T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARDIŞIK ŞARTLI RASTGELE ALANLARLA SEKANS ETİKETLEME

METİN BİLGİN

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
YRD.DOÇ.DR.MEHMET FATİH AMASYALI

İSTANBUL, 2015

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ARDIŞIK ŞARTLI RASTGELE ALANLARLA SEKANS ETİKETLEME

Metin BİLGİN tarafından hazırlanan tez çalışması 18.11.2015 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd.Doç.Dr.Mehmet Fatih AMASYALI

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd.Doç.Dr.Mehmet Fatih AMASYALI

Yıldız Teknik Üniversitesi

Doç.Dr.Banu DİRİ

Yıldız Teknik Üniversitesi

Yrd.Doç.Dr. Murat Can GANİZ

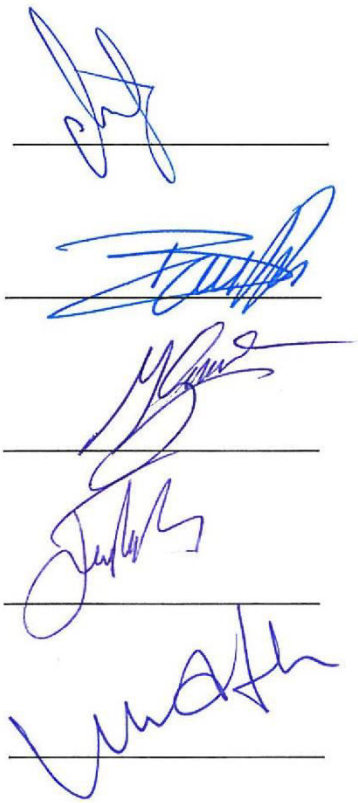
Marmara Üniversitesi

Doç.Dr.Fethullah KARABİBER

Yıldız Teknik Üniversitesi

Doç.Dr.Murat GÖK

Yalova Üniversitesi



ÖNSÖZ

Doktora çalışmalarımın her aşamasında sıkılmadan her sorumu cevaplandıran, hem mesleki hemde insani olarak bana çok şey öğreten ve bu tezin hazırlanması çok büyük emeği olan değerli Tez Danışmanım ve Hocam Yrd.Doç.Dr. Mehmet Fatih AMASYALI'ya çok ama çok teşekkür ederim. Sizin yardımlarınız olmasaydı başaramazdım.

Doktora çalışmalarının yoğunluğundan dolayı çoğu zaman ihmal etmeme rağmen bana büyük manevi destekte bulunan ve her zaman yanımda olduklarını her şekilde gösteren eşim Besim'e, kızım Erva'ya ve oğlum Emir Yusuf'a çok ama çok teşekkür ederim.

Üzerimde büyük emekleri bulunan annem Ayşe Bilgin'e, babam Ümmet Bilgin'e, ablalarım Nadiye ve Hatice Bilgin'e ve abim Üzeyir Bilgin'e çok teşekkür ederim.

Tezimin başlangıç noktasından bitişine kadar çok değerli katkıları ve yönlendirmeleri ile ufkumu açan değerli tez izleme komisyonu üyeleri Doç.Dr. Banu Diri ve Yrd.Doç.Dr. Murat Can Ganiz'e çok teşekkür ederim.

Özellikle Bağlılık Ayrıştırması için kullanılan veri setinin temininde destek veren Yrd.Doç.Dr. Gülşen Cebiroğlu Eryiğit'e çok teşekkür ederim.

Tüm süreç boyunca bana desteklerinden ötürü dostlarım Mustafa Balcı'ya, İsa Yıltırak'a, Veysel Gündoğdu'ya, Muhammet Uçar'a, Hasan Kalkan'a ve Kadir Oğuz Baştürk'e çok teşekkür ederim.

Kasım, 2015

Metin BİLGİN

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

E	Eleman
$\propto$	Öğrenme Katsayısı
ε	Hata Oranı
d	Türev (Difransiyel)
H	Hessian Matris
γ	İki noktanın türevlerinin farkı
δ	İki noktanın farkı
N	İsim (Noun)
V	Fiil (Verb)
P	Hassasiyet (Precision)
R	Çağırma (Recall)
F	F-Ölçütü (F-Measure)
ID	Tanımlama Kodu(Satır No)
DEP	Bağlantı Türü
AS_U	Doğru ID'ye Bağlanma Oranı
AS_L	Doğru ID'ye Doğru Etiketle bağlanma Oranı
R_0	Zero Rule

KISALTMA LİSTESİ

ABD	Amerika Birleşik Devletleri
AHDS	Arts and Humanities Data Sevices
AI	Yapay Zeka (Artificial Intelligent)
arff	Attribute Relationship File Format
AT	Otomatik Çeviri (Automatic Translation)
AVIATOR	Analysis of Verbal Interaction and Automated Text Retrieval
BFGS	Broyden-Fletcher-Goldfarb-Shanno Optimizasyon Yöntemi
BNC	British National Corpus
BT	Bellek Tabanlı
CANCODE	Sözlü Dilden Örnekler İçeren Derlem
CRF	Şartlı Rastgele Alanlar (Condition Random Fields)
CSAE	Corpus of Spoken Amerian English
DDA	Doğal Dil Anlama
DDÜ	Doğal Dil Üretme
DFB	Davidon-Fletcher-Powel Optimizasyon Yöntemi
DİE	Devlet İstatistik Enstitüsü
DP	Bağlılık Ayrıştırması (Dependency Parsing)
Fac	Tesis (Facility)
F-CRF	Faktöriyel CRF Yöntemi
GPE	Jeo-Politik (Geo-Politic)
HKUST	Bilgisayar Bilimi Derlemi
HMM	Saklı Markov Model (Hidden Markov Model)
HTML	Hypertext Markup Language
IBM	Uluslararası İş Makineleri (International Business Machines)
IE	Bilgi Çıkarımı (Information Extraction)
ILASA	International Institute for Applied Systems Analysis
KMÖ	Klasik Makine Öğrenmesi
L-BFGS	Sınırlı Bellekli BFGS Yöntemi
L-CRF	Lineer CRF Yöntemi
LDOCE	Longman Dictionary of Contemporary English
LLC	London-Lund derlemi
LOB	Lancaster-Oslo/Bergen derlemi
Loc	Yer (Location)

MB	Mega Bayt
MEMM	Maksimum Entropi Markov Model (Maximum Entropy Markov Model)
METU	Orta Doğu Teknik Üniversitesi (Middle East Technical University)
MI	Mutual Information
Misc	Diğer (Miscellaneous)
ML	Makine Öğrenmesi (Machine Learning)
MOS	Makinece Okunur Sözlük
MS-DOS	MicroSoft Disk Operating System
NATO	Kuzey Atlantik Antlaşması Örgütü (North Atlantic Treaty Organization)
NER	Varlık İsmi Tanımlama (Name Entity Recognition)
NLP	Doğal Dil İşleme (Natural Language Process)
OM	Fikir Madenciliği (Opining Mining)
Org	Organizasyon (Organization)
PDF	Portable Document Format
Per	Kişi (Person)
POS	Part of Speech
RL	Öğelerine Ayırma (Role Labeling)
SDD	Sonlu Durumlu Dönüştürücüler
SEC	Spoken English Corpus
SEU	Survey of English Usage
SGML	Standard Generalized Markup Language
SL	Sekans Etiketleme (Sequence Labeling)
SP	Yüzeysel Parçalama (Shallow Parsing)
SPSS	Statistical Package for the Social Sciences
SVM	Destek Vektör Makineleri (Support Vector Machine)
TBMM	Türkiye Büyük Millet Meclisi
TE	Tablodan Çıkarım (Table Extraction)
TurCo	Çebi ve Varlıkların oluşturduğu ve işaretlenmemiş derlem
WEKA	Waikato Environment for Knowledge Analysis
WSJ	Wall Street Journal
XML	Genişletilebilir İşaretleme Dili (Extensible Markup Language)
YSA	Yapay Sinir Ağı

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

ARDIŞIK ŞARTLI RASTGELE ALANLARLA SEKANS ETİKETLEME

Metin BİLGİN

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Yrd.Doç.Dr.Mehmet Fatih AMASYALI

Sekans etiketleme bir giriş dizisine karşılık bir çıkış dizisinin üretimidir. Giriş ve çıkış dizisinin içeriklerine göre doğal dil işlemenin birçok konusu (varlık isim tanıma, makine çevirisi, morfolojik analiz, cümleleri öğelerine ayırma vb.) sekans etiketleme olarak tanımlanabilir.

