metotları ile, muğlaklık oluşturan veya oluşturmayan kelimeler olarak ikiye ayrılmıştır. Sonraki aşamada muğlaklık oluşturan kelimelere ait farklı Türkçe kelime seçenekleri içinden uygun olan seçilmiştir.

Metin ön işleme çalışmalarında ön hazırlık olarak sözlük veya veri tabanı (corpus/lexicon) hazırlanmıştır. Bu sözlük ile SAB problemi oluşturabilecek kelimeler (odun, ödün, vb) belirlenmiş, bu kelimelerden bir veri seti oluşturulmuştur. Yapılan çalışmada yeterli sayıda kelime içeren bir sözlük hazırlanmasının, arama sonuçlarının doğru ve hızlı sonuçlar vermesine doğrudan etki etki ettiği görülmüştür. Hazırlanan sözlük, hem N-Gram'larda kelimelerin geçme sıklıklarının hesaplanmasında, hem de makina öğrenmesi algoritmalarında eğitim seti oluşturulmasında kullanılmıştır.

Ele alınan problem, SAB probleminin bir örneği olmasına rağmen, bu problem sadece Türkçe'ye ve Türkçe kullanan bilişim dünyasına özgü bir problem olup, Doğal Dil İşleme yöntemleri kullanılarak çözülmesi bu konudaki literatüre önemli katkıda bulunacaktır.

1.1. Türkçe Dili

Türkçe Ural-Altay dil grubuna giren bir dildir. Türkçe, diğer Türk dilleriyle birlikte Altay dil ailesinin bir kolunu oluşturur. Bu ailenin diğer üyeleri Moğolca, Mançu-Tunguzca ve Korecedir.

DDİ çalışmalarında Türkçe dili kendisine has yapısal özelliklerinden dolayı bazı problemler ile karşılaşmaktadır. Öncelikle sözcük yapısı ve yeni sözcük oluşturulması açısından Türkçe sondan eklemeli bir dildir. Bu açıdan Türkçea ynı dil ailesi içinde olmamasına rağmen Fince ve Macarca ile benzerlik göstermektedir. Bu tip dillerde sözcükler bir kök ve o sözcüğe sanki tespih taneleri gibi eklenen (ancak

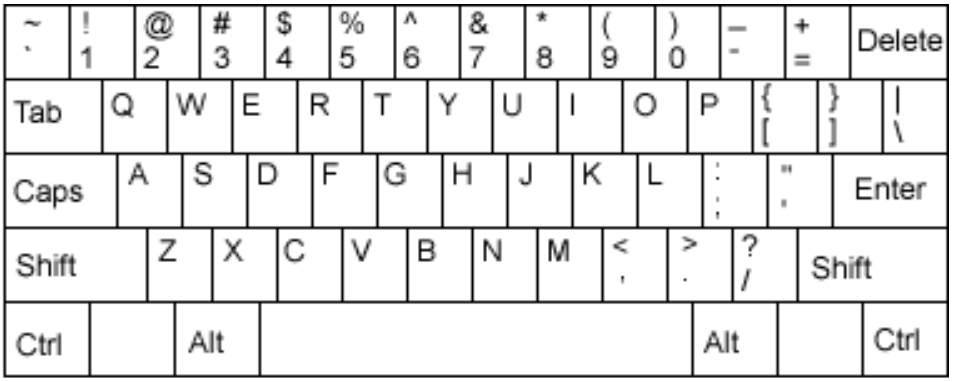
eklenirken, ünlü uyumu, ünsüz değişmesi, ünlü ve ünsüz düşmesi gibi nedenlerle değişikliğe uğrayan) biçimbirimlerden oluşurlar.

Türkçe, diğer Altay dilleri gibi eklemeli, yani sözcüklerin eklerle yapıldığı ve çekildiği, sondan eklemeli bir dildir. Bu nedenle istenildiği kadar sözcük türetilmesine müsade eden bir dildir. Büyük Ünlü Uyumu (kalınlık-incelik) ve Küçük Ünlü Uyumu (düzlük-yuvarlaklık) olarak bilinen , bir kelimenin Türkçe olup olmadığını anlamamıza yardımcı olan kuralları vardır.

Biçimbirim, kelimelere dil bilgisi bakımından biçim veren çoğu ek hâlinde olan kelime parçaları, biçim birimidir (Yapım Ekleri, Çekim Ekleri). Bu biçim birimler eklendikleri kök veya gövdenin anlamını, sözcük türünü veya sözdizimsel işlevini değiştirebilirler.

1.2. Türkçe ve İngilizce Klavye

Türkçe ve İngilizce dilleri için kullanılan klavyeler Şekil 1.1 ve Şekil 1.2'de gösterilmiştir.



Şekil 1.1: İngilizce QWERTY klavye düzeni.

Şekil 1.2: Türkçe QWERTY klavye düzeni.

Türkçe Klavye'de bulunan ancak İngilizce Klavye'de bulunmayan karakterler ve karşılıkları Tablo 1.1.a) ve Tablo 1.1.b)'de verilmiştir.

Tablo 1.1: a) Küçük harfler için klavye karakter karşılığı, b) Büyük harfler için klavye karakter karşılığı.

Türkçe Karakter		İngilizce Karakter
ç	→	c
ı	→	i
ö	→	o
ş	→	s
ğ	→	g
ü	→	u

a)

Türkçe Karakter		İngilizce Karakter
Ç	→	C
İ	→	I
Ö	→	O
Ş	→	S
Ğ	→	G
Ü	→	U

b)

2. DOĞAL DİL İŞLEME

Doğal Dil İşleme (Natural Languase Processing), Yapay Zeka (Artificial Intelligence) ve Dil Bilim (Linguistics)'in bir alt dalıdır. Türkçe, İngilizce, Almanca, Fransızca gibi doğal dillerin (insana özgü tüm dillerin) işlenmesi ve kullanılması amacı ile araştırma yapan bilim dalıdır.

Doğal dillerin bilgisayar tarafından üretimi ve anlaşılması problemleriyle ilgilenir. Doğal dil üretim sistemleri bilgisayar veri tabanındaki bilgileri doğal dile; doğal dil anlama sistemleri de insan dili örneklerini bilgisayar programlarının işlemesi kolay olan soyut betimlemelere çevirir. Ancak bu işlem doğal dile ait bazı belirsizlikler içerdiğinden ve doğal dilin kendisine has özelliklerinden dolayı zorluklar içermektedir.

Doğal Dil İşleme, doğal dillerin kurallı yapısının çözümlenerek anlaşılması veya yeniden üretilmesi amacını taşımaktadır. Bu çalışmaların bizlere sağlayacağı kolaylıklar olarak,

- yazılı dokümanların otomatik çevrilmesi,
- soru-cevap makineleri,
- otomatik konuşma ve komut anlama,
- konuşma sentezi,
- konuşma üretme,
- otomatik metin özetleme,
- bilgi sağlama ...

birçok başlık sayılabilir.

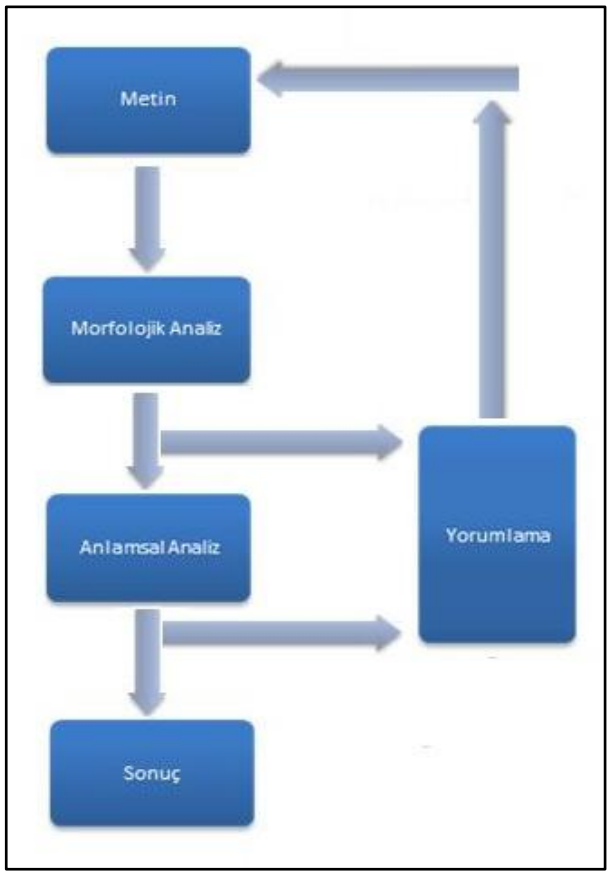
Türkçe'de Doğal Dil İşleme ile alakalı çalışmalarda bulunan Dr. Kemal Oflazer, 1997 yılındaki makalesinde [3], [4] DDİ alanında yapılan temel çalışmaları aşağıdaki şekilde gruplamıştır:

- Doğal dillerin işlevi ve yapısını daha iyi anlamak,
- Bilgisayarlar ile insanlar arasındaki arabirim olarak doğal dil kullanmak ve bu şekilde bilgisayar ile insanlar arasındaki iletişimi kolaylaştırmak,

- Bilgisayar ile dil çevirisi yapmak,
- Genellikle İngilizce ve benzeri diller temel uygulama alanı alındığı için, bu dillerden farklı dillerdeki uygulanmalarda sorunlar çıkmaktadır.

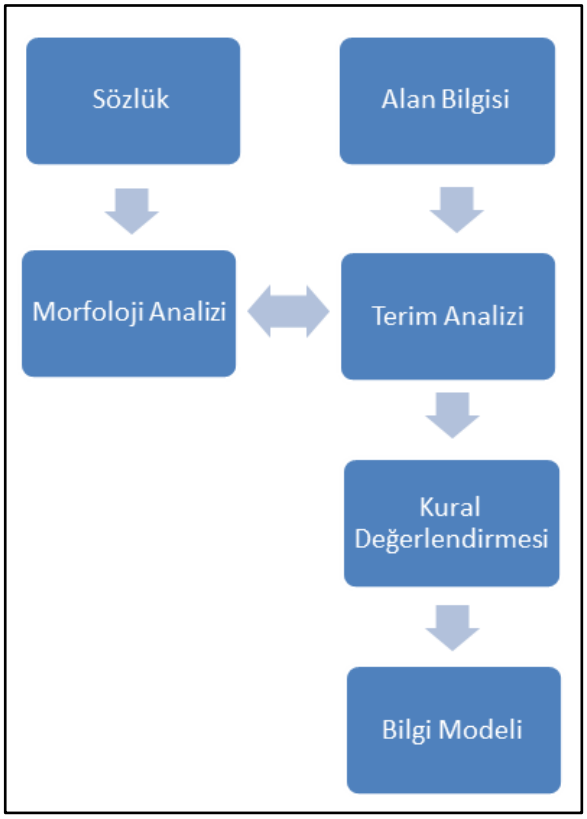
Dil yeteneği, insana özgü bir özellik olduğu için dilbilim Bilişsel Bilimler içerisinde önemli bir yer tutmaktadır. DDİ, doğal dillerin bilgisayar tarafından üretilmesi ve anlaşılması problemleriyle ilgilenmektedir. Eğer dilin bilgisayar ortamında bir modeli oluşturulabilirse, insan-makina iletişiminde için oldukça yararlı bir araç elde edilebilecektir.

Şekil 2.1'de metinden sonuç çıkarma sistemine ait bileşenler gösterilmiştir. Bir DDİ probleminin çözülebilmesi için, ele alınan problemin bilgisayar tarafından anlaşılması gerekmektedir. Bunun için cümlelerin morfolojik analizinin gerçekleştirmesi ve bu analizden elde edilen bilgilerle bir veri tabanı oluşturulmalıdır. Oluşturulan veri tabanı yardımıyla anlaşılan problem matematiksel formüllerle ifade edilerek, sonuca ulaşılmaktadır [5].



Şekil 2.1: Türkçe için bilgisayar anlaması sistemi.

DDİ farklı bilim alanları ile birlikte kullanılabilecek, faydalı bilgisayarlı otomasyonlar yapılabilecek çalışma alanları sunmaktadır. Daha önceden yapılmış olan bir çalışmada Türkçe yazılmış radyoloji raporlarının yapılandırılmış bilgi modeline dönüştüren bir sistem oluşturulmuştur. Bu sistem DDİ teknik bilgisini ontoloji tıp bilimi ile birleştirerek, raporlardan bilgi çıkarımı üzerine çalışmalarda bulunmuştur [6]. Her ne kadar çalışma vücudun sadece belli bir bölgesine ait çalışmaları kapsamış olsa da diğer organları da kapsayacak şekilde geliştirilebileceği belirtilmiştir. Yapılmış olan sistem Şekil 2.2'de gösterilen ana bileşenleri içermektedir. Raporlardan alınan bütün cümleler morfolojik analizden geçirilmiştir. Morfolojik analiz kullanılmadığı zaman önemli bilgi kayıpları ile karşılaşılmıştır. Bu morfolojik analiz sırasında alan bilgisine ait terimler (bu çalışmada Ontoloji bilimine ait terimler) anlamlarından faydalanılmış, kural değerlendirmeleri sonucunda bir bilgi çıkarımı sağlanmıştır. Şekilde gösterilen kural tabanı eğitim seti olarak kullanılan raporlar üzerinde alan uzman kişiler tarafından oluşturulmuştur. Sistemin bu kural tabanına göre test aşamasında gelen bir USG raporundan bilgi çıkarımı elde etmesi sağlanmıştır.



Şekil 2.2: Türkçe için bilgi çıkarma sistemi ana bileşenleri.

Dökümalardan yazar tahminin yapılması DDİ alanında yapılmış olan çalışmalara farklı bir örnek olarak verilebilir. Daha önceden bu konuda yapılmış çalışmada [7] yazar tahmini için;

- N-Gram'lar,
- yazarlık özellikleri olarak tanımlanan yazarın üslubunu oluşturacak yazım şekillerinden,
- makina öğrenmesi algoritmalarından

faydalanılarak bir sistem oluşturulmuştur.

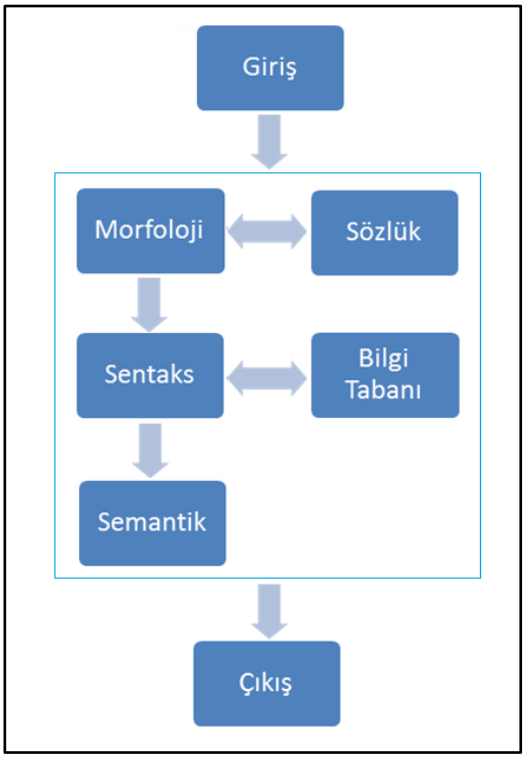
Sözlük oluşturmak için farklı yazarların çeşitli metinleri toplanmış, bu metinler üzerinde ön işlemler yapılarak yukarıda söylenen 3 farklı yöntem için sistemin eğitilmesi sağlanmıştır. Sistemin test edilmesi sırasında farklı öznitelik sayıları ile deneyler yapılmıştır [8]. Eğitilen sisteme yazarı bilinmeyen bir metin verildiğinde, öznitelik sayısı azaltılmamış vektörlerde yapılan çalışmaların doğruluk oranının(%65.8), öznitelik sayısı azaltılmış vektörlerle yapılan çalışmalara(%75.2) göre daha az olduğu gözlenmiştir. Ayrıca 3 yöntemin birlikte kullanıldığı çözümde metnin yazarının %92'lere varan doğruluk oranı ile tahmin edilebildiği görülmüştür. Böylelikle yüzyıllar öncesinden günümüze gelen ve anonim olarak bildiğimiz metinlerin yazar tahminlerinin DDİ yöntemleri ile başarılı bir şekilde yapılabildiği görülmüştür.

Yukarıda anlatılan yağılmış çalışmalardan yola çıkarak Doğal Dil İşleme projelerinde çalışırken incelenmesi gereken anahtar alanları 4 ana bölüme ayırabiliriz. (Şekil 2.3)

- Morfoloji (kelime formlarının bir grameri, kelime kökeni)
- Sentaks (cümleler oluşturmak için kullanılan kelime bileşimlerinin grameri)
- Semantik (kelime veya cümlenin anlamı)
- Sözlük (Lexicon/Corpora veya kelimeler kümesi)

Doğal Dil İşleme projelerinde çözülmesi gereken problemler iki ana gruba ayrılmıştır.

- Kelime ve fiillerin muğlaklığı
- Anlamın muğlaklığı



Şekil 2.3: Doğal dil işleme çalışma alanları.

2.1. Morfoloji

Kelimelerin yapısının çözümlemesini, tanımlanmasını ve kimliklendirilmesini inceleyen bilim dalıdır. Kelimeler genellikle en küçük yazım birimi olarak kabul edilmelerine rağmen, çoğu dilde kelimeler diğer kelimelere bazı kurallar ile bağlı olabilmektedir.

Kelimelere eklenen ekler ile yeni kelimeler oluşturulabilmektedir. Örnek: göz, gözlük, gözlükçü, gözlükçülük

Morfoloji dilbiliminin dillerdeki bu yapısal kuralların ve örnek yapıların anlaşılması için çalışan koludur. Örnek: Evdeymişler – Tek kelimelik bir cümle!

Bir kelimede yer alan eklerin ve türlerinin belirlenmesini sağlamaktadır. Örnek: Ev-ler-de-ki (Kök +çoğul+bulunma hali+aitlik)

Zengin bir türetme yapısına sahiptir. Örnek: Evlerdekilerinkiydi.

Bazı kelimeler kendilerine has kurallara sahip olabilmektedir. Örnek: Son harfin tekrarı: üs → üssü, hak → hakkı; Sesli düşmesi : burun → burnu, alın → alnı

2.2. Sözdizimsel Analiz

Sözdizimini (syntax) veya cümleyi oluşturan morfolojik öğelerin hiyerarşik kurallara uyumunu karşılaştırarak ölçümlemektir. Böylelikle söz dizimin anlamlı olup olmadığının ölçülebilmesi için düzenleyici bir süreç gerçekleştirilmesi sağlanır.

Türkçede cümleler en genel şekliyle özne, nesne ve yüklem bileşenlerinden oluşur. Cümleye eklenmek istenen anlamlar arttıkça cümleler, özne, yer tamlayıcısı, zarf tamlayıcısı, nesne ve yüklem gibi bileşenleri içerir. Ayrıca cümlenin anlamını kuvvetlendiren cümle dışı bileşenler de (bağlaç, edat, vb) cümlede bulunabilir.

Bunlara örnek olarak "ile, için, ama, çünkü" kelimeleri verilebilir. Türkçede özne ile yüklem cümlenin temel bileşenleridir ve genelde tüm cümlelerde yer alırlar. Yer tamlayıcısı, zarf tamlayıcısı, nesne gibi bileşenler bazı cümlelerde yer almayabilirler veya bazı cümlelerde sadece biri, bazılarında sadece ikisi bulunabilir. Bu bileşenlerin cümle içindeki sıralanışları da değişebilir.

Bilgisayarla doğal dilin modellenmesinde anlamsal analizden önce kelimelerden oluşturulan yapının cümle olup olmadığının test edilmesi faydalıdır. Bu işlem sentaktik eşleştirme işleminde anlamsız eşleşmelerin önlenmesine faydalı olur.

Simgeler: Ö: özne, D: dolaylı tümleç, Z: zarf tümleci, N: nesne, Y: yüklem, İG: isim grubu, SG: sıfat grubu, İN: isim nesnesi, SN: sıfat nesnesi, DZ: diğer zarflar, S: sıfat, İ: isim, ZB: zaman belirteçleri, T: tamlayan, TN: tamlanan, ZM: zamir, NE: nesne eki, TE: tamlayan eki, TNE: tamlanan eki, KE: kip eki, ZE: zaman eki, DE: dolaylı tümleç eki, EF: ek fiil

2.3. Anlambilimel Analiz

Sözdizimini oluşturan morfolojik öğelerin ayrılması yani, sözdizimsel analiz ile anlam taşıyan kelimelerin sınıflandırılması işleminden sonra gelen anlamlandırma sürecidir. Anlam taşıyan kelimelerin, ekler ve cümle hiyerarşisi içindeki konumlarının saptanması sayesinde birbirleri ile ilişkilerinin ortaya konulmasıdır. Bu

ilişkiler anlam çıkarma, fikir yürütme gibi ileri seviye bilişsel fonksiyonların oluşturulmasında kullanılmaktadır. Farklı alt alanları vardır.