Cümle analizi ve cümleden bir anlam çıkarılması, doğal dil işlemenin ana konularından biridir. Eğer ilgili cümlenin söylemek istediği gerçek anlam çıkartılabilirse bu cümle makineler tarafından eyleme dönüştürülebilir, bir dilden başka bir dile çeviri yapılabilir ya da cümleden duygusal bir anlam çıkartılması sağlanabilir.

Bağlılık ayrıştırması, bir cümle içerisindeki sözcükler arasındaki ilişkilerin ve ilişki türlerinin belirlenmesidir ve bir cümlenin anlamsal analizinin yapılabilmesi için şarttır. Bağlılık ayrıştırması sekans etiketleme problemi olarak tanımlandığında iki çıkış dizisinin (ilişki türü, ilişkili kelime) birden üretilmesi gerekmektedir.

Bir cümlenin çözümlenmesi, ilgili dilin cümle yapısına bağlıdır. Türkçe, bitişken ve cümle içi öğe dizilişleri serbest bir dildir. Bu nedenle diğer dil ailelerine göre çözümlemesi daha zor bir dildir. Literatürde Türkçe ile ilgili yapılan çalışmalar mevcut olmasına rağmen ağırlıklı olarak İngilizce için gerçekleştirilmiş çalışmalar bulunmaktadır.

Türkçe için yapılan çalışmalarda, Destek Vektör Makineleri (Support Vector Machine) tabanlı bir yapı kullanan Malt Parser ile belirli bir doğruluk oranlarına erişilmiştir. Diğer diller için yapılan çalışmalar incelendiğinde bu başarının artırılması için yeni hipotezler üretilmesi ve bunların denenmesi gereği açıktır.

Bizim önerimiz, özellikle sekans etiketleme problemlerinin çözümünde sıklıkla kullanılan Şartlı Rastgele Alanların bağlılık ayrıştırması problemi içinde kullanılabilir olduğudur. Ancak Şartlı Rastgele Alanlar tek çıkış üreten bir yöntemdir. Bu zorluğu aşabilmek için iki çıkışlı (Bağlılık Türü ve Bağlanılan Kelime) bir problem olan Bağlılık Ayrıştırması iki parçaya bölünerek çözülmüştür. Ardından elde edilen sonuçlar birleştirilerek sistemin çıktısı olarak verilmiştir.

Türkçe için gerçekleştirilen çalışma sonuçları ile literatürdeki sonuçlar karşılaştırıldığında daha yüksek bir başarı oranına ulaşıldığı görülmüştür. Türkçe dışındaki İsveç Dili, Danimarka Dili, Hollanda Dili ve Portekiz dili için de önerdiğimiz yöntem denenmiştir. İlişki türünü belirlemede literatürdeki çalışmaların başarıları aşılmıştır. İlişkili kelimeyi belirleme de ise daha kötü bir performans sergilenmiştir. Buna, Türkçe dışındaki bu dillerin cümle içi bağlılık yapılarının çok daha değişken olmasının sebep olduğu görülmüştür.

Gelecek çalışma olarak geliştirilen yöntemin diğer dillerdeki performansını arttırmak için daha dinamik bir yapının geliştirilmesi düşünülmektedir.

Anahtar Kelimeler: Sekans Etiketleme (SL), Bağlılık Ayrıştırması (DP), Şartlı Rastgele Alanlar (CRF), Makine Öğrenmesi (ML), Doğal Dil İşleme (NLP)

ABSTRACT

SEQUENCE LABELING WITH STACKED CONDITIONAL RANDOM FIELDS

Metin BİLGİN

Department of Computer Engineering

PhD. Thesis

Adviser: Asst.Prof. Mehmet Fatih AMASYALI

Sequence labeling is the production of an output sequence in return for an input sequence. Many issues (name entity recognition, machine translation, morphological analysis, resolving the sentence into its elements, etc.) of natural language processing based on the contents of the input and output sequence can be defined as sequence labeling.

Sentence analysis and making out the meaning of a sentence are one of the main topics of natural language processing. If real meaning requiring saying the relevant sentence can draw, this sentence can convert into action by machines, translate from one language to other language or enable to get the emotive meaning of the sentence.

Dependency Parsing determines the relationships and types of relationships between words within a sentence and is essential to the semantic analysis of a sentence. When attachment discrimination is defined as the problem of sequence labeling, two-output sequence (relationship type, related word) should be generated together.

Analysis of a sentence depends on the sentence structure of the relevant language. Turkish is an agglutinative language and free-intrasentence arrangements of element. Therefore, it is a language difficult to analyze compared to other language families. Although some studies exist in the literature about Turkish, there have mainly been studies on English.

Studies performed for Turkish were achieved a certain degree of accuracy with Malt Parser using a Support Vector Machines-based structure. When examining the studies performed for other languages, it is clear that new hypothesis should develop and test in order to increase this success.

Our suggestion is that conditional random fields used often especially in solving the sequence labeling problems can be available in dependency parsing problem. However, the conditional random fields is a method of producing a single output. In order to overcome this challenge, dependency parsing being a problem with dual outputs (attachment type and connected word) is resolved by dividing into two parts. After, the results is provided as an output of the system by combining.

Compared the studies carried out for Turkish with the results in the literature, it shows that a higher success rate was reached. Apart from Turkish, the method we recommended has also been tested for Swedish, Danish, Dutch and Portuguese languages. The success of studies in the literature has been exceeded to determine the kind of relationship. A poorer performance was exhibited to determine related word. This results from more variable of intra-sentence attachment structures of these languages other than Turkish.

A more dynamic structure should develop to enhance the performance of the method developed as future work in other languages.

Keywords: Sequence Labeling, Dependency Parsing, Condition Random Fields, Machine Learning, Natural Language Process

BÖLÜM 1

GİRİŞ

Sekans Etiketleme (Sequence Labeling - SL) , giriş sekansı I(i_1, i_2, .., i_n) şeklinde olan sekansa karşılık çıkış için O(o_1,o_2,...,o_n) sekansın üretilmesdir. O kümesine göre bir çok uygulama alanı bulunmaktadır. Örneğin, çıkış için üretilen O kümesi E(Eleman) {N(Noun), V(Verb) vb. } ise POS Etiketleme (Part of Speech Tagging- Pos Tagging), O kümesi E {kişi, organizasyon, yer vb.} ise Varlık İsmi Tanımlama (Name Entity Recognition - NER), O kümesi E{NP, VP vb.} ise Yüzeysel Parçalama(Shallow Parsing - SP), O kümesi E{ENG->TR, TR->ENG vb.} ise Otomatik Çeviri(Automatic Translation - AT), O kümesi E{Table, tr, td vb.} ise Tablo'dan bilgi çıkarımı(Table Extraction - TE), O kümesi E{Öznel-Nesnel, Pozitif-Negatif vb.} ise Fikir Madenciliği(Opining Mining - OM) olarak adlandırılır.

1.1. Literatür Özeti

Bu bölümde ana konumuz olan , SL ile ilgili yapılmış çalışmalarla ilgili bilgiler sunulacaktır.

Çalışma sonuçları için kullanılan metriklerimiz, Hassasiyet (Precision - P), Geri Çağırma (Recal - R) , F-Ölçütü (F-Measure - F), AS_U, AS_L ve DEP değerleri ile ifade edilmiştir.

P, erişilen ilgili sonuç sayısının erişilen toplam sonuç sayısına oranıdır.

R, erişilen ilgili sonuç sayısının derlemdeki toplam (hem erişilen hem erişilemeyen) ilgili sonuç sayısına oranıdır.

P ve R değerleri 0-1 arasında ifade edilen değerlerdir. P ve R ölçütleri tek başına anlamlı bir karşılaştırma sonucu çıkarmamıza yeterli değildir. Her iki ölçütü beraber değerlendirmek daha doğru sonuçlar verir. Bunun için F tanımlanmıştır.

F, P ve R değerlerinin harmonik ortalamasıdır. F, Eşitlik 1.1 'de görülmektedir.

$$F = (2 \times P \times R)/(P + R) \quad (1.1)$$

F metriği 0-1 aralığına ifade edilmektedir.

AS_U metriği sözcüklerin çekim eklerine ayrılarak ve bağlantı türünün olmadığı yani doğru ID'ye (doğru çekim eki grubuna) bağlanma oranıdır.

AS_L metriği ise sözcüklerin çekim eklerine ayrılarak ve bağlantı türünün sisteme verildiği yani doğru ID'ye doğru etiketle (bağlılık türü) bağlanma oranını gösteren metriktir.

DEP metriği doğru etiketi ile işaretlemeyi gösteren metriktir. AS_U , AS_L ve DEP metrikleri %'lik olarak ifade edilmektedir.

Sha ve Pereira (2003), SP için Eğitim seti CoNNL-2000 den elde edilen set kullanılmıştır. Test için Wall Street Journal(WSJ) 21. sayısından elde edilen set kullanılmıştır. Yapılan SP uygulamasında CRF ve MEMM'in performanslarını karşılaştırmışlardır. Yapılan çalışma sonucunda CRF F ölçütü cinsinden 0.943 iken ,MEMM 0.937 başarı göstermiştir [1].

Collins (1997), İngilizce için yüksek başarı gösteren ayrıştırıcısını tasarlarken ilk aşama olarak sadece sözcükler arası bağlılıklara dayanan bir ayrıştırıcı geliştirmiştir [2].Bu ayrıştırıcı, tümce içi sözcük dizilişleri çok serbest olmayan İngilizce için yüksek bir başarım sergilemese de, sözcükler arası bağlılıkların önemini vurgulaması ve ikili bağlılıkların olasılıklarına dayanan çok basit bir istatistiksel model olması nedeniyle bağlılık çözümlemesi konusundaki çalışmalara önemli bir kaynak oluşturmuştur [3].

Jurafsky ve Martin (2000), günümüzde sözcükler arası ikili bağlılıkların ayrıştırmasının başarısındaki etkisinin görülmesi ile birlikte bağlılık gramerleri kullanılmaya başlanmıştır. Bağlılık gramerlerinn en önemli üstünlüğünün, sözcük dizilişleri serbest dillerin ayrıştırmasındaki yetenekleri olduğudur [4].

Eisner (1996), İngilizce için geliştirdiği veri güdümlü ayrıştırıcısı üretimsel olasılık tabanlı modellerin bağlılık çözümlemesinde kullanılabilirliğini göstermesi açısından önemli bir çalışmadır. Eisner'in geliştirdiği, aşağıdan yukarıya ayrıştırma algoritması dinamik bir algoritmadır [5].

Haruno ve Ark. (1998), karar ağaçları kullanarak Japonca için bir ayrıştırıcı tasarlamışlardır. Japonca'da tümce çözümlemesi, "bunsetsu" adı verilen tümce parçacıkları arasındaki ilişkilerin tanımlanması ile gerçekleştirilmektedir. Bu dilde, parçacıklar tümce içerisinde serbest bir şekilde yer değiştirebildiklerinden ötürü, bağlılık gramerleri yaygın olarak kullanılmaktadır. Dilde ayrıştırmayı kolaylaştıran en önemli özellik, her parçacığın sadece sağ tarafında yer alan bir parçacığa bağlanabilmesidir. Böylece bağlılıklar her zaman için soldan sağa doğru oluşturulurlar. Bu durum arama uzayında daralmayı sağlar [6].

McDonald ve Ark., çalışmalarında DP'yi yönlü bir grafikte maksimum kapsayan ağacı bulma sorunu olarak işlemiş ve İngilizce ve Çekçe üzerinde denemiştir [7][8].

Oflazer (2003), Sonlu Durumlu Dönüştürücüler (SDD) kullanarak Türkçe için geliştirdiği ayrıştırıcısında kesişmeyen bağlılık varsayımını benimsemiştir. Gerekirci ve kural tabanlı olan bu ayrıştırıcıda SDD'ler giriş verisi üzerine belirli bir sonlanma kriterine ulaşılana dek birden çok kez uygulanmaktadır. Bu çalışmada en olası ayrıştırmayı bulabilmek için istatistiksel bir model kullanılmamış, bunun yerine toplam bağlılık uzunluklarına bakılarak olası ayrıştırmalar sıralanmıştır[9].

Nivre (2003), çalışmasında İsveççe için yığın yapılı bir bağlılık ayrıştırıcısı tasarlamıştır. Kural tabanlı olan bu ayrıştırıcıda insan tarafından oluşturulan gramer kuralları kullanılmıştır. Ayrıştırıcı, bu kuralları kullanarak karar verdiği öteleme ve çekme işlemleri sonucunda oluşan çözümlemelerin, kesişmeyen bağlılık varsayımına uygun olanlarını doğru olarak kabul etmektedir [10].

Nivre ve Nilsson (2003), yaptıkları çalışmada yine İsveççe için gerekirci bir ayrıştırıcı tasarlamışlardır. Bu çalışmada kesişmeyen bağlılık ve oluşan çıktının bir köklü ağaç olması sınırlandırması getirilmemiştir. Ayrıştırıcı girdi olarak sadece sözcük etiketlerini kullanmış ve en yakın bağlılıkları doğru olarak kabul etmiştir. Yapılan sınamalar

sonucunda en yakın bağlılık stratejisinin iyi sonuçlar verdiği ve eğer kesişmeme kısıtı getirilirse başarımda ve algoritmanın hızında artış gözlendiği belirlenmiştir [11].

Nivre ve Ark.(2004), yaptıkları çalışmada yine İsveççe için [10] nolu çalışmasına benzer yığın yapılı bir ayrıştırıcı geliştirmişlerdir. Bu çalışmadan farklı olarak, bu sefer bağlılık etiketleri de belirlenmeye çalışılmış ve insan tarafından yazılmış gramer kuralları yerine bellek tabanlı (BT) bir sınıflandırıcı kullanılmıştır. Sınıflandırmada kullanılan özelliklere farklı ağırlıklar verme, uzak ve yakın komşulara farklı ağırlıklar verme gibi yeteneklere sahiptir [12].

Sözcük etiketleri ile birlikte sözcüklerin görünüm bilgileri de ayrıştırıcıya girdi olarak verilmiş ve görünüm bilgisi eklemenin bağlılık çözümlemesinde önemli etkisi olduğu vurgulanmıştır. Bu modelde, BT sınıflandırıcı ayrıştırıcının öteleme ve çekme gibi hareketlerine karar vermede kullanılmıştır [12].

Bucholz ve Marsi (2006), CoNNL-2006 veriseti üzerinde gerçekleştirilen 13 farklı dil için gerçekleştirilen DP çalışmasının sonuçlarını yayınlamışlardır. 19 farklı ayrıştırma modeli içinden en başarılı 6 tanesi Çizelge 1.1' de gösterilmektedir. Sonuçlar AS_L metriği cinsinden gösterilmektedir [13].

Çizelge 1.1 Bucholz ve Marsi için sonuçlar [13]

Yöntem	Ar	Ch	Cz	Da	Du	Ge	Ja	Po	Sl	Sp	Sw	Tu	Bu	Ort.
McD.	**66.9**	**85.9**	**80.2**	**84.8**	**79.2**	**87.3**	**90.7**	**86.8**	**73.4**	**82.3**	**82.6**	**63.2**	**87.6**	**80.3**
Nivre	**66.7**	**86.9**	**78.4**	**84.8**	**78.6**	**85.8**	**91.7**	**87.6**	**70.3**	**81.3**	**84.6**	**65.7**	**87.4**	**80.2**
O'N	**66.7**	**86.7**	**76.6**	**82.8**	**77.5**	**85.4**	**90.6**	**84.7**	**71.1**	**79.8**	**81.8**	**57.5**	**85.2**	**78.4**
Rie	**66.7**	**90**	**67.4**	**83.6**	**78.6**	**86.2**	**90.5**	**84.4**	**71.2**	**77.4**	**80.7**	**58.6**	**0**	**77.9**
Sag	**62.7**	**84.7**	**75.2**	**81.6**	**76.6**	**84.9**	**90.4**	**86**	**69.1**	**77.7**	**82**	**63.2**	**0**	**77.8**
Che	**65.2**	**84.3**	**76.2**	**81.7**	**71.8**	**84.1**	**89.9**	**85.1**	**71.4**	**80.5**	**81.1**	**61.2**	**86.3**	**77.7**

Chen ve Ark. (2007), CoNNL-2007 veriseti üzerinde 10 farklı dil için DP çalışması gerçekleştirmişlerdir. Yapılan çalışmada DP programı olarak Malt Parser kullanarak Çizelge 1.2'de sonuçları elde etmişlerdir [14].