- Sözcüklerin ve morfemlerin anlamları,
- Sentaktik birimlerin anlamlarının, tek tek kelimelerin anlamlarından nasıl oluştuğunun araştırılması,
- Cümlelerin yorumu bu cümlenin sentaktik yapısının çözümlemesinin araştırılması,
- Gerçek ya da varsayımsal olarak cümlelerin birleşiminin çözümlenmesi,
- Birbiriyle ilişkisi olan farklı kişilerin metinlerdeki düzeyi üzerine çalışmalar yapar.

2.4. Sözlük

Bir dilde kullanılan kelime ve deyimleri alfabe sırasına göre tanımlayarak açıklayan dizime sözlük denilir. Üzerinde çalıştığımız metin içerisinde geçen kelimelerden oluşturulan kelime dizisine de sözlük adı verilmektedir. Bu kelime dizileri bize kelime çiftlerinin birlikte kullanılma durumlarını, sıklıklarını, aldığı veya alabileceği ekler hakkında yardımcı olmaktadır. Sözlükte bulunan kelimeler gürültü ve eklere sahip (üzerinde herhangi bir kelime ön işlemi yapılmamış kelimeler) olabilmektedir. DDİ çalışmalarında sözlük hazırlanması büyük önem teşkil etmektedir. Sözlükten elde edilen bilgi problem çözümünü sağlamaktadır [9].

İngilizce dili üzerinde yapılmış ve dünyada DDİ üzerinde kullanılmak üzere hazırlanan ilk proje WordNet'tir [10]. WordNet ücretsiz olarak kullanılabilen bir kütüphanedir. Princeton Üniversitesi Bilişsel Bilimler Laboratuarı'nda 1985 yılında Prof. A.G. Miller tarafından başlatılan bir çalışmadır. Bu çalışmada İngilizce dilinde bulunan kelimelere ait aşağıdaki bilgilere ulaşılabilir.

- İngilizce dilinde kullanılan kelimeler
- Kelimelerin anlamları
- Kelimelerin eşanlamlıları
- Eşanlamlı kelimeler arasındaki anlamsal ilişkiler
- Özne, tümleç, yüklem gibi hangi cümle öğesini temsil ettiği

WordNet projesine ait istatistik bilgileri Tablo 2.1'de verilmiştir.

Tablo 2.1: WordNet istatistikleri.

Öğeler	Kelime Sayısı	Eşanlamlı Kelime Sayısı	Toplam Kelime Çifti
İsim	117.798	82.115	146.312
Yüklem	11.529	13.767	25.047
Sıfat	21.479	18.156	30.002
Zarf	4.481	3.621	5.580
Toplam	155.287	117.659	206.941

WordNet'te yapılan çalışmanın 6 Balkan ülkesinin dillerini (Bulgarca, Çekce, Yunanca, Romence, Sırpça ve Türkçe) kapsayacak şekilde oluşturulması amacıyla BalkaNet projesi oluşturulmuştur [11]. Bu projenin bir parçası kapsamında Türkçe dili için kavramsal sözlük oluşturulmasına çalışılmıştır [12]. Bu projede otomatik olarak Türkçe dilbilgisi sözlüğünden bir kelimeye ait,

- eş anlamlı kelimeleri
- zıt anlamlı kelimeleri
- keliemelerin çoklu anlamlarını,
- alt ilişkili kelimeleri...

çıkarmak amacıyla çalışmalar yapılmıştır.

3. SÖZCÜK ANLAMI BELİRLEME

30 sene önce bilgisayar kullanımı ve bilgi teknolojileri toplumda bu kadar çok kabul görmemiş bir konuydu. İnsanlar kağıt-kalem ile bir çok işlerini halledebildiği için bilgisayarlardan faydalanmayı elzem bir ihtiyaç olarak görmemekteydi. Günümüzde artan internet ve bilgi kullanımı sonucunda, sadece insan gücü ile işlerin altından kalkılamayacağı anlaşılmıştır. Bilgiayarlardan daha fazla faydalanma gereksinimi, bilgisayarların insanların isteklerini anlamasının istenmesine neden olmuştur. Bilgisayarların yazılan veya konuşulan bir cümleyi doğru anlaması SAB probleminin çıkış kaynağını oluşturmuştur [13], [14].

İnsanlar konuşma dilinde bir kelimeyi farklı cümleler içinde değişik manalara gelecek şekilde kullanabilirler. Kelime yazılışı aynı olduğu halde cümle içinde kullanılışına göre farklı anlamlar ifade etmektedir. Bu konuya açıklık getirmek için aşağıdaki iki cümle üzerinde konuşabiliriz.

- Gözlerinin "kara"sına vuruldum.
- Denizciler "kara"ya çıkmak için günlerdir bekliyorlardı.

Yukarıdaki iki cümlede de kullanılan "kara" kelimesi bir cümlede renk manasında; diğer cümlede toprak parçası manasında kullanılmıştır. Sadece kelimenin kendisine bakarak hangi anlamda kullanıldığını çıkarmak çok güçtür. Bir kelimenin cümle içinde birlikte kullanıldığı diğer kelimeler onun anlamını belirlemektedir. Birinci cümledeki "kara" kelimesini anlamını kazandıran yardımcı kelime "göz" kelimesi; ikinci cümlede "kara" kelimesine anlamını kazandıran yardımcı kelimeler ise "deniz" ve "çıkmak" kelimeleridir.

Bahsettiğimiz kelime anlamı belirlemesini insanlar günlük hayatta sıklıkla yapmakta ve hiçbir zorluk hissetmemektedir. Bilgisayarların bu işlemi yapabilmesi için cümleyi işleyip analiz etmesi, cümledeki kelimelerin anlamlarının belirlenmesi, bu anlamlar ile ortak mana ifade eden kelimenin seçilmesi gerekmektedir.İşte bu şekilde kelime anlamlarının metin bağlamına bakılarak hesaplanması işlemin Sözcük Anlamı Belirleme (Word Sense Disambiguation) denilmektedir.

SAB problemi faklı çalışma alanlarında kullanılmaktdır. Bunlara örnek verecek olursak;

- Makine Çevirisi (Machine Translation),
- Bilgi Geri Kazanımı (Information Retrieval),
- Ses İşleme (Speech Processing),
- İnsan-Makine Etkileşimi (Human-Computer Interaction),
- Metin İşleme (Text Processing),
- İçerik ve Tematik Analiz (Content and Thematic Analysis)
- Gramer Çözümlemesi (Gramatical Analysis).
- Biyoenformatik (Bioinformatics)
- Semantik Web (Semantic Web)...

SAB problemi, Yapay Zeka ve Doğal Dil İşleme'nin çalışma alanlarından birisi olarak tanımlanmıştır. Sözcüklerin anlamsal düzeydeki karmaşıklıklarının çözülmesini sağlamak amacıyla yapılan çalışmaları kapsamaktadır. Yapılan çalışmalar genellikle aynı yazılışa fakat farklı anlamlara sahip olan (çok anlamlı veya eş sesli) kelimelerin içinde geçtiği metin parçasına göre hangi anlamı ifade ettiğinin belirlenmesini amaçlamaktadır [15].

Bu problemin hesaplama analizi yapıldığında NP-complete zorluk sınıfında bir problem olduğu görülmektedir. Yani matematiksel olarak polinom zamanda çözümü bulunamayan bir problemdir.

SAB probleminin çözümü bilgiye dayanmaktadır. Bilgi, cümlelerden veya her türlü yazılı metinlerden elde edilmektedir. Bir kelime anlamının bulunabilmesi için aynı parça içimde geçen diğer kelimelerden faydalanıldığını söylemiştik. Metinler kullanılmadan önde bazı işlemlerdenden geçirilerek,

- içinde geçen kelime anlamlarının belirlenmesi,
- aynı anlama gelen kelimelerin bir arada gruplanması,
- aynı konuya ait kelimelerin bir araya gruplanması,
- kelimelerin cümle içinde kullanılma sırasına göre gruplanması

gibi farklı biçimlerde gruplanması ve bir sözlük oluşturulması gerekmektedir. Hazırlanan sözlük bizim için gerekli olan bilgiyi içermektedir.

Sözlük hazırlanması SAB yöntemlerinin en önemli adımlarından birisini oluşturmaktadır. Sözlük olmadan probleme çözüm getirmek imkansızdır. Sözlüğün

hazırlanması veya önceden hazırlanmış olan bir sözlüğün probleme uygun şekilde tekrar elden geçirilmesi gerekmektedir. Eğer daha önceden farklı bir porblemin çözümünde kullanılan bir sözlükten faydalanılamıyorsa, problemin çözümüne yönelik yeni bir sözlük hazırlanması gerekmektedir. SAB yöntemlerinde kullanılmak için sözlük hazırlanması ayrı bir problem olarak değerlendirilmiş ve insanların faydalanacağı içinde bulunan kelimeler arasında ilişki bulunan önceden yapılmış birçok çalışma mevcuttur [16]-[20].

Hazırlanan sözlük bizim için gerekli bilgiyi içermektedir ancak bu bilginin probleme uygun şekilde kullanılması, bize bir çözüm üretmede yardımcı olması gerekmektedir. Daha önceden yapılmış olan SAB çalışmalarında genellikle derlem tabanlı ve makina öğrenmesi tabanlı olmak üzere iki ana yaklaşım izlenmektedir. Birinci yöntem genellikle istatistiksel tabanlı yaklaşımları içermektedir. İkinci yöntem ise makina öğrenmesi algoritmaları yardımıyla çıkarım yapılması esasına dayanmaktadır. Ayrıca bu iki yöntemin birlikte kullanıldığı karma yaklaşımlar da mevcuttur.

SAB problemlerine çözüm getiren yöntemler sözcüklerin kullanıldıkları yere ait bilgilerden faydalanır. Farklı uygulama yöntemleri olmakla birlikte, çoğu çalışmada kelimenin kullanıldığı yerdeki diğer sözcüklerle ilişkisi, hedef sözcüğe olan uzaklığı ve dilbilgisi gibi faktörler dikkate alınmaktadır. Bazı çalışmalarda ise kelimenin anlam ve bağlam açısından incelenmesi ve birlikte kullanıldığı diğer kelimeler ile semantik ilişkisi göz önüne alınmaktadır. Örneğin, bir dildeki metnin başka bir dile aktarılması probleminde, her bir kelime için cümle içindeki diğer kelimelerle olan ilişkisi puanlanmaktadır. Benzer puanlama diğer dildeki çeviri için de yapılarak, çevirinin doğruluğu hesaplanmasında da kullanılmaktadır [21].

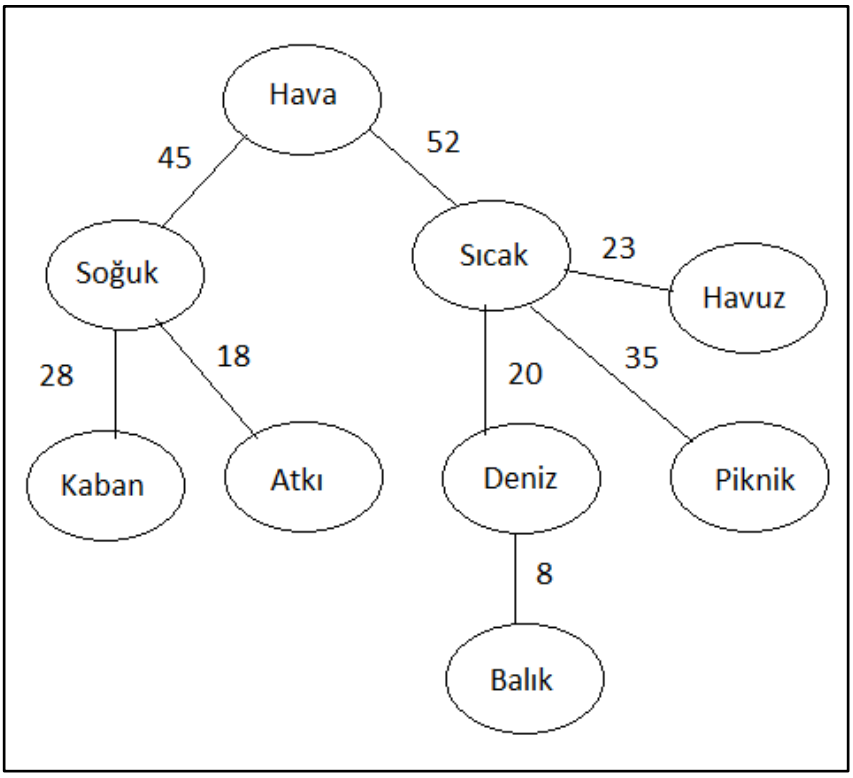
Daha önceden SAB ile alakalı yapılmış olan çalışmalarda anlam belirsizliği içeren Türkçe sözcüklerin hesaplamalı dilbilim uygulamalarıyla belirginleştirilmesi konusunda çalışmalar yapılmıştır [22]. Bu yapılan çalışmada hesaplamalı dilbilim, dilin işlenmesini matematiksel olarak ifade etmek üzere anlam ve anlatım arasındaki hesaplama yeteneğinin araştırılması olarak tanımlanmıştır. Anlam ve biçim arasında dönüşümü gerçekleştiren bir bilgisayar programı yardımıyla SAB problemlerine çözüm bulunması hedeflenmiştir. Hesaplamalı dilbilim aslında Türkçe metinlerin matematiksel olarak ifade edilmesi ve problemin matematik problemleri gibi bilgisayarların kolay anlayabileceği şekle dönüştürmeyi amaçlamaktadır.

Yapılmış olan farklı bir çalışmada ise tümevarımlı mantık programlama ile Türkçe için SAB uygulaması gerçekleştirilmiştir [23]. Tümevarımlı mantık programlama (TMP), makine öğrenmesi ve mantık programlamayı içeren bir yapay zekâ alanı olarak tanımlanmıştır. SAB probleminin kullanılan kaynak tipi baz alınarak, iki farklı yaklaşım ile değerlendirilmesinde bahsedilmiştir : bilgi tabanlı ve derlem tabanlı. Bilgi tabanlı yaklaşımlarda sözlük benzeri kaynaklardan alınan bilginin problem çözümünde kullanıldığından bahsedilmiştir. Derlem tabanlı çalışmada ise SAB problemleri çözümü için özel olarak hazırlanmış sözlüklerden faydalanılmıştır. Bulunan en başarılı yaklaşımların derlem tabanlı yöntemden elde edilen istatistiksel veya makine öğrenmesi algoritmaları olduğu söylenmiştir. Tanımlanan TMP'nin sözcükler arası ilişkilerin gösteriminde başarılı sonuçlar verdiği ve bu yeteneğinin kelime anlamı belirginleştirme konusunda kullanılmasının SAB alanında önemli bir gelişme sağlayabileceği söylenmiştir. Örnek olarak "Ali Ahmet'in dedesidir." ve "Ahmet Mustafa'nın oğludur." cümlelerinden çıkarım yaparak "Ali Mustafa'nın babasıdır" çıkarımı yapılabilmektedir. Böylelikle çıkarımlarıdan faydalanarak SAB'a ilişkin kelime tahmininde bulunabilmektedir.

Farklı diller arasında yapılan çevirilerde, SAB problemleri ile sıklıkla karşılaştığımız durumlar arasında yer almaktadır. Türkçe'den başka bir dile yapılan çevirilerde, ya da tam tersi çevirilerde, kullanılan bir kelimenin hangi anlamda kullanıldığının bilinmesi doğru çeviri yapılmasında büyük öneme sahiptir [24]. Daha önceden yapılmış bir çalışmada, dil çevirilerinde sözcüklerin bağlam içindeki anlamının belirlenmesi için Türkçe kelimelerin birbirleriyle olan ilişkisini gösteren çizelge(graph)'lardan faydalanılmıştır. Bu çizelgeler sözcüklerin birbirleriyle kullanılma sıklıklarında faydalanarak, İngilizce bir sözcük öbeğinin anlamının doğru tahmin edilmesinde kullanılmıştır. Örneğin yapılan çalışmada "suda yüzmek" sözcük öbeğinin "face in the water" olarak İngilizce'ye çevrilmesi yerine "swim on the water" şeklinde çeviri yapılmasının, "yüzmek" kelimesinin birlikte kullanıldığı kelimeler ile bağlantılı olduğu sonucu çıkarılmıştır. Bu şekilde yüksek başarım oranlarına sahip çevirilerin yapılması mümkün olmaktadır.

SAB problem çözümünde faydalanılan kelimelerin birbirleriyle ilişkilerini gösteren çizelge Şekil 3.1'de verilmiştir. Bu çizelgede aynı paragrafta bulunan kelimelerden hangisinin çok geçtiğini, birbirleriyle daha fazla kullanıldığı görülmektedir. Çizelgenin alt taraflarında bulunan kelimelerin, üst tarafta yer alan

kelimeler ile doğrudan bağlantısı olmayabilir. Örneğin, “Hava” kelimesinin “Havuz”, “Piknik” veya “Balık” kelimeleri ile doğrudan bir bağlantısı yoktur. Ancak bir metin içinde bu kelimelerin birlikte kullanılma olasılıkları “Kaban” veya “Atkı” kelimesi ile birlikte kullanılma olasılığından daha fazladır. Bu şekilde Türkçe’deki sözcüklerin birbirleriyle olan ilişkilerinin belirlenmesi, SAB problemlerinin çözümünde sıklıkla uygulanan bir yöntem olup, çok büyük kolaylıklar sağlamaktadır.



Şekil 3.1: Kelime ilişki çizelgesi.

Bir sözlükteki sözcükler arası anlam hiyerarşisinin belirlenmesine yönelik yapılmış bir çalışmada SAB yöntemlerinin kullanılmamasının, anlam hiyerarşinin düzgün oluşturulamamasına doğrudan etki ettiği belirtilmiştir [25]. Bir sözcüğün kendisini kapsayan bir üst sözcüğe bağlanması için, o kelimenin sözlük anlamının yanında kelimenin cümle içinde hangi anlamda kullanıldığı bilgisine de ihtiyaç duyulmaktadır. Örneğin, yapılmış olan çalışmada verilen örnekte “krem” kelimesi hem renk, hem de tene sürülen bir madde olarak sözlük anlamı belirtilmiştir. “güneş kremi” kelimesi ise anlam hiyerarşisinden dolayı hem “renk”, hem de “tene sürülen madde” üst ağacına bağlanmıştır; oysa ki “güneş kremi”nin renk ile alakası yoktur. Bu anlam karmaşasının önüne geçmek için SAB yöntemleri kullanılması

gerekmektedir. SAB yöntemleri ile üst ağaca bağlanacak olan kelimelerin sözlük anlamlarının yanında, cümlede kullanıldığı yere göre kazandığı anlam da dikkate alınsaydı; yukarıdaki örnekte verilen anlam karmaşasının da önüne geçilmiş olacaktı.

SAB problemlerinin farklı bir çalışma alanı da yabancı diller arasındaki kelime ilişkilerinin ortaya koyulmasına yönelik yapılan çalışmalardır. Aynı nesneyi veya anlamı ifade eden kelimeler, dilden dile değişiklik göstermektedir. Ancak köken olarak birbirine yakın dillerde bu farklılıklar daha az olmakta ve benzer kurallar ile kelime türetilmesi yapılabilmektedir. Uygurca ile Türkçe birleşik sözcüklerin karşılaştırılması şeklinde ortaya konulan bilgisayarlı çeviri çalışmasında, iki dilin benzer kurallarından faydanılarak derleme dayalı çalışan bir istatistiksel yöntem ortaya konulmuştur [26]. Böylelikle bir dilde yazılan birleşik kelimelerin farklı dildeki karşılığının tahmin edilmesi kolaylaşmaktadır.

İnsanların kullandıkları doğal konuşma ve yazım dillerinin zenginliği, bunların bilgisayarlar tarafından anlaşılmasını zorlaştırmaktadır. Metinde kullanılan bir kelimenin cümle içinde hangi manada(sözlük anlamı) ifade edecek şekilde kullanıldığının belirlenmesi, DDİ ile ilgili üzerinde en çok uğraşılan problemlerinden biri olan SAB problemlerinin ortaya çıkmasına neden olmuştur. Araştırmacıların bu konuya getirdiği yaklaşımlar genellikle Bilgi Tabanlı (Knowledge-based WSD) ve Derlem Tabanlı (Corpus-based WSD) olmak üzere 2'ye ayrılmaktadır. Derlem tabanlı yaklaşımlar ise istatistiksel veya makine öğrenme algoritmaları ile getirilen çözümlerden sonuca gitmektedir. Bir çalışmada N-Gram bilgisinden faydalanarak, muğlak kelime analizinin derlemlerden elde edilen örnekler üzerinde gerçekleştirilmesi sağlanmıştır. Örnek sayısı arttıkça başarı oranının daha da arttığı sonucuna varılmıştır [27].