Çizelge 1.2 Chen ve ark. için sonuçlar [14]

Metrik-Dil	Arabic	Basque	Catalan	Chinese	Czech	English	Greek	Hungarian	İtalian	Turkish	Ortalama
AS_L	**74.65**	**72.39**	**86.66**	**81.24**	**73.69**	**83.81**	**74.42**	**75.34**	**82.04**	**76.31**	**78.06**
AS_U	**83.49**	**78.63**	**90.87**	**85.91**	**80.14**	**84.91**	**81.16**	**79.25**	**85.91**	**81.92**	**83.22**

Ambati ve Ark. (2010), Hindi Treebank üzerinden elde edilen veriler üzerinde DP çalışması gerçekleştirmişlerdir. Eğitim seti 1000 cümle ve Test seti 228 cümledir. DP programı olarak Malt Parser ve MST Parser kullanılmıştır. Çalışma sonuçları Çizelge 1.3' te görülmektedir [15].

Çizelge 1.3 Ambati ve Ark. sonuçlar [15]

Yöntem	Cross Validation(5 Fold)			Test		
	AS_U	**AS_L**	**DEP**	**AS_U**	**AS_L**	**DEP**
Malt Parser	**82.2**	**69.3**	**73.4**	**84.6**	**72.9**	**76.5**
MST Parser	**79.4**	**66.5**	**70.7**	**81.6**	**69.4**	**73.1**

Cer ve Ark. (2010), DP problemi için Penn Treebank üzerinden elde edilen verilerle çalışmışlardır. 4 farklı ayrıştırma modeline göre çalışma sonuçları Çizelge 1.4' te görülmektedir [16].

Çizelge 1.4 Cer ve Ark. için sonuçlar [16]

Yöntem	AS_U	AS_L
Covington	**80**	**76.6**
Nivre Eager Feature Interact	**84.8**	**81.1**
MST Parser (Eisner)	**82.6**	**78.8**
RelEx	**57.8**	**48.1**

Eryiğit ve Ark. (2011), Türkçe için bir DP çalışması gerçekleştirmiştir. Yapılan çalışmada METU-SABANCI Türkçe Treebank üzerinden elde edilen 5635 cümlelik bir set kullanılmıştır. Cross-Validation yöntemi kullanılarak set 10 parçaya bölünmüş ve her seferinde 1 parçası test olarak kullanılmıştır. Bu çalışmada veriseti V_0 olarak gösterilmiştir. Her çalışmada test seti mevcut set içerisinden 1 parça olarak seçilmekte ve sonuçların ortalamaları alınmaktadır. Yapılan çalışma sonunda AS_U için 76.1 ve AS_L için 67.4'lük doğruluk değerlerine ulaşmışlardır.

1.2. Tezin Amacı

SL uygulamalarında bir çok algoritma kullanılmaktadır. Markov tabanlı HMM,MEMM ve CRF bu algoritmalardan bir kaçıdır. Markov model yapısı ardışık olarak gelen düğümlerin olasılığının bulunmasında kullanılır. Markov model yapısında düğümler sadece gözlemlenen öğeleri ifade eder. Markov modelinde gözükmeyen bir islemin modelinden söz edilemez. Tüm Markov modeli gözlemlenen olayla bire bir örtüsür. Ancak HMM, gözlemleyemediğimiz ve varsaydığımız bağlantılar için saklı düğümler olusturarak bağlantıları bizim varsayımımıza göre yapar. Birleşik olasılıkları kullanarak çıkışları üretmeye çalışır. MEMM ise, durumlar arası geçişlerden ve çıkışların durumları oluşturma olasılıklarının birleşiminden bir koşullu olasılık hesaplayarak sorunu çözmeye çalışır. Hem HMM hemde MEMM normalizasyonu bulunduğu noktaya göre yaptığından yerel bir optimum değere ulaşılır. Özellikle düşük olasılıklı niteliklerin hesaba katılması Etiket Önyargısı (Label Bias) problemini ortaya çıkarır. CRF hem bu etiket önyargısı problemini çözer hemde HMM'den farklı olarak koşullu olasılıkları hesaba katarak işlemleri gerçekleştirir. Bu artılarından ötürü CRF diğer iki yönteme göre daha yüksek başarı oranına sahiptir.

Bağlılık ayrıştırması (Dependency Parsing-DP), bir cümle içerisindeki, sözcükler arasındaki ilişkileri ve ilişki türlerini belirleyerek ilgili cümlenin çözümlemesini sağlayan yöntemdir. DP problemi de bir çeşit SL uygulamasıdır.

SL problemlerinde giriş ve çıkışlar n sayıda olabilir. Klasik anlamdaki NER, POS Etiketleme vb. uygulamalarda tek giriş ve tek çıkış bulunmaktadır. Cümleyi Öğelerine

Ayırma (Role Labeling-RL) probleminde iki giriş ve bir çıkış bulunmaktadır. DP probleminde ise n giriş ve 2 çıkış olabilmektedir.

Karmaşık bir problemi tamamen ele alıp çözmektense parçalara ayırarak çözmek daha kolaydır. Böylelikle problem daha basit alt parçalara bölünmüş ve çözüm daha sade hale getirilmiş olmaktadır.

Karmaşık problemin daha basit alt parçalara bölünmesi ile sistemin başarısının artacağı düşüncesi ile bileşik cümleler yan cümleciklerine bölünecek ve öğelerine ayırma işlemi bu alt parçalar üzerinde gerçekleştirilecektir. Bu çalışmanın sonuçları bize asıl hedefimiz olan DP problemlerinin çözümünde basit alt parçalara bölmenin ve CRF'nin kullanılabilirliği test etmek için bir ön çalışma niteliğindedir.

Amacımız, SL işleminde sıklıkla kullanılan ve Markov tabanlı modeller içinde en başarılısı olan CRF algoritmasını DP problemlerinin çözümünde kullanabilmektedir. Böylelikle bir çıkış üreten CRF sisteminin DP gibi 2 çıkışın bulunması istenen problemlere uygulanabilmesi için bir yöntem önerilmiş olacaktır.

1.3. Hipotez

Birinci hipotezimiz , karmaşık bir problemi daha basit alt parçalara bölmenin problemin çözümünü kolaylaştırdığı ve bu durumun başarı üzerinde olumlu bir etki yapacağı yönündedir. Bu nedenle gerçekleştirilmesi düşünülen içerisinde birden fazla yargı bildiren bir bileşik cümleyi öğelerine ayırmak için sisteme bir bütün halinde vermektense , cümleyi temel ve yan cümleciklerine bölerek vermenin cümleyi öğelerine ayırma işleminde başarıyı artıracağı iddia edilmektedir.

İkinci hipotezimizin temelinde karmaşık bir problemi daha basit alt parçalara bölme fikri yatmaktadır. Bu hipotezimizde ayrıca DP gibi iki çıkışlı problemlerinin çözümünde problemi ardışık olarak iki aşamada çözmenin sistemin başarısını artıracağı iddia edilmektedir.

Gerçekleştireceğimiz uygulamalar bu iddialarımızı kanıtlamaya yönelik olacaktır.

BÖLÜM 2

DOĞAL DİL İŞLEME

2.1. Giriş

Matematik alanındaki gelişmeler bilgisayar dünyasını da derinden etkilemiştir. Yüzyıllardır üzerinde çalışılan felsefe ve matematiğin mantık bilimi disiplini ile birlikteliği, günümüzde Yapay Zeka (Artifical Intelligent - AI) çalışmaları ile daha da anlam kazanmaktadır [18]. Metin Madenciliğinde veriden bilgi çıkarma yöntemlerinden biri olan doğal dil işleme disiplini ile bilgi çıkarımında daha anlamlı sonuçlar elde edilmeye başlanmıştır. Doğal dil işleme ana işlevi bir doğal dili çözümleme, anlama, yorumlama ve üretme olan bilgisayar sistemlerinin tasarımını konu alan bir mühendislik alanıdır. Doğal dil işleme çalışmaları sayesinde insan-bilgisayar etkileşiminin arttırılması başarılmıştır. NLP bilgi şifreleme, konuşma tanıma, optik karakter belirleme, yazı doğrulama gibi pek çok alanda kullanılır.