SAB problemlerinde kelime anlamlarının istatistiksel tabanlı olarak çıkartabilmek için metin örneklerinin işlenmesi gerekemektedir. Daha önceden yapılmış olan bir çalışmada incelenecek olan metinler üzerinde önceden bazı işlemler yapmanın, SAB'ye yönelik kolaylıklar sağladığı görülmüştür. Örneğin, bir kelimenin kaç farklı anlam içerdiği, kelimenin anlamına göre kelimeden önce isim mi, sıfat mı, zarf mı... geldiğinin önceden belirlenmesi muğlak kelime tahmininde bize yardımcı olmaktadır. Bu işlem için sözlük olarak kullanılacak olan metinlerin önceden gezilmesi, bu metinler üzerinde anlam karmaşası yaratan kelime anlamlarının önceden sisteme tanıtılması, kelimelerin önceden etiketlenmesi gerekmektedir [28].

Yukarıda anlatılan önceden yapılmış çalışmalardan da anlaşılacağı gibi, Türkçe dili için yeni yeni farkındalık oluşmakta ve bu çalışmaların daha başlangıç aşamasında olduğu görülmektedir. SAB problemi, İngilizce için üzerinde uzun yıllardır çalışılmış ve başarılı sonuçlar elde edilmiş olan bir konudur. Türkiye'de SAB ile alakalı yapılan çalışmaların çoğu, İngilizce dili için yapılan çalışmaların Türkçe dili için de gerçekleştirilmesi şeklinde devam etmektedir. Ancak dilbilgisi kurallarının diller arasında farklılık göstermesi nedeniyle, probleme farklı bir dil için uygulanmış olan çözümü uygulamak, probleme yeni bir çözüm getirmek kadar çaba gerektirmektedir. Çünkü her ne kadar problem tanımları birbirine benzese de, dilin kendine özgü kuralları nedeniyle bulunan algoritmaların alınıp doğrudan kullanılması mümkün olmamaktadır.

Literatürde uygulanan yöntemlere paralel olarak bizim tanımladığımız SAB problemine de, istatistiksel metotlar ile makina öğrenmesi metotları ayrı ayrı uygulanmıştır. Bizim ele aldığımız yaklaşım bu yöntemlerin, metne uygulanış biçimi ile alakalıdır. İncelenen metnin tamamı veya cümle cümle incelenmesinin, tanımladığımız SAB probleminin çözümü üzerindeki etkileri araştırılmıştır. Muğlak kelimelerin doğru tahmininde yapılan bu inceleme kayda değer bir konu olarak görülmektedir.

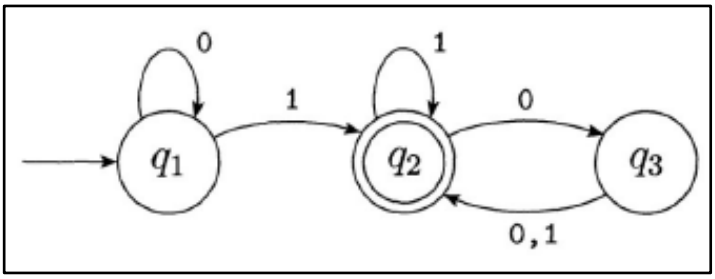
Giriş bölümde örnek olarak verilen "Avrupa'dan odun aldi" cümlesinin içerisinde yer alan "odun" kelimesinin anlamı, "odun" ya da "ödün" manasına gelebilmektedir. Bu muğlaklık ortaya SAB problemi çıkarmaktadır. Bu iki kelime arasında seçim yapmanın tek yolu cümleyi içeren metnin incelenip, İngilizce alfabe harfleri ile yazılmış "odun" kelimesinin gerçek manasının bulunmasıdır.

3.1. Sonlu Durum Dönüştürücüleri

Sonlu durum dönüştürücüleri, sınırlı sayıda tanımlanmış olan durumdan, durumlar arası geçişlerden ve bu geçişlerin oluşturduğu davranışlardan oluşan bir modeldir. Verilen girdi ve fonksiyonu kullanarak ortaya çıkan durumlara göre çıktı üretirler. Kontrol uygulamalarında sıklıkla kullanılmaktadırlar. İki farklı tipi aşağıda anlatılmaktadır.Sonlu durum dönüştürücülerinin matematiksel tanımını yapacak olursak:

Bir SSD 5-değişken grubundan (tuples) oluşur : $(Q, \sum, \delta, q0, F)$

- Q durumlar kümesi
- $\sum$ alfabe
- $\delta : Q\ x\ \sum \rightarrow Q$ durumlar arası geçişleri gösteren fonksiyon
- $q0\ \in\ Q$ başlangıç durumu
- $F\ \subseteq Q$ bitiş durumu



Şekil 3.2: Sonlu durum dönüştürücüsü örneği.

Şekil 3.2'de verilen örnek SDD için [29] matematiksel tanımı yapacak olursak,

- $Q : \{q0, q1, q2\}$
- $\sum : \{0, 1\}$
- $\delta\ tablosu$: Durumlar arası geçiş tablosu Tablo 3.1'de gösterilmektedir.

Tablo 3.1: Sonlu durum dönüştürücüsü için δ tablosu.

	0	1
q1	*q1*	*q2*
q2	*q3*	*q2*
q3	*q2*	*q2*

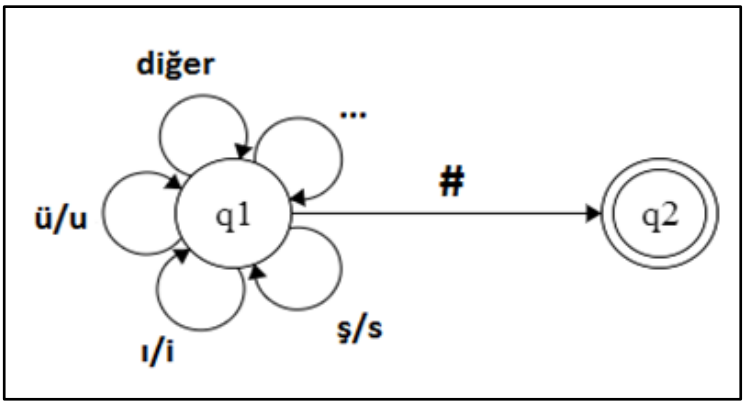
- $q0$: başlangıç durumu
- $F\ : \{q2\}$

3.1.1. Belirlenimci Sonlu Durum Dönüştürücüsü

Sonlu durum dönüştürücülerinin özel bir halidir. Bu dönüştürücü aşağıdaki 3 durumu içermelidir:

- Her durumdan (state) gidilecek koşulun tek bir durum göstermesi. Yani bir durumda başka duruma geçerken bir kelime ile sadece bir duruma gidilebilmesi
- Belirsiz veya boş kelimelerin durumlar arası geçişte yer almaması
- Tek bitiş durumunun bulunması (final state)

Şekil 3.3'de Türkçe klavye ile yazılmış bir metni, İngilizce klavye ile yazılmış hale dönüştürme problemine ait tasarlanan belirlenimci sonlu durum dönüştürücüsü yer almaktadır.



Şekil 3.3: Belirlenimci sonlu durum dönüştürücüsü.

Yukarıdaki tasarlanan dönüştürücünün özellikleri kontrol edildiğinde belirlenimci sonlu durum dönüştürücüsü (deterministic finite state transducer) olmasını gerektiren koşullar şöyledir:

- İlk adımdaki şartı sağlar çünkü herhangi bir durumdan diğerine giderken, belirsizlik söz konusu değildir. Örneğin, q1 durumundayken hangi karakter gelirse gelsin, o karakterin karşılığı olarak tek bir yere (yine q1) gidilmektedir.
- İkinci koşulu da sağlamaktadır çünkü dönüştürücüde hiç boş geçiş veya belirsiz karakter geçişi yoktur. Bütün geçişler bir değerle (harf ile) yapılmaktadır.
- Son koşulu da sağlamaktadır çünkü tek bitiş durumu söz konusudur. (q2 durumu)

Bu dönüştürücüde Türkçe klavyede bulunan her bir karakter için, İngilizce klavyede bir karşılığının bulunduğu ve bilinmeyen geçişlerin yer almadığı gözükmektedir. Bu SSD'ye ait matematiksel tanım aşağıda yer almaktadır:

- $Q : \{q1, q2\}$
- $\sum : \{a, b, c, ç, \ldots, u, ü, v, y, z\}$ (Türkçe Alfabesindeki Harfler)
- $\delta\ tablosu$: Durumlar arası geçiş tablosu Tablo 3.2'de gösterilmektedir.
- $q1$: İncelenen kelimenin ilk harfi
- $F : \{q2\}$

Tablo 3.2: Belirlenimci sonlu durum dönüştürücüsü için durum tablosu.

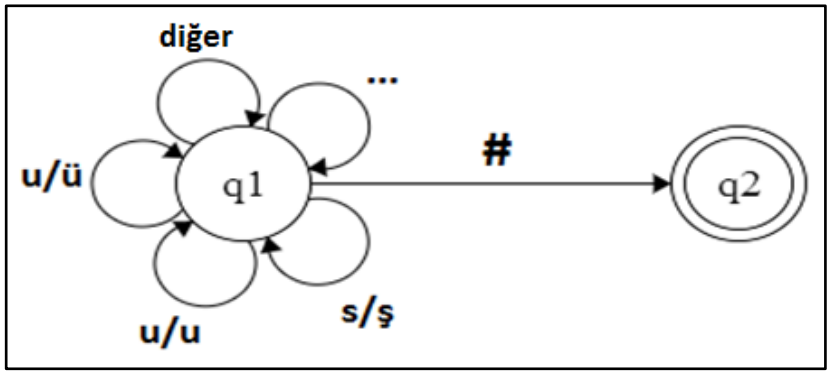
Durum	**ı:i**	**ö:o**	**ü:u**	**ş:s**	**ç:c**	**ğ:g**	**diğer**	**#**
q1	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q2*
q2	-	-	-	-	-	-	-	-

Türkçe karakterler ile yazılmış olan bir metnin İngilizce karakterler ile yeniden yazılması bir SAB problemi değildir. Çünkü bu işlemi %100 doğruluk oranı ile gerçekleştirebilecek bir çözüm yolu bulunmaktadır.

3.1.2. Belirlenimci Olmayan Sonlu Durum Dönüştürücüsü

Belirlenimci sonlu durum dönüştürücülerinin tersine her durumdan gidişin karışık olduğu ve her durum için bir sonraki kelimede nereye gidileceğinin belirli olmadığı dönüştürücülerdir.

Şekil 3.4'de İngilizce klavye ile yazılmış bir metni, Türkçe klavye ile yazılmış metne dönüştürme problemine ait tasarlanan belirlenimci olmayan sonlu durum dönüştürücüsü yer almaktadır.



Şekil 3.4: Belirlenimci olmayan sonlu durum dönüştürücüsü.

Yukarıdaki tasarlanan dönüştürücü incelendiği zaman belirlenimci olmayan sonlu durum dönüştürücüsü (non-deterministic finite state transducer) olduğunu ve tasarlanan dönüştürücüde belirsiz durumlar bulunduğu gözükmektedir. Örneğin "u" karakteri geldiği zaman, hem "u" karakterine hem de "ü" karakterine gitmek mümkündür. Bu belirsizlikler yüzünden belirlenimci olmayan sonlu durum dönüştürücü'leri insanlar tarafından kolay anlaşılan ve kullanılan; ancak bilgisayarlar tarafından algılanması ve kullanılması zor olan yapılardır.

Tasarlanan BOSSD'ye ait matematiksel tanımı yapacak olursak:

- Q : $\{q1, q2\}$
- $\sum$: $\{a, b, c, ç, \ldots, u, ü, v, y, z, q, w, x\}$ (Türkçe ve İngilizce Alfabesindeki Harflerin Birleşim Kümesi)
- $\delta\ tablosu$: Durumlar arası geçiş tablosu Tablo 3.3'de gösterilmektedir.
- $q1$: İncelenen kelimenin ilk harfi
- F : $\{q2\}$

Tablo 3.3: Belirlenimci olmayan sonlu durum dönüştürücüsü için durum tablosu.

Durum	i:i	i:i	o:o	o:ö	u:u	u:ü	s:s	s:ş	c:c	c:ç	g:g	g:ğ	diğer	#
q1	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q1*	*q2*
q2	-	-	-	-	-	-	-	-	-	-	-	-	-	-

İngilizce karakterler ile yazılmış olan bir metnin Türkçe karakterler ile yeniden oluşturulması bir SAB problemdir. Bu işlemi %100 doğruluk oranı ile hızlı biçimde gerçekleştirebilecek bir çözüm yolu bulunmamaktadır. Bu nedenle incelenen metin

üzerinde İstatistiksel ve Makina Öğrenmesi yöntemleri kullanarak çözüme gidilmektedir.

4. METİN ÖN İŞLEME

Türkçe dilinin kendine ait özellikleri göz önüne alındığında, bir kelimeye bazı ekler eklenerek kelimenin anlamı değiştirilebilmektedir. Bir anlam ifade etmek yerine kelimeleri bağlamaya veya onları vurgulamaya yönelik cümle içinde bazı kelimeler (bağlaç ve ünlem kelimeleri) sıkça kullanılmaktadır. Bu tarz eklerin ve kelimelerin ayıklanması için metin üzerinde, Kök Bulma (Stemming) ve Gürültü Ayıklama (Denoising) işlemleri yapılmaktadır.

Kelime kökü kavramı, bir anlam ifade eden her türlü harf grubu olarak tanımlanmaktadır. Türkçe'de hem "kök" halindeki bir kelime, hem de "kök + yapım_eki" almış kelime farklı anlamlara sahip olabilmektedir. Bu çalışmamızda, bir kelimede bulunan çekim ekleri atılarak geriye kalan harf grubu üzerinden işlem yapılmaktadır.

4.1. Kelime Ön İşleme

4.1.1. Durak Kelimeleri

Durak kelimelerinin bulunması demektir. Durak kelimeleri metin içinde geçen, cümlenin anlamına direk etkisi olmadığı halde metini anlaşılması için gerekli olan kelimelerdir. Örnek: benim, bile, birçoğu, biri, acaba, ama, ancak, artık …

4.1.2. Gereksiz Kelimelerin Temizlenmesi

Elimizde bulunan metin içerisinde bulunan gereksiz kelimelerin silinmesi, metnin gürültüden arındırılması işlemidir. Metnin hangi konuya ait olduğunun bulunması için metinde geçen kelimelerin ayıklanması gerekmektedir. Gürültüden arındırılma işlem için durak noktaları'nın silinmesi gerekmektedir.

4.1.3. Kelime Kökü Bulunması

Kelimelerin kökünün bulunması işlemidir. Kelime kökünün bulunması metnin hangi konuya ait olduğunun bulunmasında bize fayda sağlamaktadır. Ancak kelime

kökü bulunması işlemi tam olarak kök bulmak yerine kelimenin bir anlam ifade eden, en az ek atarak bulunabilecek parçasının bulunması işlemidir. Örnek : kitapçılarda → kitap – çı – lar – da

Bu kelimenin kökü "kitap" olduğu halde biz çalışmamızda bunu "kitapçı" olarak alacağız. Çünkü "kök + yapım eki = yeni bir kelime" olduğu için, anlam ifade eden, en az ek atarak kullanbileceğimiz harf öbeğini kullanıyoruz.

Tüm kelime kökü ve eklerinin bulunması işlemleri, Zemberek kütüphanesinin fonksiyonları yardımıyla gerçekleştirilmiştir. Zemberek'in önerdiği kök'lerin doğru olduğu kabul edilmiş ve bu sonuçlar üzerinden çalışmalar devam etmiştir. Kök veya eklerin hatalı belirlenmesi sistemin üreteceği sonuçlara doğrudan etki etmektedir.

4.2. İngilizce Karakterler ile Yazılmış Metnin Oluşturulması

Seçilen bir haber dosyası içindeki metni incelemek için ilk olarak metni teker teker kelimelerine ayırmamız gerekmektedir. Bunun için kelimeleri ayıracak ayıraç karakterlere ihtiyaç duyulmaktadır. Klavyeden girilebilecek tüm ayıraç karakterlerini token karakterler olarak tanımlıyoruz. Ayıraç karakterleri Şekil 4.1'de gösterilmektedir.

\n .:,;~é!'^+%&/()=?_@{}<>\"-*\\‘’

Şekil 4.1: Ayıraç karakterleri.

Token karakterlere göre bulunan kelimeler bir diziye atıldıktan sonra ikinci aşamaya geçilmektedir. Elimizde bulunan kelimelerde geçen Türkçeye özgü karakterler, İngilizce karakter karşılıkları ile değiştirilmektedir. Böylece İngilizce karakterler ile yazılmış kelimeler elde ediyoruz.

Yeniden Türkçe karaktere sahip metin oluşturma işlemini buradan elde ettiğimiz yeni metin üzerinde yapıyoruz. Böylelikle bir metne ait hem Türkçe karakterler ile yazılmış orjinal metin, hem de İngilizce karakterler ile yazımış bir metin elde ediyoruz. Bu iki metin doğru kelime tercihi doğruluk oranının hesaplanmasında bize yardımcı olmaktadır.

4.3. Türkçe Karakterler ile Yazılmış Metnin Oluşturulması

Metin Ön İşleme'den sonra elde ettiğimiz kelimeler, muğlak kelimenin bulunması için yapılan işlemlerde kullanılmıştır.

4.3.1. İngilizce Karakterlerin Bulunması

İngilizce'de bulunan ve Türkçe'de iki farklı olasılığı bulunan karakterlerin bulunmasına çalışılmaktadır. Şekil 2'de geçen karakterler yardımıyla kelimeler analiz edilmektedir. Herbir karakterin bulunduğu indeks hesaplanmaktadır.

- Meclis={2=c, 4=i, 5=s}
- anda={}
- yurutuldugu={1=u, 3=u, 5=u, 8=u, 9=g, 10=u}
- almistik={3=i, 4=s, 6=i}
- oldugu={0=o, 3=u, 4=g, 5=u}

4.3.2. Tüm Kelime Olasılıklarının Hesaplanması

Kelime içinde geçen muğlaklık oluşturabilecek harflerin bulunmasından sonra, bu harflerden elde edilebilecek tüm kelime olasılıkların hesaplanmasına çalışılmaktadır. Herbir harf için 2 tane olasılık bulunmaktadır. Örneğin "c" harfi yerine "c" veya "ç" harfi gelebilir. Bu durumda"n = karakter sayısı" olursa; "2n = üretilebilecek kelime sayısı" olacaktır.

- Meclis = [Meclis, Mecliş, Meclıs, Meçlis, Meclış, Meçliş, Meçlıs, Meçlış]
- anda = []
- yurutuldugu = [yurutuldugu, yurutuldugü, yurutulduğu, yurutuldügu, yurutulduğü, yurutuldügü, yurutuldüğu, yurutüldugu, yurutüldugü, yurutülduğu, yurutüldügu, yurutülduğü, yurutüldügü, yurutüldüğu, yurutüldügü, yurütuldugu, yurütuldugü, yurütulduğu, yurütuldügu, yurütulduğü, yurütuldügü, yurütuldüğu, yurütuldüğü, yurütüldugu, yurütüldugü, yurütülduğu, yurütüldügu, yurütülduğü, yurütüldügü, yurütüldüğu, yurütüldüğü, yürutuldugu, yürutuldugü,

yürutulduğu, yürutuldügu, yürutulduğü, yürutuldügü, yürutuldüğu, yürutuldüğü, yürütüldugu, yürutüldugü, yürutülduğu, yürutüldügu, yürutülduğü, yürutüldügü, yürutüldüğu, yürutüldüğü, yürütuldugu, yürütuldugü, yürütulduğu, yürütuldügu, yürütulduğü, yürütuldügü, yürütuldüğu, yürütuldüğü, yürütüldugu, yürütüldugü, yürütülduğu, yürütüldügu, yürütülduğü, yürütüldüğü]

- almistik = [almistik, almistık, almiştik, almıstik, almiştık, almıstık, almıştik, almıştık]
- oldugu = [oldugu, oldugü, olduğu, oldügu, olduğü, oldügü, oldüğu, oldüğü, öldugu, öldugü, ölduğu, öldügu, ölduğü, öldügü, öldüğu, öldüğü]

4.3.3. Olasılıklar İçinden Türkçe Kelimelerin Bulunması

Bütün kelime olasılıkları bulunduktan sonra elde ettiğimiz olasılıkların hepsinin Türkçe kelime olmadığı görülmektedir. Amacımız bu olasılıklar içinden Türkçe olan kelimelerin ayrıştırılarak, bu kelimeler üzerinden çalışmalara devam edilmesini sağlamaktır. Bunun için Zemberek açık kaynak koda sahip kütüphane'nin bize önerdiği sonuçlar üzerinden çalışmalara devam edilmesi sağlanmıştır.

- Meclis = [Meclis]
- anda = []
- yurutuldugu = [yürütüldüğü]
- almistik = [almıştık]
- oldugu = [olduğu, öldüğü]

Türkçe diline ait muğlak kelime olasılıkları bulunduktan sonra, bu muğlak kelime olasılıkları içinden doğru kelimenin bulunması için yapılacak çalışmalara geçilmiştir. Bu çalışmalar muğlak kelimelerin cümle içinde kullanıldığı yere ait bilgilerden faydalanarak, Bölüm 5'de anlatılan İstatistiksel Yöntemler ve Bölüm 6'da anlatılan Makine Öğrenmesi Yöntemleri sonuçlarına göre doğru kelime tercihinin yapılması sağlanmıştır.

5. İSTATİSTİKSEL YÖNTEMLER

5.1. N-Gram

N-Gram'lar verilen bir cümle içinde geçen "n" tane kelime dizisinin arka arkaya gelmesiyle oluşmuş gruplardır. Kendisinden önce gelen "n-1" tane kelimeye bakarak, kendisinden sonra gelecek olan kelimeleri bulmaya çalışan dil modelidir. Türkçe dili için Doğal Dil İşleme uygulamalarında, kelimelerin ard arda gelme olasılıklarının hesaplanmasında kullanılır.

N-Gram'lar :

- İstatistiksel makine çevirisi sistemlerinde,
- Yazım hatalarının düzeltilmesinde,
- Sözcük etiketleme (POS tagging),
- Doğal dil üretme,
- Yazar belirlenmesi ...

alanlarda kullanılmaktadır.

N-Gramların büyüklüğü 1 ise "unigram" (veya "onegram"); büyüklük 2 ise "bigram" (veya "digram"); büyüklük 3 ise"trigram"; ve büyüklük 4 veya daha fazla ise "n-gram"olarak adlandırılmaktadır.

Örneğin; 'abc' kelime dizisinden oluşan cümle için 'c' kelimesinin 'ab' kelime çiftinden sonra gelmesi olasılığı Denklem (5.1)'deki gibidir.

$$Pr\ (c|ab) = \frac{C(abc)}{C(ab)} \tag{5.1}$$

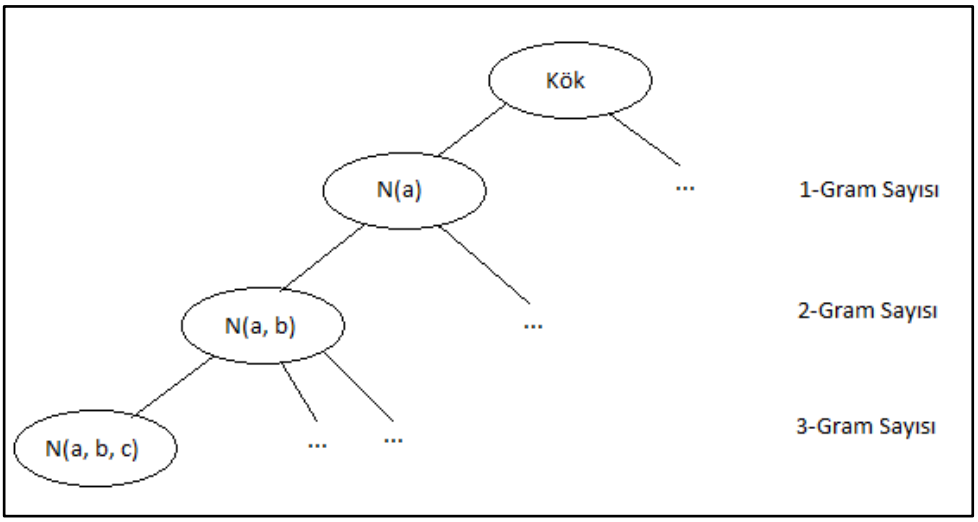
N-Gram sayıları, oluşturulan hazırlanan sözlüğün büyüklüğüne bağlı olarak değişmektedir. Sözlük'ün büyük olması, bulunan N-Gram ilişkilerinde daha doğru sonuçlar elde etmemize neden olmaktadır.

Bölüm 7.1'de anlatılan "Veri Kümesi Hazırlama" işleminden sonra, kelimeler arasındaki N-Gram ilişkileri veri tabanından uygun sorguların çekilmesi ile

hesaplanmaktadır. Yapılan çalışmada 1-Gram, 2-Gram ve 3-Gram hesaplamaları yapılmıştır. Daha fazla sayıdaki N-Gram ilişkileri kullanılmamasının nedeni,

- Çok fazla kısıt içerdiği için, arka arkaya en az 4 kelimenin birlikte kullanılması gibi, kelimelerin birlikte geçme sıklıkları çok azalmaktadır.
- 3-Gram ile elde ettiğimiz sonuçlar bizim ihtiyaçlarımıza cevap verecek nitelikte olduğu içinfazla sayıdaki N-Gram ilişkileri kayda değer yarar sağlamamıştır.

Arka arkaya gelen 3 kelimeye ait N-Gram ağacı, Şekil 5.1'de gösterilmektedir. Bu ağaçta kökten sonra gelen her bir ilişki, alt sıradaki ilişkilere göre daha az bilgi, ancak daha fazla olasılık içermektedir. "ab" kelime çiftinin metinlerde geçme sıklığı, "abc" kelime çiftinin birlikte kulanılma sıklıklığından daha fazladır ancak bize kelime birliktelikleri hakkında daha az bilgi vermektedir.



Şekil 5.1: N-Gram ağacı.

N-Gram ilişkileri genellikle istatistiksel alanlarda sıklıkla kullanılmaktadır. Günümüzde bir çok hesaplamada bu bilgilere ihtiyaç duyulmakta, gelecek ile ilgili çıkarım yapmak için geçmişte kullanılmış verilerden yararlanılmaktadır. Google N-Gram Viewer [30] bu konuda internette ücretsiz olarak kullanıma sunulan en popüler N-Gram uygulamasıdır. Bu uygulamaya ait veri tabanında 1500-2008 yılları arasında incelenip elektronik ortama aktarılan 5.2 milyon kitaptan elde edilen veriler bulunmaktadır. Amerkan İngilizcesi, İngiliz İngilizcesi, Fransızca, Almanca,

İspanyolca, Rusça ve Çince dillerinde olmak üzere yaklaşık 500 milyon kelimeyi içermektedir. Aranmak istenilen kelimelerin, hazırlanan veri tabanında geçme sayılarını vermektedir.

SAB probleminin çözümüne N-Gram'ların etkisini inceleyecek olursak, her bir kelime için o kelimeden önce ve sonra gelen kelimeler önem taşımaktadır. Arka arkaya gelen kelime çiftlerinin kullanılma sıklıkları, istatistiksel olarak SAB probleminde doğru kelime seçeneğinin bulunmasında yardımcı olmaktadır.

SAB problemi oluşturan bir kelimenin diğer metinlerde geçme sıklığı (bu kelimenin kullanılma sayısı), doğru kelimenin bulunması hakkında ipucu vermektedir. İlk olarak incelenen kelimelerin kökleri bulunmaktadır. Kök halindeki kelimelerin hazırladığımız sözlük içinde geçme sayıları hesaplanmaktadır. Bu hesaplama muğlak kelimeye ait tüm olasılıklar için ayrı ayrı yapılmaktadır.

Örneğin, muğlak kelime "odun" olursa, bu kelimeden 2 tane Türkçe kelime anlaşılabilir: "odun" veya "ödün". Öncelikle 1-Gram uygulanarak "odun" kelimesinin hazırlanan sözlük içinde geçme sıklığı ile "ödün" kelimesinin geçme sıklığı karşılaştırılarak daha sık geçen kelime tercih edilmektedir.

2-Gram uygulamasında "kelime + sonraki_kelime" veya "önceki_kelime + kelime" çiftinin hazırlanan sözlük içinde geçme sıklıkları karşılaştırılmıştır. Örneğin; "odun aldı" veya "ödün aldı"; "Avrupa odun" veya "Avrupa ödün".

3-Gram işleminde ise "önceki_kelime + kelime + sonraki_kelime" üçlüsünün hazırlanan sözlük içinde geçme sıklıkları karşılaştırılarak bir sonuca gidilmiştir. Örneğin; "Avrupa'dan odun aldı" veya "Avrupa'dan ödün aldı".

Tablo 5.1: Muğlak kelime geçme sayıları tablosu.

	1-Gram	2-Gram	3-Gram
Odun	243	-	-
Ödün	261	-	-
odun aldı	-	1	-
ödün aldı	-	25	-
Avrupa odun aldı	-	-	0
Avrupa ödün aldı	-	-	1

Tablo 5.1'de veri tabanına kaydedilmiş olan sözlükten yapılan sorgulamalar sonucunda elde ettiğimiz N-Gram değerleri gözükmektedir. Yukarıdaki örneklerden

de anlaşılacağı gibi istatistiksel yöntemlerin belli bir seviyede bahsedilen problemi çözmesine rağmen, bazı durumlarda yetersiz kalabildiği görülmektedir.

5.2. Zincir Kuralı ve Olasılık

Markov zinciri, mevcut bir durum verildiğinde gelecek durumların geçmiş durumlardan bağımsız olarak bulunmasını ifade eder. Yani mevcut durumun açıklaması, gelecekte karşılaşacağımız durumları bulmamızı sağlayacak bilgileri içermektedir. Gelecek durumların tahmini, olasılıksal bir süreçle gerçekleşmektedir.

Bir kelimenin cümle içinde bulunma olasılığı, cümle içinde kendisinden önce gelen diğer kelimeler yardımıyla hesaplanabilir. Bu olasılık aynı cümle içinde arda arda gelen kelimelerin tahmininde kullanılabilmektedir. Muğlak bir kelimenin aynı cümle içinde geçen diğer kelimeler ile olan ilişkisine ait Markov Zinciri formülü Denklem(5.2)'de verilmiştir.

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2) \dots P(w_n|w_1^{n-1}) = \prod_{k=1}^{n} P\left(w_k|w_1^{k-1}\right) \tag{5.2}$$

Bir örnek ile açıklayacak olursak, "Çocuk topu cama attı" cümlesi için hesaplanacak olan Zincir Kuralı formülü Denklem(5.3)'de verilmiştir.

$$P(\text{Çocuk topu cama attı}) = P(\text{çocuk})\ x\ P(topu|\text{çocuk})\ x\ P(cama|\text{çocuk}, topu)\ x\ P(\text{attı}|\text{çocuk}, cama, topu) \tag{5.3}$$

Bu örnekte görüldüğü gibi her bir kelimenin cümle içinde bulunma olasılığı kendisinden önce gelen kelimelere bağlanmıştır. Bir kelimenin cümleye olan etkisi incelenebilmekte, cümle üzerinde hangi kelimenin kritik öneme sahip olduğu bulunabilmektedir.

$$P(\text{attı}|\text{çocuk}, cama, topu) = \frac{C(\text{çocuk}, cama, topu, \text{attı})}{C(\text{çocuk}, cama, topu)} \tag{5.4}$$

i) $C(çocuk, cama, topu, attı) = 4\ kelimenin\ arka\ arkaya\ bulunma\ sayısı$

ii)$C(çocuk, cama, topu) = 3\ kelimenin\ arka\ arkaya\ bulunma\ sayısı$

Eğer bir cümlede fazla sayıda kelime bulunursa, zincir kuralının uygulanmasında problemlerle karşılaşılmaktadır. Çok fazla kelimenin bir arada bulunduğu bir olasılığın hesaplanması hem zaman hem de bellek bakımından aşırıya kaçılmaktadır. Bunu engellemek için "Markov Varsayımı (Assumption)" yapılmaktadır. "Markov Varsayımı"nda bir cümlede işlem yapılan kelimeden önce gelen diğer tüm kelimeleri hesaba katmak yerine; sadece öncesinde gelen bir, iki veya üç kelimenin işleme katılması bize ard arda gelen kelime grubu hakkında detaylı bilgi verebilmektedir. Yukarıdaki örnek cümle için Tablo 5.2'de yapılması gereken varsayımlar ve Denklem(5.5)'de uygulanacak formül gösterilmektedir.

Tablo 5.2: Markov varsayımı tablosu.

	"Çocuk cama topu attı"	
Unigrams	**Bigrams**	**Trigrams**
Çocuk	<s> Çocuk	<s> <s> Çocuk
cama	Çocuk cama	<s> Çocuk cama
topu	cama topu	Çocuk cama topu
attı	topu attı	cama topu attı
</s>	attı </s>	topu attı </s>

$$P(attı|topu), P(attı|cama, topu) \text{ veya } P(attı|çocuk, cama, topu) \tag{5.5}$$

Markov Varsayımını Denklem (5.6)'daki 2-gram dil modeli ile formülize edecek olursak,

$$P(w_1|w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})} \tag{5.6}$$

6. MAKİNA ÖĞRENMESİ YÖNTEMLERİ

SAB problemlerinin çözümünde anlamsal analiz için literatürde sözlük tabanlı yaklaşımların yanında, makina öğrenmesi tabanlı yöntemler de denenmektedir. Çalışmamızda aşağıda yer alan özgün makina öğrenmesi yaklaşımları kullanılmaktadır.

Kullanılan makina öğrenmesi yaklaşımları Bölüm 7.3.2'de anlatılan Weka Makina Öğrenmesi Aracı yardımıyla uygulanmaktadır. Yapılan çalışmada hangi makina öğrenmesi yönteminin kelimeler üzerinde daha iyi sonuç verdiğini anlamak için AdaBoost, J48 Tree ve Naive Bayes olmak üzere 3 farklı algoritma ile eğitim kümeleri eğitilmiş ve testle yapılmıştır.

Makina öğrenmesi algoritmaları genel olarak öngörmeli (supervised) ve öngörmesiz (unsupervised) olmak üzere ikiye ayrılır. SAB problemine çözüm getirmek için yapılan çalışmalarda öngörmeli yaklaşımların, muğlak kelimeler üzerinde öngörmesiz yaklaşımlardan daha iyi sonuçlar verdiği görülmüştür [31]. Bu sebeble çalışmalarımızda öngörmeli makina öğrenmesi yaklaşımlarından 3 tanesi seçilmiştir.

6.1. Makina Öğrenmesi Algoritmaları

6.1.1. Naive Bayes

Naive Bayes Sınıflandırıcısı Makina öğrenmesi yöntemlerinde kullanılan öngörmeli bir sınıflandırıcıdır. Bayes teoreminin bağımsız önermeler ile basitleştirilmiş halidir.

Bu sınıflandırıcı farklı Bayes yöntemleri ile implement edilebilmiştir. Bernoulli modeli olarak adlandırılan yaklaşımda kelimeler arasında herhangi bir ilişki bulunmayan ve ikili (binary) kelime özelliklerinin kullanıldığı Bayes Ağ Modeli (Bayesian Network) kullanılmıştır [31]. Çok terimli (multinominal) modelde ise eğitim setindeki kelime sayılarından faydalanarak 1-Gram kelime modelleri ile çözüme gidilmeye çalışılmıştır [32]. Bu iki yöntemin kıyaslanmasında ise küçük boyutlu sözlükler üzerinde Bernoulli yönteminin, büyük boyutlu sözlükler üzerinde ise çok terimli yaklaşımın daha iyi sonuçlar verdiği saptanmıştır [33].

Bayes Teoremi, bilgi edinilmek istenen bir sistem hakkında, toplanan veriler yardımıyla çıkarım yapılmasını sağlar. Bir sisteme yeni veriler eklendiğinde, sistemin o anki durumunu tahmin etmemize yardımcı olur. Bayes teoremine ait denklem, Denklem(6.1)'de verilmiştir.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{6.1}$$

i) $P(A|B) =$
B *olayı gerçekleştiği durumda* A *olayının meydana gelme olasılığı*
ii) $P(B|A) =$
A *olayı gerçekleştiği durumda* B *olayının meydana gelme olasılığı*
iii) $P(A)$ *veya* $P(B) = A$ *veya* B *olayının kendi başına gerçekleşme olasılığı*

Naive Bayes Sınıflandırıcısı,

- sınıflandırma için gerekli olan parametreler az miktardaki eğitim kümesinde bile başarılı olabilmekte,
- niteliklerin hepsi aynı önem derecesine sahip,
- nitelikler birbirinden bağımsız (bir niteliğin değeri diğer nitelik hakkında bilgi içermemekte)

şekilde yer almaktadır.

Naive Bayes Sınıflandırıcısı aşağıdaki Denklem(6.2) ile ifade edilmektedir.

$$P(A|S_1, S_2, \dots S_n) = \frac{p(A)\prod_{i=1}^{n} p(S_i|A)}{\prod_{i=1}^{n} p(S_i)} \tag{6.2}$$

i) $P(A|S_1, S_2, \dots S_n) =$
A *olayının hangi herhangi bir* S *sınıfına ait olma olasılığı*
ii) $p(A) = A$ *olayının tek başına hangi olma olasılığı*

iii) $\prod_{i=1}^{n} A_t h_t(x) =$

A olayının herbir S sınıfına ait olma olasılıklarının çarpımı

iv) $\prod_{i=1}^{n} p(S_i) =$

Herhangi bir S sınıfına ait olma olasılıklarının çarpımı

Tablo 6.1: Sınıflandırıcı örnek veri kümesi tablosu.

İş	Maaş	Yaş	Tecrübe
Yazılım	3000	25	3
Muhasebe	2000	22	2
Yazılım	5000	30	9
Muhasebe	2500	30	7
Yazılım	2800	22	1
Muhasebe	750	18	0
Yazılım	7500	40	15
Muhasebe	6000	40	15

Bir örnekle Naive Bayes Sınıflandırıcısını anlatacak olursak, Tablo 6.1'deki verilerden faydalanarak "Maaş : 3000, Yaş : 30, Tecrübe : 5yıl" olan kişinin hangi mesleği yaptığını nasıl bulabiliriz?

$$P(Yaz.) = \frac{p(Yaz.)p(Maaş|Yaz.)p(Yaş|Yaz.)p(Tecrübe|Yaz.)}{p(Yaz.)p(Muh.)} \quad (6.3)$$

$$P(Muh.) = \frac{p(Muh.)p(Maaş|Muh.)p(Yaş|Muh.)p(Tecrübe|Muh.)}{p(Yaz.)p(Muh.)} \quad (6.4)$$

Yukarıda yazan $P(Yaz.)$ ve $P(Muh.)$ değerleri sonuçlarına göre kişinin Muhasebe mesleği yaptığını tahmin edebiliriz.

6.1.2. AdaBoost

AdaBoost(Adaptive Boosting), Yoav Freund ve Robert Schapire tarafından geliştirilmiş olan bir makina öğrenmesi algoritmasıdır [34]. Boosting algoritmasının özelleştrilmiş halidir. Basit zayıf sınıflandırıcıların linear kombinasyonundan güçlü bir sınıflandırıcılar oluşturulmaktadır [35]-[40].

Boosting algoritmalarında genel olarak;

- Eğitim kümesinde bulunan her özniteliğin bir ağırlığı bulunmaktadır.
- Her bir eğitim aşamasında, her sınıflandırıcı için yapılan sınıflandırmaya bağlı olarak özniteliklerin ağırlığı güncellenmektedir.
- Bir özniteliği sınıflandırmak için, sınıflandırıcıların doğruluk oranına bağlı olarak ağırlıklı ortalamaları alınmaktadır.

AdaBoost algoritmasının Boosting algoritmasından farklarını sıralayacak olursak;

- Daha hızlıdır.
- Basit ve kolay implement edilebilir.
- Farklı alanlarda kullanılabilir.
- Aşırı öğrenme(overfitting) davranışlarına eğimli değildir.
- Sınıflandırmanın doğruluğunu arttırmaktadır.
- Zayıf sınıflandırıcıların önceki hata bilgilerine ihtiyaç duymaz.
- Gürültü ve sınır değerlere karşı duyarlıdır.
- Her bir eğitim adımında bu zayıf sınıflandırıcıların lineer birleşiminden kuvvetli bir sınıflandırıcı elde edilir.