Doğal dil işleme çalışmaları kapsamında yürütülen girişimler beş ana grup altında toplanabilir [19]. Ses Bilimi- Morfolojik Analiz (Biçimbirim) - Sözdizimsel Analiz (Sentaktik) - Anlamsal Analiz (Semantik) - Söylem.

- **Ses Bilim:** Sesbilim bir dilde seslerin nasıl kullanıldığıyla ilgili çalışmaları kapsamaktadır. Her dilde, onu diğer dillerden ayıran bir ses abecesi vardır. Bunların herbirine sesbirim adı verilir. Türkçe abecesi 29 harften oluşur ve bilinen diller arasında, bir harfe karşı bir ses düştüğü varsayılır. Ancak, bu alanda çalışanların ortaya koyduğuna göre, Türkçenin sesbirim sayısı 45 tir. Bunlardan 15 tanesi sesli, 27 tanesi ise sessizdir [20].

- **Morfolojik Analiz (Biçimbirim):** Biçimbirim sözcüklerin yapısıyla ile ilgilenir. Türkçe için sözcüklerin türetilmesi ve ekler çok önem taşır. Her dilde iki farklı şekilde sözcük oluşturulabilir. Bunlardan biri çekim, 11 diğeri ise türetme yöntemidir. Çekim yoluyla sözcük oluşturulurken bir sözcüğün farklı şekilleri kullanılır. çiçek sözcüksine çekim eki eklenmesiyle çiçekler sözcüğünün ortaya çıkması bu yolla oluşturulan sözcüklere örnek olarak verilebilir. Türetme ise var olan eski sözcüklere yapım ekleri eklenmesi yoluyla yeni sözcük oluşturma yöntemidir [21].
- **Sözdizimsel Analiz (Sentaktik):** Sözdizim bir tümcenin içindeki sözcüklerin dizilişi ile ilgilenen düzeydir. İnsanlar konuşurken ya da yazarken var olan sözcükleri belli bir sırada yerleştirerek yeni tümceler oluştururlar. Bu nedenle sözdizimsel düzey, sesbilim ve biçimbirim düzeylerinin üstünde yer alır [21].
- **Anlamsal Analiz (Semantik):** Anlambilim düzeyi dilin anlamıyla ilgilenir, böylelikle gerçek dünyayla dil arasında bağlantı kurar. Bu düzeyde yapılan çalışmalar daha önce anlatılan düzeylere göre çok azdır [21].
- **Söylem:** Söylem düzeyi dilin bağlamda kullanımıdır. Anlambilim ve söylem düzeyleri arasındaki fark çok küçüktür. Bazı araştırmacılar bu iki düzeyi birlikte ele alır [21].

2.2. Doğal Dil İşlemenin Amaçları

Doğal dil; Türkçe, İngilizce, Almanca, gibi insanların iletişim için kullandığı herhangi bir dil olup, yapay olarak insanlar tarafından geliştirilen programlama dillerinden farklıdır. NLP ise doğal dili bilgisayarlarla işlemek için yapılan çalışmaların bütünü olarak tanımlanabilir.

Bilgisayarlar ve insanlar kendilerine verilen bir metni veya konuşmayı veri olarak aynı şekilde almalarına rağmen; bunlardan anladıkları birbirinden çok farklı olmaktadır. Bunun en önemli nedeni insanların yaşamları boyunca kazandıkları tecrübelerle iletişim için geliştirmiş oldukları ortak bilgi birikimi, çıkarım yapabilme yeteneği ve tecrübelerini kullanabilme yetisidir. Oysa bilgisayarlar sadece belirli donanım ve yazılım birimleri içerir; bunun dışında insanlarda bulunan potansiyel yetenek ve yetilere sahip değildir.

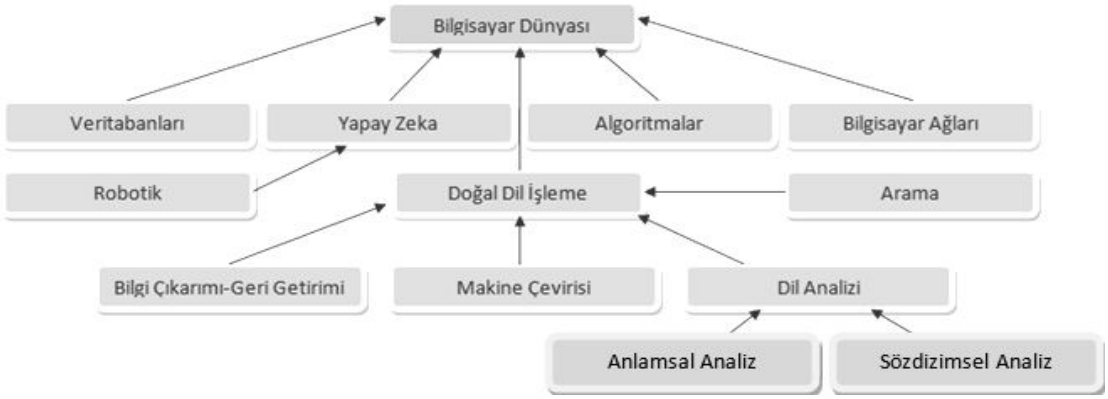
Belirli işlemleri yerine getirebilmeleri ancak onlara bu işlemleri nasıl yapabileceklerinin öğretilmesi ile sınırlı olarak mümkündür; çünkü buradaki öğrenme kavramı insanlardaki öğrenmeden farklıdır ve daha çok bilgisayar dilinde verilen komutların belli bir sıra dahilinde işlenmesini sağlayarak sonuçlara ulaşmak anlamındadır [18].

İngilizce veya başka herhangi bir dil için tanımlanan kurallar ve algoritmaların aynen Türkçe'ye veya başka bir dile uyarlanması dillerin yapısının farklılığından dolayı mümkün değildir. Mevcut sistemlerin kullanılacak dile uyarlanması mutlaka uzun ve yorucu çalışmalar sonunda gerçekleşmektedir. Hatta çoğu zaman bunların uyarlaması bile mümkün olmamakta ve dile özgü pek çok sistemin baştan oluşturulması zorunluluğu ortaya çıkmaktadır. Ayrıca; bu alandaki çalışmalar pek çok konuda özellikle bilgisayar bilimi ve dilbilimi konusunda uzmanlık gerektirmektedir. Bu nedenle herhangi bir dil ile ilgili olarak yapılacak bilişimsel çalışmalar, hem o dilin dilbilimcileri ve bilgisayar bilimleri uzmanları, hem de incelenen dili çok iyi bilen diğer bilim insanları tarafından gerçekleştirilebilir [18].

NLP çalışmalarının amaçları aşağıdaki gibi özetlenebilir:

- Dilin doğasını incelemek**:** Dilbilim.
- Bilişsel yetileri anlamaya yönelik bir kapı aralamak **:** Psikoloji
- Kullanıcı arayüzü teknolojisi olarak faydalanmak**:** İnsan-Bilgisayar Etkileşimi.
- Metin veya konuşmaların çevirisini yapmak**:** Bilgisayarlı Çeviri.
- Bilgi yönetimi teknolojisi olarak kullanmak**:** Bilgi Geri Getirimi/Çıkarımı.

Bu amaçlar da göz önünde bulundurularak yapılacak bir değerlendirme ile, NLP'nin disiplinlerarası bir çalışma alanı olarak pek çok farklı bilim dalıyla birlikte, bilgisayar biliminin alt dallarıyla da yakın ilişkisi bulunduğu sonucuna ulaşılır. Bu ilişkiler Şekil 2.1'de gösterilmiştir.



Şekil 2.1 NLP'nin farklı bilim dallarıyla ilişkisi [18]

2.3. Doğal Dil İşlemenin Gelişimi

Alan Turing'in II. Dünya Savaşı sırasında Almanların şifreli haberleşmesini çözmek için yaptığı çalışmalar sonlu durum makineleri ve Turing makinelerinin temelini oluşturmuş, ortaya atılan programlanabilir makine fikri ile çağımızın en önemli gelişmesi olan bilgisayarlar ortaya çıkmıştır. Turing'in bu çalışmalarının AI ve NLP konularının da temelini oluşturduğu kabul edilir. Aslında, AI kavramı ile ulaşılmak istenen hedef, insanda var olan insan yetisinin özgününden ayırt edilemeyecek bir şekilde benzetiminin gerçekleştirilmesidir.

Yine Turing tarafından gerçekleştirilen Turing testi, bu hedefin ilk defa ortaya konulduğu uygulama olarak kabul edilmiştir. Turing testine göre; "eğer görmeden konuşulan bir makine ile insan birbirinden ayırt edilemiyorsa, makine insanı mükemmel şekilde taklit etmektedir" denilebilir.

Turing testinden sonra bunu izleyen benzeri pek çok çalısma yapılmıştır. Örneğin, 1966 yılında Joseph Weizenbaum‘un tasarladığı ELIZA adlı program çok basit kalıpları tanıyarak kullanıcısına psikolojik terapi uygulayabiliyordu [22]. Bu program en çok tanınan ve en eski olarak kabul edilen AI programıdır. Programın amacı, bir psikanalist ile hastası arasındaki konuşmaları makineyle basit bazı analizler yaparak sağlamaktır.

Başlangıçta AI'nın küçük bir kolu olarak kabul edilen NLP, kısa sürede uygulama alanlarını genişleterek tek başına bir disiplin olmuştur. Son yıllarda bilişim alanında gerçekleştirilen önemli gelişmeler, bilgiyi ön plana çıkararak bilgi toplumu kavramını

doğurmuştur. Günümüzde özellikle İnternet üzerinden istenilen bilgiye ulaşma ve iletişim çok kolaylaşmıştır. Buna paralel olarak da bilgi ve teknolojinin farklı birey ve toplumlar arasında paylaşımını kolaylaştırma, diller arası çeviri veya belge indeksleme/bulma gibi geniş kapsamlı öte veri (meta data) içeren araştırmalar için daha gelişmiş dil işleme yazılımları gereksinimi ortaya çıkmıştır [18].

Bilgi Çıkarımı (Information Extraction-IE), NLP çalışmalarının ilklerindendir. İlk IE çalışmalarında basit bir yaklaşımla sözcükler çevirisi ve sıralaması üzerinde durulmuştur. Bu anlayışa dayanan sistemler; çeviri için sözcüklerin diğer dillerdeki karşılığını iki dilli bir sözlükten bulmuş ve çeviri yapılan dilin sözcük sıralama kurallarına göre bulunan sonuçları sıralamıştır. Doğal olarak bu yaklaşımdan elde edilen sonuçlar çok başarılı olamamıştır. Çeviri işlemini otomatik olarak yapma çalışmaları çok eskilere dayansa da gerçek anlamda uygulanabilirliği yirminci yüzyılda mümkün olmuştur.

II. Dünya Savaşı sırasında şifre çözme konusunda çalışan Weaver'a göre bütün uluslar, aslında farklı diller de konuşsalar birbirine benzemektedir [23]. Bu nedenle de bir dilde yazılmış bir metin, şifrelenmiş bir metin olarak düşünülebilir. Eğer bu şifre çözülebilirse bu metnin başka bir dilde ifade edilmiş şekli elde edilebilir. Bu düşünceye göre Türkçe yazılmış bir metin, aslında İngilizce bir metnin şifrelenmiş halidir.

Bu ilginç önerilerin ardından Amerika, Rusya, Fransa ve İngiltere'de pek çok üniversitede IE çalışmaları başlatılmıştır. İlk çalışmalarda daha çok Almanca ve İngilizce çeviriler ele alınmıştır. Bu dillerin seçilmesindeki en önemli etken, savaştan kalan teknik belgelerin çevirilmesi gereksinimidir. Zamanla bu belgelere gereksinim kalmadığı için ve soğuk savaş nedeniyle çeviriler İngilizce, Rusça ve Fransızca dillerine kaymıştır. Her ne kadar iki dili, anadili gibi konuşan kişiler IE çalışmalarında yer alsa da, onların bilgi aktarımının bilgisayar programına dönüşmesi hiç de kolay olmamıştır.

Bu nedenle 1950'li yıllarda dilbilimciler de IE çalışmalarında yer almaya başladılar. Ancak dilbilim alanında da henüz güçlü teoriler bulunmuyordu. 1957 yılında Noam Chomsky, yazdığı *Syntactic Structures* isimli eseri ile teorik alanda günümüze kadar etkisini pek çok alanda gösterecek bir çalışma ortaya koydu [24]. Biçimsel yapıların tanımlanması için ortaya koyduğu kurallar NLP çalışmalarını derinden etkiledi.

NLP konusunda yapılan çalışmalar bu dönemde konuşma işlemeyi de içine alacak şekilde çeşitli alanlara da yayıldı. Ancak, araştırmacılar arasındaki farklı görüşler bu alandaki çalışmaları ikiye böldü. Bir grup, tamamen dilbilimi üzerine yoğunlaşarak Chomsky'nin yaklaşımını benimseyip istatistiksel yöntemleri kullanmamayı tercih ederken, diğer bir grup da tamamen istatistiksel yöntemlere yöneldi.

1980'li yıllarda bilgisayarlar ve elektronik kaynaklardaki gelişmelerle NLP çalışmalarında da önemli ilerlemeler olmuş, bu dönemde istatistiksel yöntemler de sembolik yöntemleri tamamlayıcı bir faktör olarak tekrar ele alınmıştır.

1990'lı yıllar ise elektronik metinlerin oldukça artması, bilgisayar kapasite ve hızının çok iyileşmesi ve İnternet sayesinde, NLP'de çok gelişmiş ve istatistiksel yöntemler farklı alanlarda kullanılır olmuştur. Yapılan çalışmalarda artık genel metinler incelenmekte ve çok farklı uygulamalar geliştirilmeye çalışılmaktadır. Bu dönemde IBM (International Business Machines) tamamen istatistksel yöntemler kullanan CANDIDE sisteminin sonuçlarını açıklamıştır [25]. Japonya'daki çalışmalarda örnek tabanlı yaklaşım ortaya atılıp kullanılmaya başlanmıştır. İstatistiksel ve örnek tabanlı bu yaklaşımların ortak noktası biçimsel ve anlamsal kurallar yazma gereği bulunmayışı ve gerekli bilginin büyük derleme metinlerden çıkarılmasıdır.

Günümüzde yaygın olarak kullanılmakta olan bilgisayarlarda ses kontrolü ön plana çıkmaktadır. Burada ulaşılmak istenen hedef, insan-makine iletişimini daha özgür ve rahat bir konuma getirmektir. Çalışmalar esas olarak; makineye bir problemin iletilmesini ve çözümünün oluşturulup uygun bir biçimde kullanıcıya aktarılmasını gerçekleştirmeyi hedeflemektedir.

Anlamsal analiz ise bu tür çalısmaların kalbidir ve bu konudaki incelemeler yoğun olarak devam etmektedir. Burada asıl amaç, mevcut bir bilgi tabanının bilgisayar tarafından yorumlanabilecek şekilde bir gösteriminin elde edilmesidir. Yapılan çalışmalarla sürekli olarak farklı alanlarda yeni kullanım olanakları ortaya çıkmaktadır.

2.4. Doğal Dil İşleme Kapsamında İncelenen Konular

NLP bağlamındaki ana konular aşağıdaki ana başlıklarla özetlenebilir [18]:

• **Konuşma sentezi:** İlk bakışta çok kolay gibi görünen bir işlem olmakla birlikte, doğal konuşma şekline yakın sentezleme işlemi teknik olarak oldukça karmaşıktır ve birleştirilecek konuşmanın doğru yapılabilmesi (örneğin, doğru vurgulama elde edilmesi) için bazı ayrıntıların belirginleştirilmesi gerekir.

• **Konuşma tanıma:** Temel olarak; sürekli ses dalgalarının sözcüklere dönüştürülmesidir.

• **Doğal Dil Anlama (DDA):** Konuşma tanıma veya metinlerden elde edilen sözcüklerden, anlama dönüşümü gerçekleştiren basamaktır.

• **Doğal Dil Üretme (DDÜ):** Verilen girdi bilgilerine karşılık uygun doğal dil cevabının oluşturulması işlemidir.

• **IE:** Verilen metnin bir doğal dilden diğerine bilgisayarlar yardımı ile çevrilmesidir.