AdaBoost algoritmasının matematiksel tanımını yapacak olursak:

$$f(x) = \sum_{t=1}^{T} A_t h_t(x) \tag{6.5}$$

$$h_t(x): x \to \{-1, 1\}$$

$$H(x) = sign\big(f(x)\big)$$

i) $h_t(x) = Her\ bir$ öznitelik'e ait zayıf sınıflandırıcı

ii) $H(x) =$ Güçlü sınıflandırıcı

AdaBoost Algoritmasına ait sözde kod Şekil 6.1'de yer almaktadır.

Eğitim verisi $(x_1, y_1), \dots, (x_m, y_m)$ olsun.
$y_i \in \{-1, +1\}, \quad x_i \in X$
for $t = 1, \dots, T$
    crate distribution D_t on $\{1, \dots, m\}$
    select weak classifier with smallest error $\in_t$ on D_t
    $\in_t = Pr_{D_t}[h_t(x_i) \neq y_i]$
    $h_t(x): X \rightarrow \{-1, 1\}$
Output single classifier $H_{final}(x)$

Şekil 6.1: AdaBoost algoritmasına ait sözde kod.

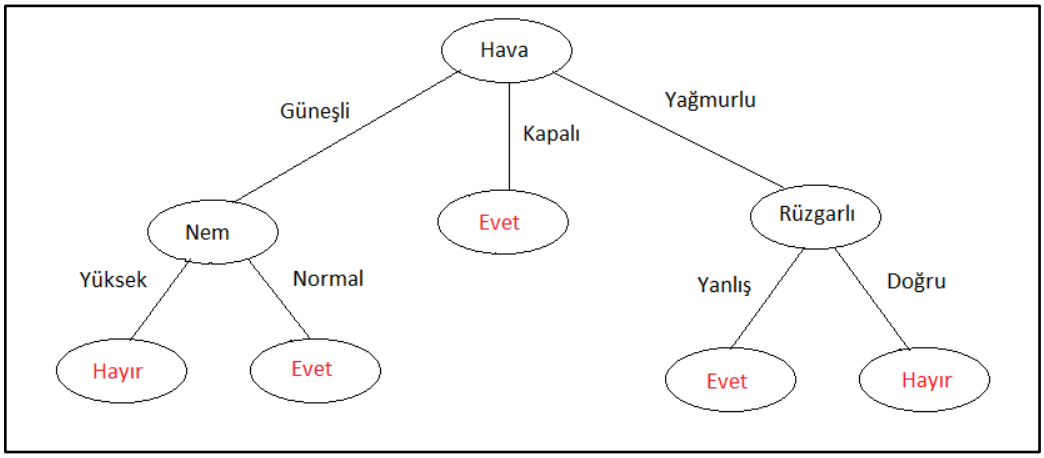
Eğitim kümesini temsil eden her öznitelik bir zayıf sınıflandırıcı olarak tanımlanmaktadır. Her bir öznitelik üzerinde bir zayıf sınıflandırıcı eğitilmektedir. Eğitim kümesi içinde yanlış sınıflandırılan verilerin ağırlıkları arttırılırken, doğru sınıflandırılan verilerin ağırlıkları azaltılmaktadır. Her adımda zayıf sınıflandırıcılar eğitildikten sonra bu ağırlıklar ile hata hesaplaması yapılmaktadır. Döngünün bir sonraki adımında hesaplanan hata göz önünde bulundurularak, en küçük hata oranına sahip öznitelik üzerinden eğitim devam etmektedir. Eğitim setindeki tüm elemanlar gezildikten sonra güçlü sınıflandırıcı elde edilmektedir [36], [40].

6.1.3. J48 Ağacı

J48 Ağacı makina öğrenmesi yöntemi C4.5 algoritmasının Weka aracında implement edilmiş halidir. Ross Quinlan tarafından geliştirilmiş olan C4.5

algoritması [40]-[42] karar ağaçları oluşturmak için kullanılmaktadır. Bu algoritma genellikle istatistiksel sınıflandırmada kullanılır. Şekil 6.2'de bir karar ağacının yapısı yer almaktadır. Karar ağaçlarının özellikleri;

- Anlaşılması ve yorumlanması basittir.
- Hızlı veri ön işlemesi ile veri kullanılabilir hale getirilerek kullanılabilir.
- Hem sayısal hem de metinsel verilerin işlenmesi için kullanılabilir.
- Her karar adımı görüntülenebilir ve yorumlanabilirdir.
- Düşük hesaplama karmaşıklığına sahiptir. Basit ve hızlı olmasından dolayı yüksek miktardaki veriyi kısa sürede işleyebilir ve alternatif diğer yöntemlere göre veri miktarı arttığında daha avantajlı bir yöntemdir.



Şekil 6.2: Dışarıda oyun oynamak için oluşturulan karar ağacı.

C4.5 algoritmasının normal karar ağacı oluşturma algoritmalarından farkı,

- Bütün öznitelikler için üzerinde çalışılan veri kümesine ait entropi hesaplanır
- En küçük entropi'e sahip öznitelik bulunduğunda veri kümesinin bölünmesi gerçekleştirilir.
- Bu öznitelik'i içerek yeni bir karar ağacı oluşturulur.
- Bu işlemler bütün öznitelikler için tekrarlanır.
- Yüksek entropi değerine sahip öznitelik sınıflandırmanın mükemmelleştirilmesinde kullanılır.

Eğer hesaplanan Entropi $H(S) = 0$ olursa S kümesi mükemmel sınıflandırılmış demektir. Entropi şu formülle hesaplanır.

$$H(S) = -\sum_{x \in X} p(x) log_2 p(x) \tag{6.6}$$

i) $S = Entropi\ hesaplamak\ için\ kullanılacak\ olan\ data$

ii) $X = S\ veri\ kümesi\ içindeki\ her\ bir\ sınıf$

iii) $p(x) = \frac{x\ sınıfı\ içindeki\ eleman\ sayısı}{S\ sınıfı\ içindeki\ eleman\ sayısı}$

Bilgi Kazancı(Information Gain), Entropi'den farklı olarak her bir öznitelik için hesaplanmaktadır. Yani bir A özniteliği için bilgi kazancı, o özelliğin bağlı bulunduğu bütün veri kümesine ait entropi'nin, sadece A öznitelliğine ait ağacın entropi'si arasındaki farka eşittir. Ağacın bölünmesi sonrasında elde ettiğimiz kazanımı sayısal olarak ifade etmektedir. En büyük bilgi kazancına sahip öznitelik veri kümesine ait ağacın bölünmesinde kullanılır.

Bilgi Kazancı şu formülle hesaplanır.

$$IG(A) = H(S) - \sum_{t \in T} p(t) H(t) \tag{6.7}$$

i) $H(S) = S\ kümesinin\ Entropi'si$

ii) $T = S\ veri\ kümesinin\ ayrılması\ ile\ oluşan\ A\ öznitelik'ine\ ait\ alt\ küme$

iii) $p(t) = \frac{t\ sınıfı\ içindeki\ eleman\ sayısı}{S\ sınıfı\ içindeki\ eleman\ sayısı}$

iv) $H(t) = t\ alt\ kümesinin\ entropi'si$

6.2. Yerel Bazlı İnceleme

"Yerel İnceleme" veya "Cümle Bazlı İnceleme" olarak adlandırdığımız bu yöntemde, eğitim kümesi içinde bulunan tüm metinler teker teker cümlelere ayrılarak, cümle bazında değerlendirmeye alınmaktadır.

Bir kelimenin, aynı cümle içinde yer alan diğer kelimeler üzerindeki etkisini anlamak için cümle bazlı inceleme yöntemine başvurulmuştur. Türkçe'de bazı kelime gruplarının birbirleri ile aynı cümle içinde sıklıkla kullanılması, doğru kelime tahminine yardımcı olmaktadır. Örneğin,

- "gol" kelimesi ile "maç", "göl" kelimesi "yüzmek veya piknik",
- "sınır" kelimesi ile "ülke", "sinir" kelimesi ile "insan",
- "yasa" kelimesi ile "meclis", "yaşa" kelimesi ile "insan"
- "acı" kelimesi ile "yemek", "açı" kelimesi ile "üçgen" ...

Bu şekilde aynı cümle içinde kullanılma sıklığı daha fazla olan kelimeler baz alınarak, muğlak kelimeler için doğru kelime tercihi yapılmaktadır. Eğitim setinde bulunan tüm metinlerin önceden elden geçirilmesi ve muğlaklığa neden olan kelime için, sadece o muğlak kelimeye ait bir eğitim verisi (".arff" uzantılı dosya) oluşturulmalıdır.

Yerel bazlı makina öğrenmesi yaklaşımı için tüm metinlerin cümlelere ayrılması gerekmektedir. Cümle sonu ayıraçları (nokta, üç nokta, ünlem, enter..) kullanılarak metinler cümlelere ayrılmaktadır. Bu cümlelerde geçen her bir kelime, o kelimeye ait öznitelik vektörünün bir elemanını olarak dosyaya eklenmektedir.

Yerel bazlı inceleme'de kullanılacak olan ".arff" uzantılı dosyada kullanılacak olan öznitelik vektörünün oluşturulmasına ait algoritma Şekil 6.3'de verilmiştir.

1. Eğitim veri setinden bir metin dosyası al.
2. Bu metin dosyasını cümlelere ayır.
3. Metin dosyasındaki cümleler içinde muğlak kelime var mı diye kontrol et.
4. Eğer cümle içinde muğlak kelime var ise o cümle içinde geçen tüm kelimeleri öznitelik vektörüne kaydet.
5. Muğlak kelime yok ise metindeki bir sonraki cümleye geçip devam et.
6. Eğitim setindeki tüm metinler bitinceye kadar işleme devam et.

Şekil 6.3: Yerel öznitelik vektörünün oluşturulmasına ait algoritma.

Bu dosya içindeki öznitelikler yukarıdaki algoritmada oluşturulan öznitelik vektörünün elamanlarıdır. Öznitelik vektörünün toplam öznitelik sayısı, eğitim kümesi içinde muğlak kelime ile aynı cümlede geçen toplam kelime sayısıdır.

Tablo 6.2: Yerel bazlı öğrenme için örnek eğitim seti tablosu.

Cümle No	
1	ABCDE.
2	BEAFBH.
3	GEHDA.

Yukarıdaki tabloda her harf bir kelimeyi temsil etmektedir. "A" kelimesinin muğlak kelime olduğunu varsayarsak, Tablo 6.2'de bu kelimeyi içeren cümleler gösterilmektedir. Bu cümleler için hazırlanacak olan ".arff" uzantılı dosyanın içeriği öznitelikler ve bu özniteliklere ait verilerden oluşmaktadır.

Tablo 6.3'de verilen her bir öznitelik (attribute), Tablo 6.2'de bulunan cümlelerdeki kelimelerdir.

Tablo 6.3: Attribute relation file format dosyası için öznitelik tablosu.

attribute	a	Real
attribute	b	Real
attribute	c	Real
attribute	d	Real
attribute	e	Real
attribute	f	Real
attribute	g	Real
attribute	h	Real

Tablo 6.4'da verilen her bir satır içinde muğlak kelimenin bulunduğu cümleyi temsil etmektedir. Her bir değer ise o cümle içinde kelimelerin kaçar defa geçtiğini göstermektedir.

Tablo 6.4: Attribute relation file format dosyası için veri tablosu.

a	b	c	d	e	f	g	h
1	1	1	1	1	0	0	0
1	2	0	0	1	1	0	1
1	0	0	1	1	0	1	1

Tüm muğlak kelimeler için ayrı ayrı ".arff" uzantılı dosya hazırlanmıştır. Eğitim setimizde toplam 3.947 tane muğlak kelime için dosya bulunmaktadır.

6.3. Küresel Bazlı İnceleme

"Küresel İnceleme" veya "Metin Bazlı İnceleme" olarak adlandırılan bu yöntemde, eğitim kümesi içinde bulunan her bir metin kendi içinde bir bütün kabul edilerek, metin bazında değerlendirmeye alınmaktadır.

Bir kelimenin, aynı metin içinde yer alan diğer kelimeler üzerindeki etkisini anlamak için cümle bazlı inceleme yöntemine başvurulmuştur. Bu yöntemde aynı metin içinde birlikte kullanılma sıklığı daha fazla olan kelimeler önem kazanarak, muğlak kelimeler için doğru kelime tercihi yapılmaktadır. Yerel Bazlı İnceleme'de olduğu gibi, eğitim setinde bulunan tüm metinlerin önceden elden geçirilmesi ve muğlaklığa neden olan kelime için, sadece o muğlak kelimeye ait bir eğitim verisi (".arff" uzantılı dosya) oluşturulmalıdır.

Küresel bazlı inceleme'de kullanılacak olan ".arff" uzantılı dosyada kullanılacak olan öznitelik vektörünün oluşturulmasına ait algoritma Şekil 6.3'de verilmiştir.

1. Eğitim veri setinden bir metin dosyası al.
2. Metin dosyasındaki cümleler içinde muğlak kelime var mı diye kontrol et.
3. Eğer metin içinde muğlak kelime var ise o metin içinde geçen tüm kelimeleri öznitelik vektörüne kaydet.
4. Muğlak kelime yok ise bir sonraki metne geç.
5. Eğitim setindeki tüm metinler bitinceye kadar işleme devam et.

Şekil 6.4: Küresel öznitelik vektörünün oluşturulmasına ait algoritma.

Bu yöntemde eğitim kümesinde yer alan bütün metinlerde bulunan kelimeler öznitelik vektörünün bir elemanı olarak kaydedilmiştir. Cümle Bazlı İnceleme metodunda olduğu gibi, öznitelik vektörüne ait değerler, bir metin içinde o kelimenin kaç defa geçtiğini göstermektedir.

Öznitelik vektörünün toplam öznitelik sayısı eğitim kümesinde yer alan metinlerde geçen bütün kelimeler olarak belirlenmiştir. 15.989 tane kelime eğitim kümemizde yer almaktadır, bu nedenle öznitelik vektörü 15.989 tane eleman içermektedir. ".arff" dosyasındaki heri bir veri satırı muğlak kelimenin geçtiği metni ifade etmektedir.

Tablo 6.5: Küresel bazlı öğrenme için örnek eğitim seti tablosu.

Metin No	
1	ABCDE. DGAE. BJCEH. GDBF.
2	BEAFBH. CBHJA. BHDJG.
3	GEHDA. DGHICB.

Yukarıdaki tabloda her harf bir kelimeyi temsil etmektedir. "A" kelimesinin muğlak kelime olduğunu varsayarsak, Tablo 6.5'de bu kelimeyi içeren ve farklı sayıda cümlelerden oluşan metinler gösterilmektedir. Bu metinler için hazırlanacak olan ".arff" uzantılı dosya içeriği Tablo 6.6'da verilmiştir.

Tablo 6.6: Attribute relation file format dosyası için öznitelik tablosu.

attribute	a	real
attribute	b	real
attribute	c	real
attribute	d	real
attribute	e	real
attribute	f	real
attribute	g	real
attribute	h	real
attribute	j	real

Tablo 6.6'da verilen her bir öznitelik (attribute), Tablo 6.5'de bulunan metinlerdeki kelimelerdir.

Tablo 6.7: Attribute relation file format dosyası için veri tablosu.

a	b	c	d	e	f	g	h	j
2	3	2	3	3	1	2	1	1
2	3	1	1	1	1	1	3	2
1	1	1	2	1	0	2	2	0

Tablo 6.7'de verilen her bir satır içinde muğlak kelimenin bulunduğu metni temsil etmektedir. Her bir değer ise o metin içinde kelimelerin kaçar defa geçtiğini göstermektedir.

7. DENEYLER

7.1. Veri Kümesi Hazırlama

Çalışmamızda eğitim kümesinde kullanılmak üzere oluşturduğumuz sözlük, internette bulunan haber ve forum sayfalarında yer alan metinlerden oluşmaktadır. Bu metinlerin bir kısmı Türkçe diline ait bir sözlük oluşturma amacıyla Haşim Sak, Tunga Güngör ve Murat Saraçlar tarafından yapılan bir çalışma sonucu elde edilmiştir [43]. Ayrıca daha önce metin madenciliği alanında yüksek lisans çalışması yapan bir arkadarkadaşımızın hazırladığı sözlükten de faydalanılmıştır [44]. Hazırladığımız sözlükte yer alan tüm kelimeler herhangi bir ön işlemden geçmemiş, ham halde saklanan kelimelerdir. İçerisinde hem düzgün Türkçe yazılmış metinler, hem de günlük konuşma dilinden örnekler bulunmaktadır.

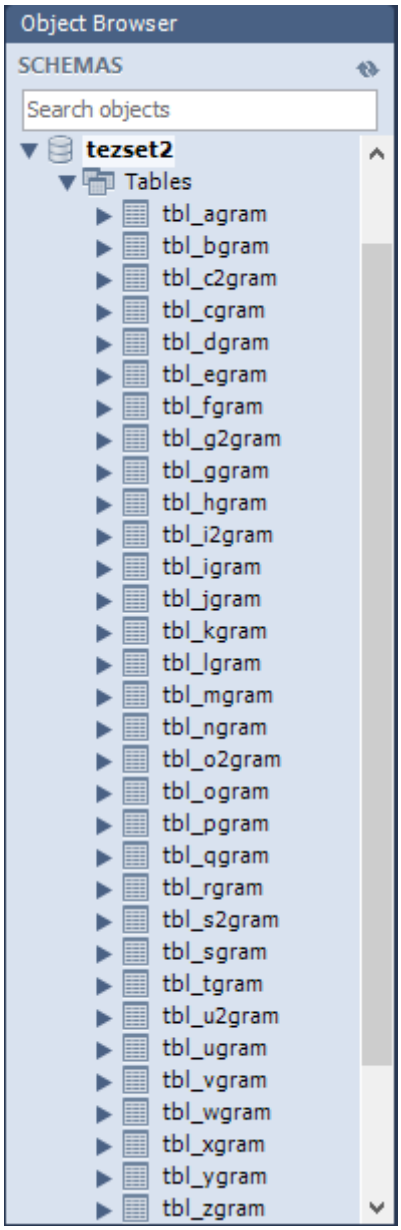
Sistemin hızlı bir şekilde kullanabilmesi için cümleler formatında bulunan metinlerin 3'lü parçalara ayrılarak (tri-gram karşılaştırması yapabilmek için) saklanması amaçlanmıştır. Çalışmamızın başlarında veri tabanı kullanılmadan sadece bellek üzerinde işlemler yapılması düşünülmüştür. Kelimeler arası ilişkilerin 3 boyutlu bir dizi halinde saklanıp, ihtiyaç halinde bu diziden birim zamanda istenilen kelimeye ait bilgilerin çekilmesi amaçlanmıştır. Kelimeler arası ilişkileri dizilerde tutmak, her ne kadar bellek kaybı yaşatmış olsa da, zaman yönünden büyük avantaj sağlayacağı düşünülmekteydi. Ancak artan kelime sayılarının neden olduğu aşırı bellek kullanımı problemi nedeniyle bu yöntemden vazgeçilmek zorunda kalınmıştır.

Farklı bir yöntem denenerek kelime ilişkilerinin ayar dosyalarında olduğu gibi ".ini" dosyası formatında saklanması düşünülmüştür. Elimizdeki eğitim kümesi metinlerinden kelime ilişkileri çıkartılmış ve bu ilişkilerden Şekil 7.1'de gösterildiği gibi ayar dosyası formatında yazdırılmıştır. Her bir kelime için bir grup oluşturulmuş ve o kelime ile birlikte kullanılan diğer tüm kelimeler teker teker bu ayar dosyasında saklanmıştır. Ancak bu ayar dosyalarının da sistem ilk çalışmaya başladığında belleğe yüklenmesi gerekmektedir. Fazla miktardaki kelime sayısı için aşırı bellek kullanımına neden olduğundan bu yöntemden de vazgeçilmek durumunda kalınmıştır.

Şekil 7.1: Ayar dosyası formatında saklanan kelime ilişkileri.

Kelime ilişkilerini saklayabilmemiz için yapılan denemeler neticesinde, bu ilişkilerin saklanması için veri tabanı kullanılmasının en iyi çözüm olacağı sonucuna varılmıştır. Oluşturulan sözlük MySql veri tabanında oluşturulan tablolarda saklanmıştır. Ücretsiz erişilebilmesi ve sistemin programlama dili olan Java ile uyumluluğunun kolay olması nedeniyle, "MySql [45]" veri tabanı seçimi yapılmıştır.

Veri tabanı tabloları her kelimenin ilk harfi bir tablo adını belirtecek şekilde oluşturulmuştur. Veri Tabanı tabloları Şekil 7.2'de gösterilmiştir. Örneğin; armağan, atölye, adam, ayar.. kelimeleri "a" harfi ile başladığı için bu kelimelere ait Ngram ilişkileri "tbl_agram" tablosunda; bakan, bebek, bulmak, baskı... kelimeleri "b" harfi ile başladığı için bu kelimelere ait Ngram ilişkileri "tbl_bgram" tablosunda saklanmaktadır. Bu şekilde toplam 32 tane tablo oluşturulmuştur.



Şekil 7.2: Veri tabanında saklanan tablolar.

Veri tabanına kayıt edimiş olan veri ve alanlara ait örnek Tablo 7.1‘de gösterilmektedir.

Tablo 7.1: Veri tabanı alanları tablosu.

Id	pre_word	word	post_word	ngram_count
38	Bu	ayarın	Bozuk	1
67	Kuyuya	attığı	Taş	27
85	Parçası	Avrupada	Ve	2
140	Kaza	ağabeyi	Kadar	3
178	Bir	av	Bekliyor	2
201	Filmde	anlatılmak	Istenenlerin	1

Oluşturulan veri tabanında toplam 331.850 tane farklı kelime (word) bulunmaktadır. Bu kelimelerden 1.206.514 tane farklı N-Gram ilişkisi ortaya çıkmıştır. Toplam 8.342.117 tane N-Gram ilişkisi içeren kelime 3’lüleri (pre_word + word + post_word) bulunmuştur.

Bu haber metinlerinin İngilizce alfabesi harflerini içerecek şekilde yeniden oluşturulması, Denklem (1) kullanılarak sağlanmıştır. Bu yöntem sayesinde deney sonuçlarının otomatik olarak karşılaştırılabildiği ve insan faktörü içermeyen bir gerçeklik tabanı elde edilmektedir.

7.2. Eğitim ve Test Kümesi Hazırlama

Makina Öğrenmesi algoritmalarında kullandığımız eğitim veri kümesinin nasıl hazırlanıldığından Bölüm 7.1’de bahsedilmiştir.

Muğlak kelime tahminine, metin konusunun etkisinin olup olmadığını araştırmak için hazırladığımız metinler 4 ana konu başlığı (Sanat, Spor, Siyaset, Ekonomi) altında kategorilere ayrılmıştır. Bu kategoriler haber sitelerinde sıklıkla geçen haberlerin, konu başlıklarına göre belirlenmiştir. Daha fazla veya daha az sayıda konu başlığının belirlenmesi eğitim kümesi üzerinde ezberleme veya öğrenememe problemlerine yol açmaktadır. Eğitim kümesi kümesi içinde yer alan metin dosyalarının sayısı aşağıda verilmiştir.

- Ekonomi : 498
- Sanat : 491
- Siyaset : 520
- Spor : 515

Eğitim kümemiz içinde yer alan toplam 2024 metin dosyası içinde toplam 436.707 kelime bulunmaktadır.

Test veri kümesinde sadece internetten bulunan haber sitelerinden [46], [48] alınan ve yukarıda yazdığımız 4 ana başlığa ait toplam 200 metin dosyası bulunmaktadır.

7.3. Deneysel Kurulumlar

Yapılan çalışma Java programlama dilinde kodlanmıştır. Veri tabanı olarak MySql veri tabanı kullanılmıştır. Geliştirme ortamı olarak Eclipse uygulama geliştirme aracı tercih edilmiştir.

Sistem 4 çekirdekli ve 10 GB RAM'e sahip bir bilgisayar üzerinde çalıştırılmıştır. İşletim sistemi olarak Windows 7 (64 bit) tercih edilmiştir.

7.3.1. Zemberek

Türkçe dili üzerinde çalışan ve açık kaynak koda sahip Zemberek 0 Doğal Dil İşleme kütüphanesidir. Zemberek Türkçe dilbilgisine ait bir çok fonksiyona sahip bir araçtır. Kelimelerin morfolojik ve sözdizimsel yapılarını bize bilgi vermektedir. Kelime Ön İşleme sırasında,

- kelimenin kök ve ek'lerine ayrılması,
- kelimenin Türkçe olup olmadığına karar verilmesi,
- kelimenin hecelerine ayrılması,
- kelimenin çözümlenmesi..

fonksiyonları Zemberek yardımı ile yapılmıştır.

Yaptığımız çalışmalarda Zemberek'in verdiği öneri ve sonuçların doğru olduğu kabul edilmiştir.

7.3.2. Weka

Makina öğrenmesi algoritmalarının uygulanabilmesi için Weka [49] Makina Öğrenmesi Aracı kullanılmıştır. Açık kaynak koda sahip ve bir çok DDİ çalışmalarında başarı ile kullanılan bir araçtır.

Weka içindeki algortimalara Java programlama dili yardımıyla dışarıdan erişim mümkündür. Eğitim ve test kümesi olarak ARFF formatında hazırlanmış dosya yapısını kullanmaktadır. Bu nedenle elimizdeki verilerin, Weka'nın okuyup işleyebileceği formatta düzenlenmesi gerekmektedir. Weka, ".arff" uzantılı dosya formatı ile verileri kullanabilmektedir. ".arff" uzantılı dosya içinde başlık (attribute) ve veri (data) yer almaktadır. Öznitelik makina öğrenmesinde kullanılan her bir kelime, veri ise bu özniteliklerin nicel değerleridir. Her bir özniteliğe karşı gelen bir veri bulunmaktadır ve bunlar aynı sırada dosyada dizilmektedir.

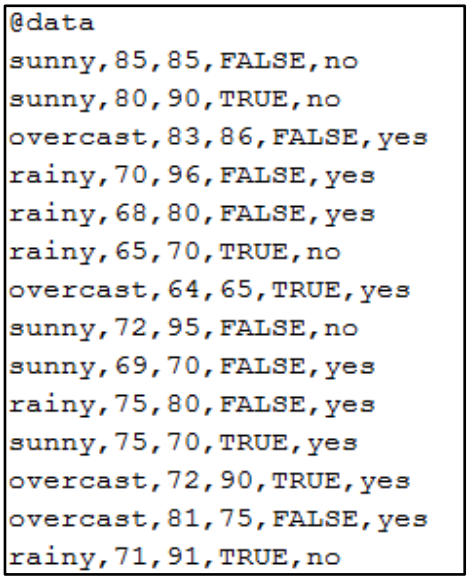
Başlık bölümü öznitelikleri ve bu özniteliklerin tiplerini gösterir. (Şekil 7.3)