Konuşma tanıma teknolojisinin kullanımı ile yapılabilecekler öncelikle bilim kurgu filmlerinde insanlığın ulaşmak istediği hedefler olarak ortaya konmaktadır.

Filmlerdekiler kadar kapsamlı olmasa da, artık günlük hayatta sıkça kullanılan bazı konuşma tanıma sistemleri modern toplumlarda yerini birer birer yerini almaktadır. Bunlara örnek olarak, operatör servislerindeki faturalama işlemlerinin otomatikleştirilmesi, aramaların sesle yönlendirilmesi, bazı standart formlar veya benzeri dökümanların otomatik olarak doldurulması verilebilir. Konuşma tanıma sistemleri daha çok konuşmacıyı doğrulama esasına göre çalışır ve insanlardan daha üstün bir başarı sağlar. Suçlu belirleme için hukuk alanında, güvenliği sağlamak için bankacılık sistemlerinde, özel güvenlik bölgelerine girişlerde, sese dayalı şifreleme sistemlerinde konuşma tanıma uygulamalarından sıkça yararlanılmaktadır.

DDA, pek çok NLP uygulaması için önemlidir. Ancak herhangi bir yazı veya konuşmayı tam olarak anlama, her zaman gerekli olmasa da, kısmen anlama tam olarak anlama işlemine gerek olup olmadığı konusunda bilgi vermesi bakımından çoğu zaman faydalı bir ilk basamak oluşturur.

Yüzeysel veya kısmen yapılan metin analizleri, sınırlandırılmamış metinlerin etkin bir şekilde sınıflandırılmasında kullanılabilir. Bu ön işlemden sonra elde edilen bilgilerle

anlamsal olarak metnin daha derinlemesine incelenmesi sadece gerekli bölümlerde yoğunlaşabilmektedir. Ayrıca, istatistiksel ve dilbilimsel bilgilerle birlikte kullanıldığında sistemlerin bilgi tabanlarına eklenebilecek bilinmeyen öğelerle ilgili dilbilimsel özelliklerin çıkarılmasında da faydalı olmaktadır.

Anlam, kavramlar ve bu kavramlar arasındaki ilişkilerden oluşan anlamsal modellerle gösterilmektedir. Bu anlamsal model kullanılarak herhangi bir şekilde istenen bir bilginin veya verilen bir sorgu ifadesinin bu ifade veya sorgunun dilinden veya anlatımından bağımsız olarak bir anlama eşlemesi yapılabilmektedir. Bu şekilde bilgiye farklı dillerle erişim olanağı dilden veya yapıdan bağımsız olarak sağlanmaktadır.

Analiz ve üretimle birlikte kullanılan anlamsal bir modelle IE veya başka bir NLP uygulaması gerçekleştirilebilir. Ancak günümüzdeki uygulamalarda bunun başarılabilmesi için genel olarak sınırlı sözcük ve kavram hazinesi kullanılmaktadır. Belge yapıları için çıkarılan bazı kalıplar, değişken parçaları olan sık kullanılan ifadeler yardımıyla yüksek kalitede metin üretimi gerçekleştirilmektedir.

Çeviri yapan sistemler daha karmaşık ve üst sistemlerdir ve diğer alanlarda yapılan ilerlemelere bağlı olarak gelişimlerini sürdürmektedirler. Bu konuda çalışma yapan labaratuvarlar ve sistemler dünyanın dört bir tarafında dev şirketler tarafından oldukça yüksek bütçelerle kurulmaktadır.

İnternet ve mobil teknolojinin dünyada hızla yaygınlaşmasıyla çeviri sistemlerine gereksinim de artmıştır. Bu nedenle Google gibi arama motorları da çeviri alanında önemli yazılımlar geliştirmiştir. Genellikle bu işlemleri gerçekleştirebilmek için, sözcükten tümceye ve oradan da anlam çıkarma işlemleri için çeşitli analiz yöntemlerine gereksinim vardır [18].

BÖLÜM 3

DERLEM

3.1. Giriş

Türkçe'de derlem (corpus) sözcüğü Güncel Türkçe Sözlük'te iki anlamlı olarak verilmiş, birinci anlamında "koleksiyon" sözcüğüne karşılık olarak, ikinci anlamında da terim anlamı olarak "Bir incelemede kullanılmak üzere bir araya getirilmiş metinlerin bütünü." biçiminde tanımlanmıştır [26].

3.2. Derlem Kavramı

Derlem, terim olarak özellikle dil araştırmalarına temel olmak üzere belirli ilkeler, kurallar çerçevesinde bir araya getirilmiş, özel ya da genel çeşitli türlerde hazırlanmış, elektronik ortamda tutulan ve bilgisayarca okunur biçime getirilmiş sözlü ya da yazılı bütündür [27].

Derlem araştırmalarının deyiş biliminden (stilistik), bütün dil bilgisi araştırmalarına, NLP çalışmalarındaki dil modellemesine kadar geniş bir kullanım alanı vardır. Son 20 yılda gelişen bilgi teknolojilerine bağlı olarak derlem oluşturma ve sorgulama biçim ve hızında büyük ilerlemeler yaşanmıştır [28].

İngilizce için derlem oluşturma çalışmalarıyla tanınan John Sinclair, derlemin temsil gücünün önemine değinerek, derlemin rasgele bir araya getirilmiş metinler yığını olmadığını özellikle vurgulamıştır. Belirli dilsel ölçütlere göre bir araya getirilen metinlerin derlem oluşturabileceğini ve derlemi geliştirme amacının birincil ölçütlerden olduğunu belirtmiştir. Böylelikle derlem kavramı makinece okunur, örneklemesi belirli ölçütler çerçevesinde konuşma dilinden ya da yazılı metinlerden yapılmış ve dil bilimsel

işaretlemesi hazırlanmış bir koleksiyon olduğu modern dil bilimde kabul edilen tanımdır [29].

Derlem tabanlı çalışmaların ya da derlem dil biliminin bir yöntem mi olduğu yoksa söz dizimi, biçim bilimi, ses bilimi gibi dil biliminin bir alt alanı mı olduğu sorusu tartışmalı konular arasındadır. Bununla birlikte, Tognini ve Bonelli'nin 2001 yılındaki çalışmalarında derlem dil bilimini ayrı bir dil bilimi disiplini olarak kabul ettiklerini belirtmektedirler [29].

Derlem dil biliminin önemli araştırmacılarından Geoffrey Leech bir derlemin özelliğini şu biçimde dile getirmiştir: "Derlemin (bilgisayar destekli derlem) nadiren gelişigüzel toplanan bir bütün olduğu düşünülmelidir. Derlemler genellikle belirli amaçlar için dilin bir bölümünü ya da kimi metin türlerini *temsil etmek* için oluşturulurlar" [29].

Derlem oluşturma yöntemlerinin kuramsal çerçevesi ve uygulamalarına yönelik araştırmalar günümüzde hız kazanmış, İngilizce başta olmak üzere diğer dillere yönelik derleme dayalı yöntemler dil bilgisel modellerin oluşturulması ya da desteklenmesi amacıyla daha çok kullanılmaya başlamıştır. İşin kuramsal yönünde doğrudan derlemden çıkan sonuçlar üzerinden dil bilgisel modellemelerle, var olan kuramsal dil bilgisi çerçevesinin derlemle desteklenmesi konusunda da bir ayrıma gidildiği görülmektedir [28]. Bu konuda Ooi, derleme dayalı bu iki bakış açısı arasındaki farkları şöyle belirlemiştir [30]:

Derlem Tabanlı Yöntem

- Dil bilimci var olan kuramı kanıtlama için derlemi kullanır.

- Dil bilimci dayandığı kuramın sağlamlığından kuşku duymaz dolayısıyla derlemden çıkan sonucu tehdit olarak görmez.

Derlem Yönelimli Yöntem

- Gözlemlenen veriden yeni kuramsal çıkarımlar yapılabilir.

- Dil bilimci derlem verisinin kuramda değişiklikler yapmasını bekleyebilir.

3.3. Derlem Araştırmalarının Tarihçesi

Derlem terimi, 1980'lerin başlarında ortaya çıksa da, derlem dil bilimi yöntemini Chomsky öncesi dil bilimcilerinden Boast ve yapısalcı dil bilimi akımının Amerika'daki temsilcilerinde Sapir, Mewman, Bloomfield ve Pike kullanmışlardır [28].