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
```

Şekil 7.3: Attribute relation file format dosyası başlık örneği.

Veri bölümü özniteliklere ait bilgileri içermektedir. (Şekil 7.4)



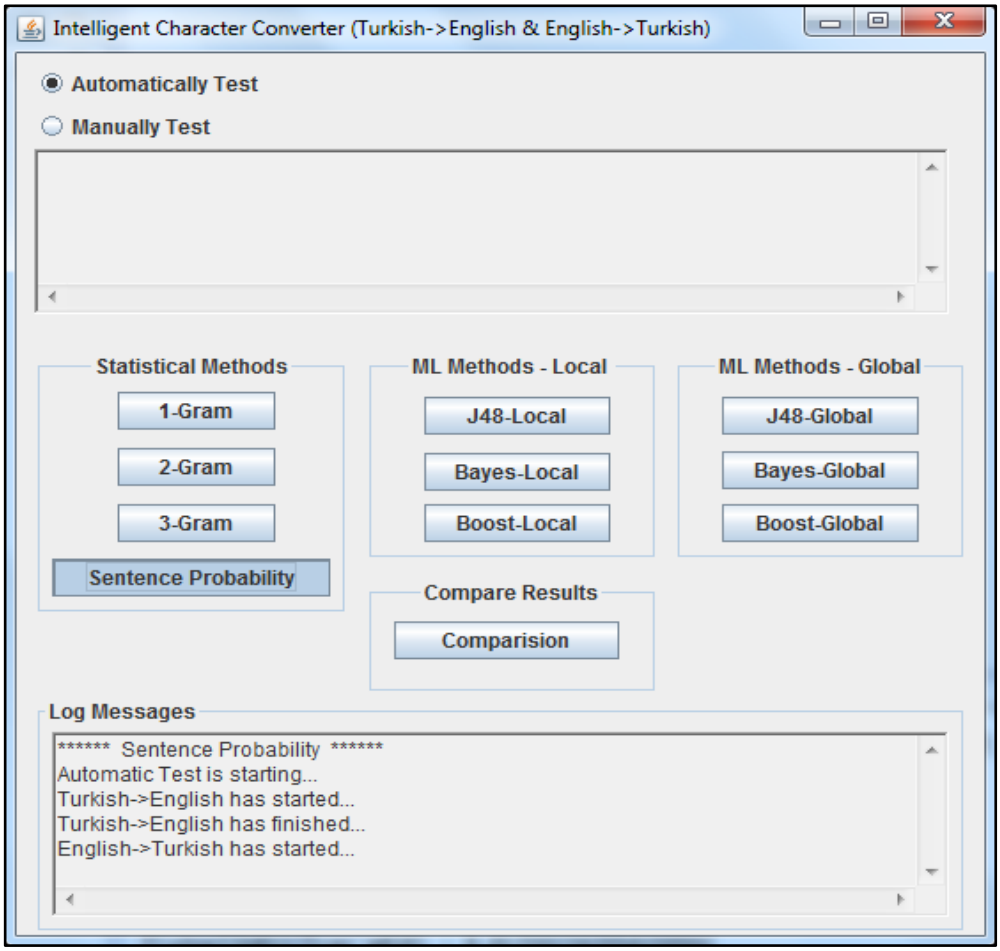
Şekil 7.4: Attribute relation file format dosyası veri örneği.

Ayrıca çalışmamızın değişik aşamalarında farklı açık kaynak kodlu kütüphanelerden faydalanılmıştır. (düzgün log'lama yapabilmek için ini4j Log kütüphanesi gibi)

7.4. Program Arayüzü Oluşturulması

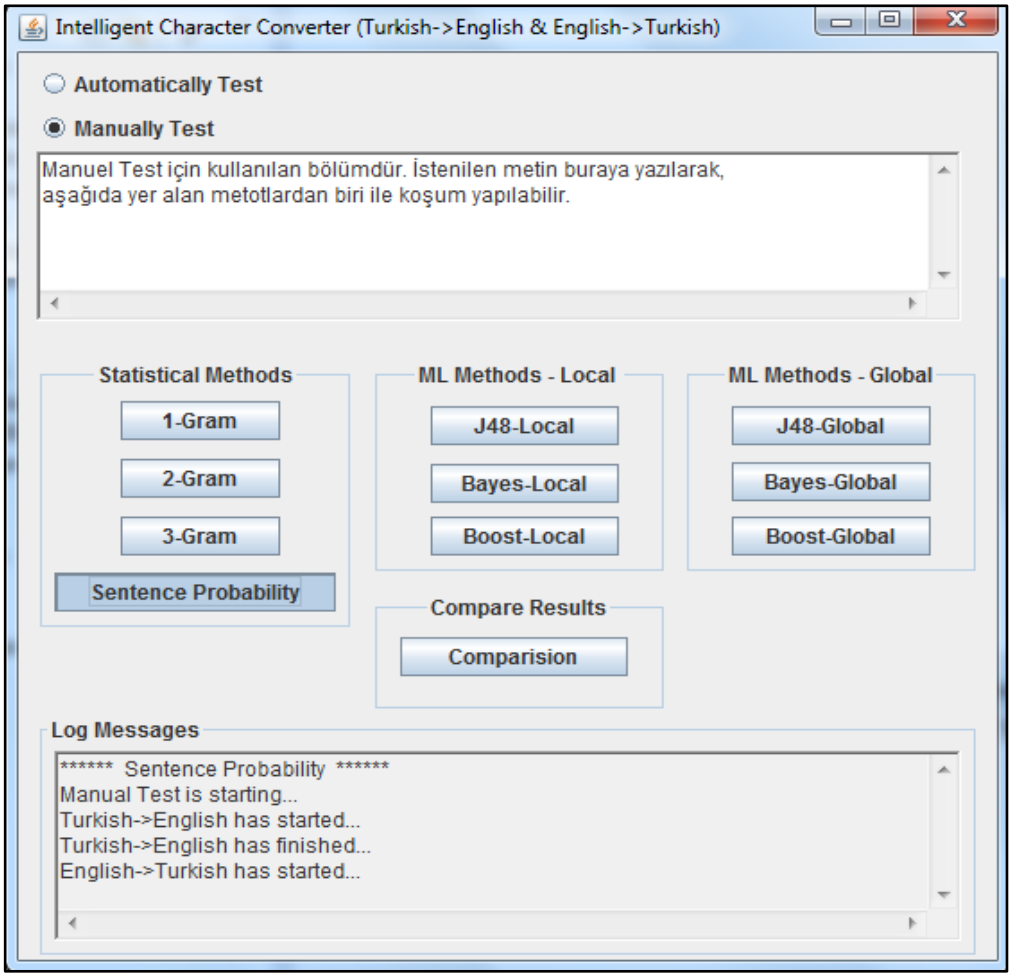
Testlerin daha hızlı ve kolay şekilde çalıştırılabilmesi için bir arayüz tasarlanmıştır. Bu arayüzde "Automatically Test" ve "Manually Test" olmak üzere 2 seçenek konulmuştur.

"Automatically Test" seçeneği seçilirse istenen İstatistiksel veya Makina Öğrenmesi Metoduna göre test veri kümesinde bulunan tüm metinler sırasıyla çalıştırılmaktadır.(Şekil 7.5)



Şekil 7.5: Otomatik test.

Ayrıca istenilen metin veya cümlenin anında test edilebilmesini sağlayan "Manuel Test" seçeneği arayüze eklenmiştir. Yazılan bir metnin istenilen İstatistiksel veya Makina Öğrenmesi Metoduna göre test edilmesi sağlanmıştır. (Şekil 7.6)



Şekil 7.6: Manuel test.

7.5. Deney Sonuçları

7.5.1. Sonuç Karşılaştırma

Test kümesinde yüzlerce metin olduğu için bu test veri setine ait sonuç kümesinin teker teker göz ile karşılaştırılması çok uzun zaman almaktadır. Yapılan çeviri işleminde aşağıda yazan değerlerin otomatik hesaplanması gerekmektedir.

- Metinde toplam kaç kelime bulunduğu,
- Metinde bulunan SAB problemi içeren kelime sayısı,
- Metindeki Muğlak Kelime Oranı(%),
- Metinde Algoritma Uygulanamayan Kelime Sayısı,
- Metinde Algoritma Uygulanan Kelime Sayısı,

- Metinde Toplam Yanlış Bulunan Kelime Sayısı,
- Metinde Yanlış Bulunan SAB problemine neden olan kelime sayısı,
- Çalıştırılan Algoritmanın Doğruluk Oranı,
- Algoritmanın Çalışma Zamanı Değeri,
- Algoritmanın Kullanıdığı Bellek Miktarı,
- Algoritmanın Kullanıdığı CPU Miktarı.

Bu nedenle giriş metni ile program çıktısı olan sonuç metinlerinin otomatik karşılaştırılması yapılmıştır. Yapılan Arayüz'de "Comparision" tuşu ile yukarıdaki maddeler bir sütun olarak, excel dosyasına otomatik yazdırılmaktadır. Tablo 7.2'de örnek sonuç tablosu sütunları gösterilmektedir.

Tablo 7.2: Çalışma zamanı, CPU ve memory kullanımı sonuç tablosu.

Dosya Adı	Çalışma Zamanı(msec)	CPU Kullanımı(%)	Memory Kullanımı(KB)
haber8.txt	27420	26,56909752	84544
haber9.txt	34616	26,27356911	23040
haber10.txt	78495	27,11807823	1027072
haber11.txt	89248	26,39403915	49408
haber12.txt	207615	25,72402763	60224
haber13.txt	42003	25,56190872	17664
haber14.txt	42088	26,27939224	30912
haber15.txt	158019	25,92958069	26432
haber16.txt	57329	25,52441788	1024
haber17.txt	181353	26,19758606	68480

7.5.2. Vektör Uzunluklarının Normalizasyonu

Yerel Bazlı İnceleme için hazırlanan Eğitim kümesinde her bir muğlak kelime için ayrı bir eğitim dosyası tutulmaktadır. Bu eğitim kümesi içindeki ".arff" uzantısına sahip dosyalar, farklı sayıda özniteliklere sahiptirler. Test işlemi sırasında her bir muğlak kelimeye ait öznitelik vektörü büyüklükleri belli bir oran ile eşitlenerek normalize edilmektedir. Özitelik kümesine ait vektör uzunluklarının normalize işlemi sonucunda doğruluk oranında artış kaydedilmiştir. Tablo 7.3 ve Tablo 7.4'te, normalizasyon işleminin 10 haber metni için sağladığı doğruluk oranı artışı gösterilmektedir.

Tablo 7.3: Normalize yapılarak koşturulan test.

Dosya Adı	Alg. Doğruluk Oranı(%)	Çalışma Zamanı(msec)	CPU Kullanımı(%)	Memory Kullanımı(KB)
haber2501.txt	77,27272727	396696	26,53852654	848000
haber2502.txt	90	96823	26,00876045	779840
haber2503.txt	91,66666667	98652	25,86654472	4160
haber2504.txt	100	3787	28,83567619	9088
haber2505.txt	75	100415	25,9833374	26432
haber2506.txt	83,92857143	242260	26,12949181	41216
haber2507.txt	88,88888889	213823	26,09160805	72256
haber2508.txt	96,55172414	381516	25,35881042	45632
haber2509.txt	100	89559	25,44015312	17536
haber2510.txt	100	98715	25,77102089	58432
ORTALAMA	90,33085784			

Tablo 7.4: Normalize yapılmadan koşturulan test.

Dosya Adı	Alg. Doğruluk Oranı(%)	Çalışma Zamanı(msec)	CPU Kullanımı(%)	Memory Kullanımı(KB)
haber2501.txt	75,75757576	365204	26,49894905	1541312
haber2502.txt	100	103580	25,62991905	175680
haber2503.txt	75	101065	25,85095596	176512
haber2504.txt	100	4055	29,62287712	-8960
haber2505.txt	75	113453	25,67863846	167040
haber2506.txt	85,71428571	241506	26,04312134	178560
haber2507.txt	83,33333333	212086	26,00004387	88896
haber2508.txt	96,55172414	372467	25,65449142	65728
haber2509.txt	100	87473	25,58314514	17536
haber2510.txt	90,90909091	95893	25,8583622	23296
ORTALAMA	88,22660099			

7.5.3. Weka Parametrelerinde Değişiklik

Makina Öğrenmesi yöntemleri için Weka'da algoritmalar standart parametreler ile kulanılmaktaydı. Weka'da bulunan makina öğrenmesi algoritmaları parametreler değiştirilerek 10 metinlik test kümesinde teker teker koşturulmuştur. Parametre değişikliklerinin kayda değer bir doğruluk oranı artışı sağlamadığı gibi çalışma sürelerinde de artışa sebeb olduğu gözlemlenmiştir. Bu işleme ait sonuçlar Tablo 7.5 ve Tablo 7.6'da gösterilmektedir. Bu nedenle Weka'da bulunan algoritmalardaki, standart parametreler ile testler çalıştırılmıştır.

Tablo 7.5: Parametre değişikliği sonrası yapılan test.

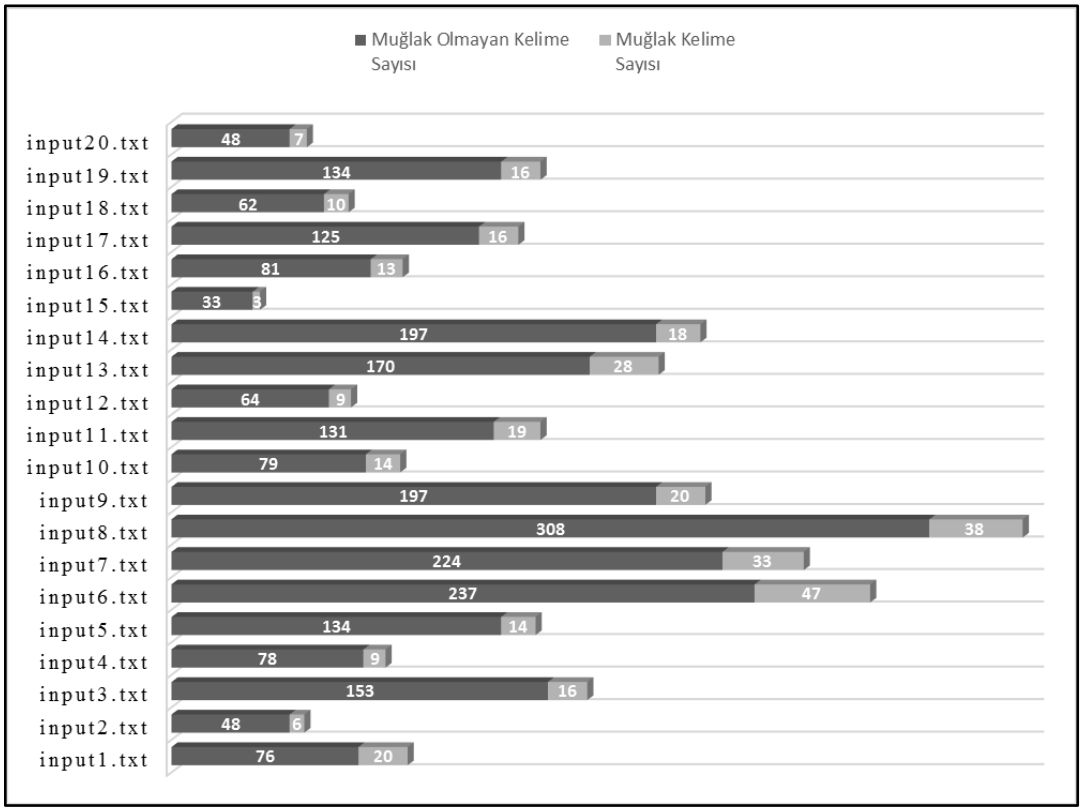
Dosya Adı	Alg. Doğruluk Oranı(%)	Çalışma Zamanı(msec)	CPU Kullanımı(%)	Memory Kullanımı(KB)
haber8.txt	57,14285714	27420	26,56909752	84544
haber9.txt	37,5	34616	26,27356911	23040
haber10.txt	57,14285714	78495	27,11807823	1027072
haber11.txt	92,85714286	89248	26,39403915	49408
haber12.txt	71,42857143	207615	25,72402763	60224
haber13.txt	75	42003	25,56190872	17664
haber14.txt	80	42088	26,27939224	30912
haber15.txt	61,11111111	158019	25,92958069	26432
haber16.txt	100	57329	25,52441788	1024
haber17.txt	67,74193548	181353	26,19758606	68480
ORTALAMA	69,99244752	91818,6		

Tablo 7.6: Standart parametreler ile yapılan test.

Dosya Adı	Alg. Doğruluk Oranı(%)	Çalışma Zamanı(msec)	CPU Kullanımı(%)	Memory Kullanımı(KB)
haber8.txt	57,14285714	27618	26,47746658	62400
haber9.txt	37,5	34730	26,27716446	7616
haber10.txt	57,14285714	77765	27,29240227	943552
haber11.txt	92,85714286	81316	26,57058525	51456
haber12.txt	71,42857143	195584	25,80289078	114880
haber13.txt	75	41236	25,81037903	56512
haber14.txt	80	41442	26,43494415	89344
haber15.txt	61,11111111	158193	26,11308098	216384
haber16.txt	100	58125	25,87268257	2944
haber17.txt	67,74193548	181259	26,1251049	147072
ORTALAMA	69,99244752	89726,8		

7.5.4. Test Seti

Kullandığımız sözlüğün içinde 4.000 civarında SAB problemi içerebilecek kelime bulunmuştur. Bu kelimeler, sözlükteki kelimelerin önceden işlenmesi sonucu SAB problemine neden olup olmadığının tespit edilmesiyle belirlenmiştir. Bulunan muğlak kelimeler için istatistik bilgileri üretilmiş ve makina öğrenmesi algoritmaları eğitilmiştir. Her bir muğlak kelime için makina öğrenmesi algoritmalarının kullanabileceği bir ".arff" uzantılı dosya oluşturulmuştur. Hem cümle bazlı inceleme, hem de metin bazlı inceleme için iki ayrı şekilde veriler oluşturulmuştur. Yani algoritmaların herhangi bir muğlak kelime için cümle bazlı incelemede kullanacağı ".arff" dosyası ile metin bazlı incelemede kullanacağı ".arff" dosyası farklıdır.



Şekil 7.7: Test metinlerindeki muğlak kelime sayısı grafiği.

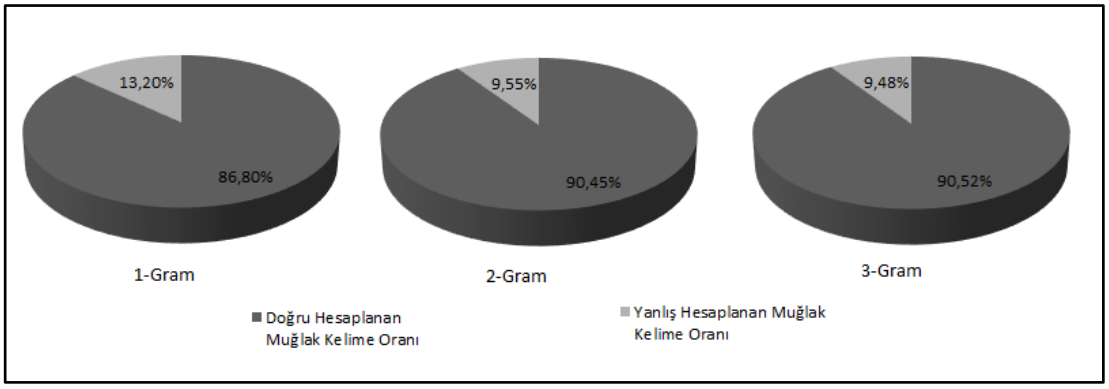
Test kümemizde yer alan 356 muğlak kelime (SAB problemi içeren kelimeler) için hem istatistiksel, hem de makina öğrenmesi temelli yöntemler denenmiştir. Geliştirilen uygulama muğlak kelimeler içinden uygun seçimi, istenilen yönteme göre otomatik olarak yapmaktadır. Kullanılan test veri kümesine ait bilgiler Şekil 7.7'de verilmiştir.

Tablo 7.7: Yöntem doğruluk tablosu.

No	Yöntem	Doğruluk Yüzdesi
1	1-Gram	86,80%
2	2-Gram	90,45%
3	3-Gram	90,52%
4	AdaBoost Yerel	66,66%
5	AdaBoost Küresel	92,12%
6	J48 Yerel	80,18%
7	J48 Küresel	93,33%
8	Naive Bayes Yerel	92,85%
9	Naive Bayes Küresel	90,24%
10	Markov Zinciri	90,02%

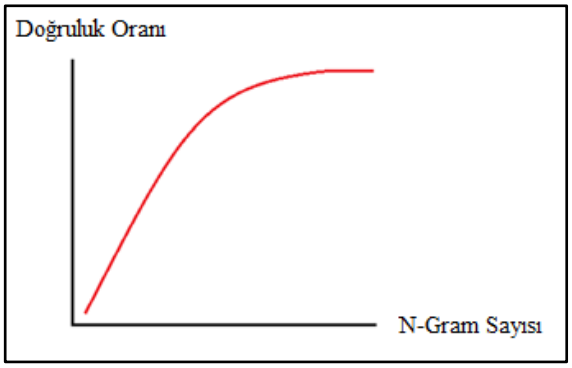
Tablo 7.7'de verilen sonuçlarda uygulanan yöntemlerin doğruluk oranları yüzdesel olarak verilmiştir. Hem makina öğrenmesi tabanlı yöntemler ile istatistiksel yöntemler; hem de yerel bazlı inceleme ile küresel bazlı inceleme ayrı ayrı karşılaştırılmıştır. Yapılan çalışmalar sonucunda;

- Makina öğrenmesi algoritmaları kullanılarak yapılan çevirilerin istatiksel yöntemlere göre,
- Küresel bazlı incelemelerin yerel bazlı incelemelere göre daha başarılı sonuçlar verdiği gözlenmektedir.



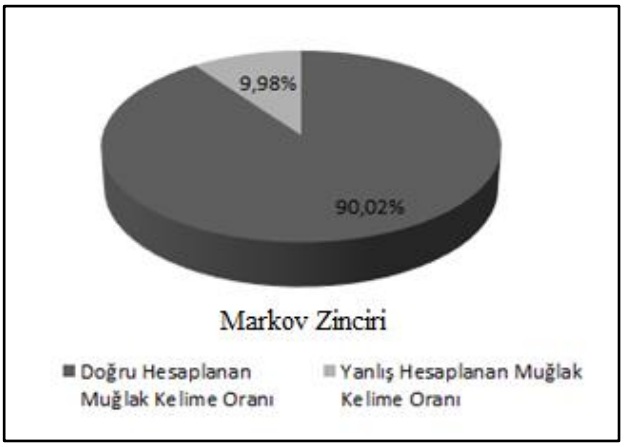
Şekil 7.8: İstatistiksel yöntemler doğruluk oranı grafiği.

Şekil 7.8'de bulunan grafikler incelenecek olursa, istatistiksel yöntemlerde 2-Gram ve 3-Gram'ın, 1-Gram'a göre daha başarılı olduğu ancak; 3-Gram'ın 2-Gram'dan belirgin bir şekilde ayrılmadığı gözlemlenmiştir. Bunun temel sebebinin kullanılan sözlüğün, 3-Gram ve üstü yöntemler için yeterli büyüklükte olmamasından kaynaklanmaktadır. N-Gram sayısı ile Doğruluk Oranı arasındaki ilişkinin Şekil 7.9'da gösterildiği gibi olduğu gözlenmiştir.



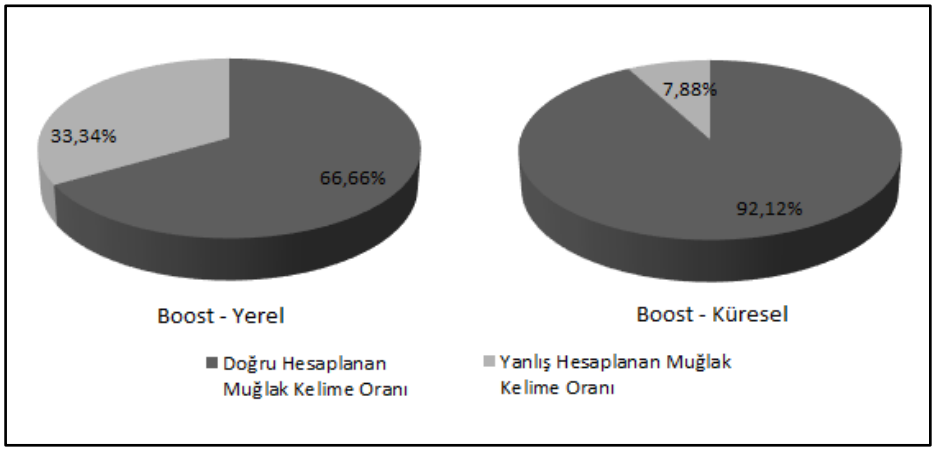
Şekil 7.9 : NGram sayısı - Doğruluk oranı eğrisi.

Ayrıca sözlük büyüklüğü arttıkça, N-Gram doğruluk oranlarında da artış olacağı beklenmektedir.

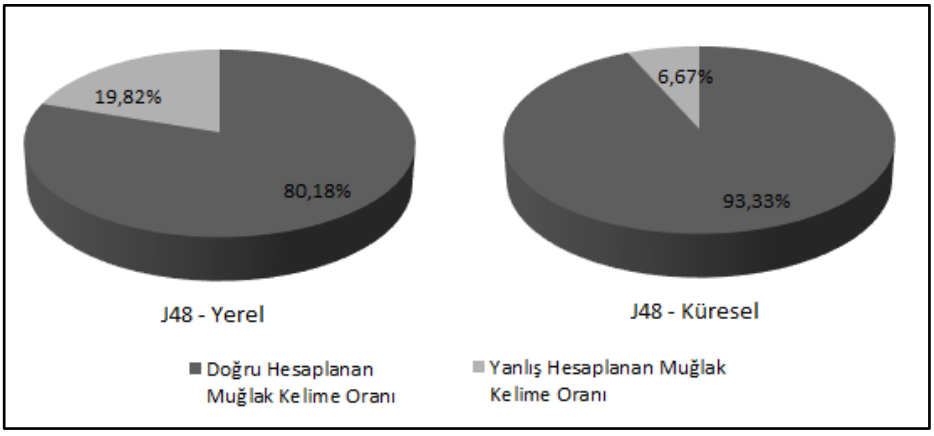


Şekil 7.10: Markov zinciri doğruluk oranı grafiği.

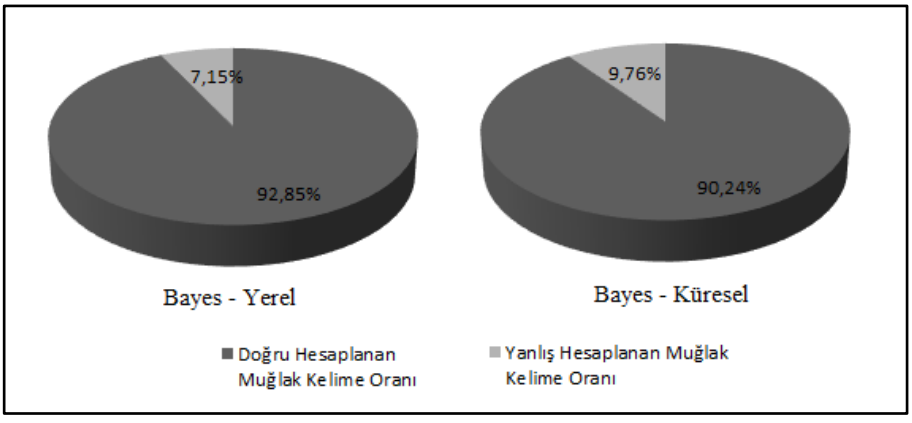
Markov Zinciri ile yapılan çalışmada doğruluk oranının Şekil 7.10'da olduğu gibi görülmektedir. Bu yöntem diğer denenen yöntemlere göre daha başarılı sonuçlar vermektedir. Bunun nedeninin yerel bazlı inceleme ile aynı olduğunu söylemek mümkündür. Sadece yerel bilgiye bakarak kelime hakkında çıkarım yapmak bize yüksek başarı oranı vermemektedir.



Şekil 7.11: AdaBoost makine öğrenmesi doğruluk oranı grafiği.



Şekil 7.12: J48 ağacı makine öğrenmesi doğruluk oranı grafiği.



Şekil 7.13: Naive Bayes makine öğrenmesi doğruluk oranı grafiği.

Makina öğrenmesi yöntemlerinin, genel olarak istatistiksel yöntemlere göre daha iyi sonuçlar verdiği gözlemlenmiştir. Bunun temel sebebi istatistiksel yöntemlerin, yerel bilgi kullanmasından (sadece komşuluk bilgisi) kaynaklanmaktadır.

Şekil 7.11, Şekil 7.12 ve Şekil 7.13 incelenecek olunursa benzer eğilimin yerel ve küresel makina öğrenmesi teknikleri arasında da gözlemlendiği anlaşılmaktadır. Küresel bilgi'den faydalanarak yapılan çalışmaların, bağlam ve içerik açısından metinle ilgili daha çok bilgi verdiği görülmüştür.

Cümle bazlı yapılan inceleme'de muğlak kelime ile aynı cümlede bulunan kelimeler incelenmektedir ancak bu kelimeler sınırlı sayıda olmaktadır. Türkçe'de bir cümle en fazla 10-12 kelime'den oluşmaktadır. 10-12 kelimeye bakarak, bu kelimeler içinde birden fazla sayıda muğlak kelime olabilir, cümlenin içerik bilgisinin anlaşılması zorlaşmaktadır.

Metin bazlı inceleme'de her metinde ortalama 15-25 cümle yer almaktadır. Her cümlede ortalama 8-10 kelime olduğunu varsayarsak, metin ile alakalı yaklaşık 150 civarında kelimeden faydalanmaktayız. İçerik bilgisinin anlaşılmasında elimizde ne kadar fazla bilgi olursa yapacağımız çıkarım o derece kuvvetlı olmaktadır. Metin bazlı inceleme'de daha fazla sayıda kelimeden yapılan muğlak kelime tercihi bizi daha yüksek doğruluk oranına ulaştırmaktadır.

Tablo 7.8: Muğlak kelime içeren cümle tablosu.

Cümle No	Cümle
1	beklenen *olumlu* tepkiyi alamadi (olumlu, ölümlü)
2	yaptigi *is* hakkinda (is, iş, ıs)
3	Filmi elestiren en onemli *isim* (isim, işim, ısım)
4	konusunda bir *kisim* prensiplerde (kişim, kısım, kışım)
5	her *turlu* sorunlarina (turlu, türlü)
6	satin almak uzere *Nisan* ayi (nisan, nişan)
7	yeni filmi *kis* uykusunda (kis, kış, kıs)
8	yeni filmi belli *oldu* (oldu, öldü)

Tablo 7.8'de muğlaklık oluşturan kelimelerin yer aldığı örnek cümleler verilmiştir. Muğlak kelimeler koyu harflerle gösterilmiştir. Muğlak kelimeler için

alternatifler parantez içerisinde verilmektedir. Yapılan çalışma muğlak kelime alternatifleri içinden doğru olan seçimi bulmaya çalışmaktadır.

Tablo 7.9: Cümle yöntem başarı tablosu.

Cümle No	1	2	3	B.Y	B.K	J.Y	J.K	N.Y	N.K
1	+	+	+	-	+	+	+	-	+
2	-	-	-	-	+	+	+	+	+
3	-	-	-	-	+	-	+	+	+
4	-	-	-	+	+	-	-	-	+
5	-	+	+	+	+	-	-	+	+
6	-	+	+	-	+	+	+	+	+

Tablo 7.8'de ise Tablo 7.9'da verilen cümlelerin uyguladığımız yöntemlere göre doğru çevrilip çevrilmediğine ilişkin bilgiler yer almaktadır. Tablo 7.9'da yer alan kısaltmalar:

- 1: 1-Gram,
- 2: 2-Gram,
- 3: 3-Gram,
- B.Y: AdaBoost Yerel,
- B.K: AdaBoost Küresel,
- J.Y: J48 Yerel,
- J.K: J48 Küresel,
- N.Y: Naive Bayes Yerel,
- N.K: Naive Bayes Küresel

yöntemlerini ifade etmektedir.

Tablo 7.9'da görüleceği gibi, istatistiksel yöntemlerde uygulanan NGram sayısı arttıkça çeviri doğruluğunda artış görülmektedir. Makina öğrenmesi yöntemlerinde küresel yöntemler, yerel yöntemlere göre daha başarılı sonuçlar vermektedir. Tanımladığımız SAB probleminin çözümünde, istatistiksel yöntemlerden bir sonuç elde edilemediği zaman makina öğrenmesi yöntemlerine başvurmamız, başarı oranımızı arttırmaktadır.

7.5.5. Çarpraz Geçerleme Testi

Testlerimizi zenginleştirmek amacıyla 3000 haber metninden oluşan yeni bir küme seçilmiştir. Bu kümedeki 2500 metin Eğitim Seti; 500 metin Test Seti olmak üzere iki parçaya ayrılmıştır. Bu seçim Set1 olarak adlandırılmıştır. Set1 eğitim kümesinde bulunan 2500 metin içinde, SAB problemi içeren kelimeler ayrı ayrı bulunarak eğitim dosyaları hazırlanmıştır. Ayrıca bu eğitim kümesine ait metinler üzerinde N-Gram ilişkilerinin bulunduğu "tezset1" isimli yeni veri tabanı tabloları oluşturulmuştur. Bütün İstatistiksel ve Makina Öğrenmesi Yöntemleri için Set1 kümesine ait test metinleri (toplam 500 tane) teker teker çalıştırılmıştır.

Seçilen 3000 haber metni bu sefer farklı bir 2500 metin Eğitim Seti, 500 metin Test Seti olacak şekilde ikiye ayrılmıştır. Yukarıdaki paragrafta anlatılan tüm işlemler Set2 olarak adlandırılan bu küme için de ayrı ayrı yapılmıştır.

Tablo 7.10: Test kümeleri veri tabanı analizi tablosu.

Test Kümeleri	Toplam Farklı Kelime Sayısı*	N-Gram İlişkisi Sayısı**	Toplam N-Gram Sayısı***
TezSet1	59.915	328.167	498.152
TezSet2	60.405	335.635	510.874
Tez_Complex	335.528	1.210.394	8.376.716
* Veri tabanında bulunan farklı kelime sayısıdır. ** Veri tabanında bulunan 3'lü N-Gram İlişkileri sayısıdır. ** Veri tabanında bulunan 3'lü N-Gram İlişkilerinin toplam sayısıdır.			

Tablo 7.10'da hazırlanan eğitim kümelerine ait n-gram ilişkilerine ait değerler yer almaktadır. Eğitim kümesinde bulunan haber metinlerinin büyüklüklerine göre veri tabanında bulunan n-gram ilişkileri sayıları değişiklik göstermektedir. Bu değişiklikler hem İstatistiksel Yöntemlerin doğruluk oranlarına, hem de bu eğitim kümelerinden oluşturulan ".arff" uzantılı dosyaların içeriğine doğrudan etki etmektedir.

Set1 ve Set2 kümeleri ile yapılan testler sonucunda Tablo 7.11'de yer alan sonuçlar elde edilmiştir. Tabloda yer alan sonuçlar incelendiği zaman İstatistiksel Yöntemlerin, Makina Öğrenmesi Yöntemlerine göre genelde daha iyi sonuçlar verdiği gözlemlenmiştir. Ayrıca Küresel Bazlı İnceleme'nin, Yerel Bazlı İnceleme'den daha iyi sonuçlar verdiği gözlemlenmiştir.

Tablo 7.11: Çarpraz geçerleme testi sonuçları.

Yöntem	Set1	Set2	Average
1-Gram	84,82%	85,50%	85,16%
2-Gram	90,91%	91,44%	91,18%
3-Gram	91,00%	92,58%	91,79%
J48-Local	70,20%	71,73%	70,97%
J48-Global	71,21%	72,50%	71,86%
Bayes-Local	85,94%	86,90%	86,42%
Bayes-Global	86,77%	88,36%	87,57%
Boost-Local	70,10%	73,88%	71,99%
Boost-Global	83,93%	85,38%	84,66%
Chain Rule	90,41%	91,65%	91,03%

Önceki yaptığımız testlerde Makina Öğrenmesi Yöntemlerinin, İstatistiksel Yöntemlere göre daha iyi sonuçlar verdiğini göstermiştik. Ancak Çarpraz Geçerleme Testi sonuçlarında benzer sonuçların elde edilememesinin nedeni; Set1 ve Set2 olarak hazırlanan kümelerden kaynaklanmaktadır. Set1 ve Set2'ye ait eğitim kümeleri rastgele seçildiği için, sistemin düzgün bir şekilde eğitilememesine yol açmıştır. Eğitim işlemi sırasında seçilecek olan küme, SAB problemi içeren kelimeler için yeterli büyüklüğe erişememiştir. SAB problemi içeren keliemelere ait eğitim kümesinin yetersiz olması Makina Öğrenmesi yöntemlerinin İstatistiksel Yöntemlere göre biraz daha düşük çıkmasına neden olmuştur. Buna rağmen Küresel Bazlı İnceleme her iki test setinde de Yerel Bazlı İnceleme'den daha iyi sonuçlar vermiştir.

8. SONUÇ

"İngilizce alfabesi ile yazılmış olan Türkçe metinlerin, Türkçe alfabesi içeren şekilde yeniden yazılması" olarak ifade ettiğimiz SAB probleminin çözümü, günümüzde ihtiyaç duyulan önemli bir konudur. Bu problemin çözümü için kullandığımız istatistiksel ve makina öğrenmesi teknikleri, muğlak kelimeleri içeren metin üzerinde analiz edilmiştir.

Küresel bilgiler kullanıldığı zaman, SAB probleminin başarı yüzdesinin arttığı görülmüştür. Bu durum elimizdeki problemin tipik bir anlam ve bağlam problemi olduğunun göstergesidir. Daha önceden yapılmış çalışmalara paralel olarak anlam ve bağlam problemlerinde, makina öğrenmesi yöntemlerinin, istatistiksel yöntemlere göre daha fazla bilgi verdiği anlaşılmıştır. Makina öğrenmesi tekniklerinin başarı oranının seçilen eğitim kümesine bağlı olduğu, eğitim kümesi ne kadar kapsamlı ise başarı oranının o kadar yüksek olduğu görülmüştür.

Hem istatistiksel yöntemler hem de makina öğrenmesi yöntemleri kullanılan veri seti ile doğrudan ilişkilidir. Kullanılan veri seti ne kadar büyük olursa başarımın oranının arttığı görülmüştür. Makina öğrenmesi yöntemleri her ne kadar çalışma süresi yönünden dezavantajlı olsa da, gittikçe artan doğruluk oranına sahiptir.

Tez sonuçları oldukça geniş bir test kümesi altında elde edilmiştir. Test kümelerinde 250'den fazla kelime içeren cümle örnekleri bulunduğu gibi, tek kelimelik cümleler de bulunmaktadır. Test yelpazesinin geniş tutulması her türlü örneğin incelenmesi fırsatını doğurmuştur.

Yapılan çalışmada Türkçe dili için tanımlanan bir SAB problemine çözüm getirilmiş, getirilen bu çözümün ileride yapılacak olan çalışmalara ilham kaynağı olacağı beklenmektedir. Sonradaki yıllarda yapılacak olan Türkçe alfabesindeki harflerin kullanım sıklığının belirlenmesi, Türkçe dilinin bağlama olan bağlılığının incelenmesi, Türkçe dili için hatalı yazıların düzeltilmesinde akıllı çözümler bulunması... yeni çalışma alanlarına öncülük edecektir.

KAYNAKLAR

[1] Web 1, (2013), https://code.google.com/p/zemberek/, (Erişim Tarihi : 12/06/2013)

[2] Web 2, (2012), http://turkce-karakter.appspot.com/, (Erişim Tarihi : 12/06/2013)

[3] Ay A., (2009), "İngilizce klavye kullanılarak yazılan Türkçe metinlerin Türkçe karakterler içeren metinler haline çevrilmesi", Lisans Bitirme Tezi, Gebze Yüksek Teknoloji Enstitüsü.

[4] Oflazer K., Bozşahin H. C., (1997), "Türkçe Doğal Dil İşleme", Dil Devriminden Bu Yana Türkçenin Görünümü Dergisi, 7(51).

[5] Oflazer K., Bozşahin H. C., (1994), "Natural Language Processing in Turkish", Third Turkish Symposium on Artificial Intelligence and Artificial Neural Networks, Ankara, Turkey, 22-24 Haziran.

[6] Quillian, M. R., (1969), "The teachable language comprehender: a simulation program and theory of language", Communications of the ACM, 12(8), 459-476.

[7] Sosyal E., Çiçekli N. B., Baykal, N., (2009), "Radyoloji Raporları için Türkçe Bilgi Çıkarım Sistemi", VI. Ulusal Tıp Bilişimi Kongresi, 304-312, İstanbul, Türkiye, 12-15 Kasım.

[8] Amasyalı M. F., Diri B., Türkoğlu, F., (2006), "Farklı özellik vektörleri ile Türkçe dokümanların yazarlarının belirlenmesi", The Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks, Muğla, Turkey, 21-24 Haziran.

[9] Lewis D. D., (1992), "Speech and natural language", 1st Edition, Morgan Kaufmann.

[10] Guvenir H. A., Oflazer K., (1995), "Using A Corpus For Teaching Turkish Morphology", In Proceeding of the Seventh Twente Workshop on Language Technology, 28-40, Enschede, The Netherlands, 16-17 June.

[11] Miller G. A., (1995), "WordNet: a lexical database for English", Communications of the ACM, 38(11), 39-41.

[12] Tufis D., Cristea D., Stamou, S., (2004), "BalkaNet: Aims, methods, results and perspectives. A general overview", Romanian Journal of Information science and technology, 7(1-2), 9-43.

[13] Bilgin O., Çetinoğlu Ö., Oflazer K., (2004), "Building a wordnet for Turkish", Romanian Journal of Information Science and Technology, 7(1-2), 163-172.

[14] Navigli R., (2009), “Word sense disambiguation: A survey”, ACM Computing Surveys (CSUR), 41(2), 10.

[15] Yuret D., (2007), “KU: Word sense disambiguation by substitution”, 4th International Workshop on Semantic Evaluations, 207-213, Prague, Czech Republic, 23-24 June.

[16] Yngve V. H., (1955), “Machine translation of languages”, 1st Edition, John Wiley & Sons.

[17] Miller G. A., (1995), “WordNet: a lexical database for English”, Communications of the ACM, 38(11), 39-41.

[18] Amasyalı M. F., (2005), “Türkçe Wordnet’in Otomatik Oluşturulması”, 13. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 119, Kayseri,Türkiye, 16 - 18 Mayıs.

[19] Amasyalı M. F., İnak B., Ersen M. Z., (2010), “Türkçe Hayat Bilgisi Veri Tabanının Oluşturulması”, XII. Akademik Bilişim Konferansı, 38, Muğla, Türkiye, 10 - 12 Şubat.

[20] Mihalcea R., Chklovski T., Kilgarriff A., (2004), “The Senseval-3 English lexical sample task”, Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, 25-28, Barcelona, Spain, 25–26 July.

[21] Stamou S., Oflazer K., Pala K., Christoudoulakis D., Cristea D., Tufis D., Grigoriadou M., (2002), “BalkaNet: A multilingual semantic network for the Balkan languages”, International Wordnet Conference, 21-25, Mysore, India, 21-25 January.

[22] Stevenson M., Wilks Y., (2003), The Oxford Handbook of Comp. Linguistics, 1st Edition, Oxford University Press.

[23] Altan Z., Orhan Z., (2005), “Anlam Belirsizliği İçeren Türkçe Sözcüklerin Hesaplamalı Dilbilim Uygulamalarıyla Belirginleştirilmesi”, XX. Ulusal Dilbilim Kurultayı, İstanbul, Türkiye, 12-13 Mayıs.

[24] Aydın Ö., Kılıçaslan Y., (2010), “Tümevarımlı Mantık Programlama ile Türkçe için Kelime Anlamı Belirginleştirme Uygulaması”, XII. Akademik Bilişim Konferansı Bildirileri, 138, Muğla, Türkiye, 10-12 Şubat.

[25] Silahtaroğlu G., Demircan F., (2013)"Çeviri Yazılımlarında Sözcüklerin Bağlam İçindeki Anlamını Algılamaya Yönelik Bir Öneri.", XIV. Akademik Bilişim Konferansı Bildirileri, 67, Antalya, Türkiye, 23-25 Ocak.

[26] Orhan Z., Altan Z., (2005), “Makine Öğrenme Algoritmalarıyla Türkçe Sözcük Anlamı Açıklaştırma”, X. Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Ulusal Kongresi, 344-348, İstanbul, Türkiye, 22 - 25 Eylül.

[27] Orhun M., (2011), “Uygurca ile Türkçe Birleşik Sözcüklerin Karşılaştırılması”, XIII. Akademik Bilişim Konferansı, 36, Malatya, Türkiye, 2-4 Şubat.

[28] Aydın Ö., Tüysüz M. A. A., Kılıçaslan Y., (2007), “Türkçe İçin Bir Kelime Anlamı Belirginleştirme Uygulaması”, XII. Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Ulusal Kongresi, Eskişehir, Türkiye, 14-18 Kasım.

[29] Altan Z., Yanık E., (2001), “Kelime Anlamlarının İstatistiksel Çıkarımı için Metin Örneklerinin İşlenmesi”, İstanbul Üniversitesi Elektrik& Elektronik Dergisi, 1(2), 287-295.

[30] Sevilgen F. E., (2010), “Hesaplama Teorisi”, Lisans Dersi, Gebze Yüksek Teknoloji Enstitüsü.

[31] Web 3, (2013), http://ngram.google.com, (Erişim Tarihi : 12/06/2013)

[32] Larkey L. S., Croft W. B., (1996), “Combining classifiers in text categorization”, 19th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, 289-297, Zurich, Switzerland, 18 – 22 August.

[33] Lewis D. D., Gale W. A., (1994), “A sequential algorithm for training text classifiers”, 17th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, 3-12, Dublin, Ireland, 03 – 06 July.

[34] McCallum A., Nigam K., (1998), “A comparison of event models for naive bayes text classification”, AAAI-98 workshop on learning for text categorization, 41-48, Wisconsin, USA, 26–30 July.

[35] Freund Y., Schapire R. E., (1997), "A decision-theoretic generalization of on-line learning and an application to boosting.", Journal of computer and system sciences, 55(1), 371-561.

[36] Schapire R. E., (1999), "A brief introduction to boosting.", 16. International Joint Conference on Artificial Intelligence, 1401, Stockholm, Sweden, 31 July – 6 August.

[37] Freund Y., Schapire R., Abe N., (1999), "A short introduction to boosting.", Journal-Japanese Society For Artificial Intelligence, 771-780, Japan, 01 September.

[38] Gökmen M., Kurt B., Kahraman F., Çapar A., (2007), “Çok Amaçlı Gürbüz Yüz Tanıma”, İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü, Tübitak Projesi, Proje No:104E121.

[39] Schapire R. E., Singer, Y., (2000), "BoosTexter: A boosting-based system for text categorization.", Machine Learning Springer, 7(1-2), 163-172.

[40] Meir R., Rätsch G., (2003), “Advanced Lectures on Machine Learning”, 1st Edition, Springer Berlin Heidelberg.

[41] Schapire R. E., (1990), “The strength of weak learnability”, Machine learning, 5(2), 197-227.

[42] Quinlan J. R., (1993), “C4.5: programs for machine learning”, 1st Edition, Morgan Kaufmann.

[43] Ruggieri S., (2002), "Efficient C4.5 classification algorithm", IEEE Transactions on Knowledge and Data Engineering, 14(2), 438-444.

[44] Sak H., Güngör T., Saraçlar M., (2008), “In Advances in natural language processing”, 1st Edition, Springer Berlin Heidelberg.

[45] Karadağ A., (2011), “Web’den Hazırlanan Türkçe Sözlük”, Yüksek Lisans Tez Çalışması, Gebze Yüksek Teknoloji Enstitüsü.

[46] Web 4, (2013), www.mysql.com, (Erişim Tarihi : 12/06/2013)

[47] Web 5, (2013), http://www.milliyet.com.tr/, (Erişim Tarihi : 12/06/2013)

[48] Web 6, (2013), http://www.zaman.com.tr/, (Erişim Tarihi : 12/06/2013)

[49] Web 7, (2013), http://www.cs.waikato.ac.nz/ml/weka, (Erişim Tarihi : 12/06/2013)

ÖZGEÇMİŞ

1987 yılında Giresun'da doğan Burak Çağrı OKUR ilk, orta ve lise öğrenimini Giresun'da tamamladı. 2010 yılında Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümünden mezun oldu. Aynı yıl Gebze Yüksek Teknoloji Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başladı. 2010 yılında TÜBİTAK – BİLGEM'de Araştırmacı olarak başladığı görevini halen sürdürmektedir.

EKLER

EK A: Terimler

İngilizce - Türkçe

A	
Analysis	Analiz
Artificial	Yapay
Artificial Intelligence	Yapay Zeka
Asciifier	Ascii Karaktere Dönüştürme
Assumption	Varsayım
Attribute	Öznitelik
B	
Bayesian Network	Bayes Ağ Modeli
Binary	İkili
Bioinformatics	Biyoenformatik
C	
Classification	Sınıflandırma
Computer	Bilgisayar
Constant Time	Birim Zaman
Content	İçerik
Content and Thematic Analysis	İçerik ve Tematik Analiz
Corpus	Sözlük
Corpus Based	Derlem Tabanlı
Cross Validation	Çarpraz Doğrulama
D	
Data	Veri
Deasciifier	Ascii Karaktere Geri Alma
Denoising	Temizleme
Deterministic	Belirlenimci
Deterministic Finite State Transducer	Belirlenimci Sonlu Durum Dönüştürücüsü
E	
F	
Feature	Özellik, Öznitelik
Final	Sonuç
Finite	Sonlu
G	

Gain	Kazanç
Gramatical	Gramer
Gramatical Analysis	Gramer Çözümlemesi
Graph	Çizelge
H	
Human	İnsan
Human-Computer Interaction	İnsan-Makine Etkileşimi
I	
Implementation	Gerçekleştirme
Information	Bilgi
Information Gain	Bilgi Kazancı
Information Retrieval	Bilgi Geri Kazanımı
Initial	Başlangıç
Intelligence	Zeka
Interaction	Etkileşim
Iteration	Döngü
J	
K	
Knowledge Based	Bilgi Tabanlı
L	
Language	Dil
Lexicon	Sözlük
Linguistics	Dil Bilim
M	
Machine	Makina
Machine Translation	Makine Çevirisi
Morphology	Morfoloji
Multinominal	Çok terimli
N	
Natural	Doğal
Natural Languase Processing	Doğal Dil İşleme
Network	Ağ Modeli
Nondeterministic	Belirlenimci Olmayan
Non-deterministic Finite State Transducer	Belirlenimci Olmayan Sonlu Durum Dönüştürücüsü
O	
Overfitting	Aşırı Öğrenme
Overfitting	
P	
POS tagging	Sözcük etiketleme
Processing	İşleme

Pseudocode	Sözde Kod
Q	
R	
Retrieval	Kazanım
Retrieval	
S	
Semantic	Semantik
Semantic Web	Semantik Web
Sentax	Sentaks/Söz Dizimi
Speech	Ses
Speech Processing	Ses İşleme
State	Durum
Stemming	Kök Bulma
Stopper	Durak
Stopper Words	Durak Kelimeleri
Supervised	Öngörmeli
T	
Tagging	Etiketleme
Text	Metin
Text Processing	Metin İşleme
Thematic	Tematik
Token	Ayıraç
Training	Eğitim
Transducer	Dönüştürücü
Translation	Çeviri
Tuples	Değişken Grubu
U	
Underfitting	Öğrenememe
Unsupervised	Öngörmesiz
V	
Validation	Doğrulama
W	
Word	Sözcük
Word	Kelime
Word Sense Disambiguation	Sözcük Anlamı Belirleme
X	
Y	
Z	

Türkçe – İngilizce

A	
Ağ Modeli	Network
Analiz	Analysis
Ascii Karaktere Dönüştürme	Asciifier
Ascii Karaktere Geri Alma	Deasciifier
Aşırı Öğrenme	Overfitting
Ayıraç	Token
B	
Başlangıç	Initial
Bayes Ağ Modeli	Bayesian Network
Belirlenimci	Deterministic
Belirlenimci Olmayan	Nondeterministic
Belirlenimci Olmayan Sonlu Durum Dönüştürücüsü	Non-deterministic Finite State Transducer
Belirlenimci Sonlu Durum Dönüştürücüsü	Deterministic Finite State Transducer
Bilgi	Information
Bilgi Geri Kazanımı	Information Retrieval
Bilgi Kazancı	Information Gain
Bilgi Tabanlı	Knowledge Based
Bilgisayar	Computer
Biyoenformatik	Bioinformatics
C	
Ç	
Çarpraz Geçerleme	Cross-Validation
Çeviri	Translation
Çizelge	Graph
Çok terimli	Multinominal
D	
Değişken Grubu	Tuples
Derlem Tabanlı	Corpus Based
Dil	Language
Dil Bilim	Linguistics
Doğal	Natural
Doğal Dil İşleme	Natural Languase Processing
Doğrulama	Validation
Döngü	Iteration
Dönüştürücü	Transducer
Durak	Stopper
Durak Kelimeleri	Stopper Words
Durum	State
E	
Eğitim	Training
Etiketleme	Tagging
Etkileşim	Interaction

F
G

Gerçekleştirme	Implementation
Gramer	Gramatical
Gramer Çözümlemesi	Gramatical Analysis

Ğ
H
I
İ

İçerik	Content
İçerik ve Tematik Analiz	Content and Thematic Analysis
İkili	Binary
İnsan	Human
İnsan-Makine Etkileşimi	Human-Computer Interaction
İşleme	Processing

J
K

Kazanç	Gain
Kazanım	Retrieval
Kelime	Word
Kök Bulma	Stemming

L
M

Makina	Machine
Makine Çevirisi	Machine Translation
Metin	Text
Metin İşleme	Text Processing
Morfoloji	Morphology

N
O
Ö

Öğrenememe	Underfitting
Öngörmeli	Supervised
Öngörmesiz	Unsupervised
Özellik, Öznitelik	Feature
Öznitelik	Attribute

P
R
S

Semantik	Semantic
Semantik Web	Semantic Web
Sentaks/Söz Dizimi	Sentax
Ses	Speech
Ses İşleme	Speech Processing
Sınıflandırma	Classification
Sonlu	Finite
Sonuç	Final
Sözcük	Word
Sözcük Anlamı Belirleme	Word Sense Disambiguation
Sözcük etiketleme	POS tagging

Sözde Kod	Pseudocode
Sözlük	Corpus
Sözlük	Lexicon
Ş	
T	
Tematik	Thematic
Temizleme	Denoising
U	
Ü	
V	
Varsayım	Assumption
Veri	Data
Y	
Yapay	Artificial
Yapay Zeka	Artificial Intelligence
Z	
Zeka	Intelligence