1950'li yılların sonlarında derlem çalışmalarına, araştırmacıların bireysel çabalarının yetersiz kalmasının ve teknolojinin ilk derlem çalışmaları denebilecek çalışmalar için bulunmamasının yol açtığı kimi olumsuzluklar özellikle Chomsky'nin sert eleştirilerine hedef olmuştur. İlk derlem tabanlı çalışmaların ses bilimsel ayırt edici özellikler üzerine kurulan küçük boyutlu derlemler olduğu görülür. Jesperson ve Fries ses bilimi dışında genel dil bilgisine yönelik derlem tabanlı çalışmalar yapmışlardır.

Teknolojinin gelişimiyle küçük, elle toplanan verilerden bilgisayar verisi olarak derlenen ve depolanan derlemler oluşturulmaya başlanmıştır. 1960'lı yıllarda Brown Üniversitesi'nde hazırlanan Brown Derlemi, bilgisayar destekli olarak hazırlanan ilk derlem olarak derlem teknolojisinin başlamasını sağlamıştır. 1980'li yıllardan günümüze devam eden teknolojik gelişmelere bağlı olarak bugün birçok derlem çeşitli amaçlar için hazırlanmaktadır [29].

3.4. Derlem Türleri

Derlemler genel ve özel derlemler olarak iki gruba ayrılmaktadır. Genel derlemler bir dildeki bütün alt çeşitliliği barındıran derlemlerdir. British National Corpus (BNC), 100 milyon işaretlenmiş sözcükten oluşan ve derlem dil bilimi kaynaklarında sözü edilen ilk büyük derlemdir.

Özel derlemler alan ve türe özgü derlemlerdir. Hukuk, tıp, mühendislik gibi bilim alanlarına göre ayrılabildikleri gibi, yine kendi içinde gazete, roman, dergi, akademik yazılar gibi türlerden de oluşabilmektedir. Genel ve özel derlemler temsil güçlerine göre farklılıklar gösterirler. Genel derlemlerin içlerinde barındırdıkları türe bağlı olarak temsil güçleri artar ya da eksilir. Özel derlemler de barındırdıkları sözcük düzeyi kadar temsil gücüne sahiptirler [29].

Bugün derlem dil bilimi çalışmalarının en çok gerçekleştirildiği ve buna bağlı uygulama yazılımlarının en çok yazıldığı dil İngilizcedir. İngilizce için geliştirilen genel ve özel

amaçlı derlemlerin yanında özellikle bir derlem türü öne çıkmakta, bu derlem türünün temsil niteliği ve kullanım alanıyla ilgili tartışmalar sürmektedir.

Monitör derlem (monitor corpus) adı verilen bu derlem türünde sürekli gelişen ve genişleyen bir yapı söz konusudur. Malzemesi örneklemeyle elde edilen durağan (statik) derlemler yerine bu türde, belirlenen amaç için çeşitli konularda genişleyen boyutlarda metin yığınları güncel olarak bir araya getirilmektedir. Aylık, yıllık ve hatta günlük olarak konuyla ilgili metinler değerlendirilmek, ön işlemlerden geçmek üzere ham olarak toplanır. Bank of English 524 milyon sözcüklük derlem büyüklüğüyle bugün için monitör derlemin en önemli temsilcisi olarak öne çıkmakta, 1980'den günümüze sürekli geliştirilmektedir. Global English Monitor Corpus adlı derlem de 2001 yılından günümüze var olan metinlerden oluşturulmakta, 1 milyar sözcüklük bir büyüklüğe ulaşacağı öngörülmektedir. Bu derlemin oluşturulma amacı; İngiltere, Amerika Birleşik Devletleri (ABD), Avustralya, Pakistan ve Güney Afrika'da kullanılan İngilizcenin farklılıklarını ve değişimini izlemektir. Bir başka monitör derlem örneği Birmigham Üniversitesinde geliştirilen AVIATOR (Analysis of Verbal Interaction and Automated Text Retrieval) adlı derlem yapılı sistemdir. Sistemin kurulma amacı, İngilizce metinleri sürekli ve güncel olarak izleyerek dilsel değişimleri ve özellikle yeni ortaya çıkan ögeleri belirlemektir. Bu tür derlemde bir sınırlama getirilmediğinden artan ya da büyüyen bir arşiv olarak da değerlendirilmektedir [29].

3.5. Başlıca Derlemler

Derlem araştırmalarıyla ilgili yayınlarda, tarihsel olarak ilk makinece okunur biçimli derlemin Brown derlemi olduğu belirtilmektedir. 1964'te bir el kitapçığıyla birlikte tamamlanıp sunulan Brown derlemini Nelson Francis ve Henry Kučera hazırlamıştır. Derlem Brown Corpus olarak kısaltılarak kullanılmakla birlikte orijinal adı Brown University Standard Corpus of Present-Day American English'tir [28].

Chomsky'nin 1957'de yayımladığı Sözdizimsel Yapılar (Syntactic Structures) adlı yapıtının dil biliminde üretici-dönüşümsel gramere dayalı bakış açısının ortaya çıkmasını sağladığı ve bu kurama dayalı çalışmaların arttığı bu yıllarda Brown derleminin ortaya konması, dil biliminde ürün olarak ortaya çıkmış dilsel ifadelerin değil de ideal konuşucu ve dinleyicinin üretebileceği zihinsel yapılara dayalı bir gramer

anlayışı karşısında oldukça anlamlıdır. Chomsky'nin derlem çalışmalarının hiçbir zaman yeterince kapsayıcı olmayacağı düşüncesinde olduğu bilinmektedir [28].

Brown derlemi 1961 yılına dayanan Amerikan İngilizcesinin eşzamanlı bir derlemidir. Her biri yaklaşık 2000 sözcük içeren 500 örnekten toplam 1.014.300 sözcük o zamanın bilgisayar teknolojisine göre öncelikle delikli kartlara sonra da manyetik teyplere kaydedilmiştir. Çalışmanın amacı varsayılan bir biçemsel kalitenin temsili amacıyla değil de o güne ait İngilizcenin karşılaştırmalı çalışmalar ve çözümlemeler için ölçünleştirilmiş bir görünümünü sunmaktır. Derlemin malzemesi yerel kütüphanelerden elde edilen metinlerden oluşmaktadır. Sözcüklerin türlerini belirlemek için 80'in üzerinde etiket belirlenmiştir [31].

1970-1978 yılları arasında Brown derlemine koşut olarak Oslo ve Lancaster Üniversitelerinin ortak olarak hazırladığı Lancaster-Oslo/Bergen derlemi (LOB), Brown derleminin oluşturulma teknikleri örnek alınarak oluşturulmuştur. Bu derlemle birlikte derlem dil biliminin hem kuramsal hem de yapısal birtakım sorunları gün ışığına çıkmaya başlamıştır. Özellikle bir derlemin temsil gücünün metinlere oranlanması konuları tartışılmaya başlamıştır. Bu bağlamda derlem ekibinin başkanı Johansson, LOB'un metin seçiminin rastlantısallıktan uzak olarak amaçlı bir sınıflamaya dayandığını belirtmiştir. Brown derlemine göre LOB'un metin türlerinin sınıflandırılması daha tutarlı görünmektedir. 1970'li yıllardaki bilgisayar teknolojisinde kaydedilen ilerlemelere paralel olarak derlem, Macintosh, MS-DOS ve Unix platformlarına taşınmak üzere kaydedilmiştir. Ayrıca bu derlemle birlikte cümle sınırlarının ve kısaltmaların işaretlenmesi için de ayrıca bir kodlama sistemi geliştirilmiştir. Bağlamlı dizin oluşturucu bir sistemde sorgulanan sözcüğün ortada gösterimine dayanan bağlamlı satırların ortaya konulması yöntemi de bu derlemde uygulanmaya başlamıştır. İngiliz İngilizcesini temel alan bu derlemle Amerikan İngilizcesini yansıtan Brown derlemindeki sözcük sıklıkları bilgisi bu iki İngilizcenin karşılaştırılmalı çalışmalarına da katkı sağlamıştır [31].

Konuşma diline yönelik olarak hazırlanan ilk derlem London-Lund derlemidir (LLC). Bu derlem Survey of English Usage (SEU) olarak bilinen derlemin çeşitli ses bilimsel ölçütler içerinde işaretlenmesini içermektedir. 87 adet metin ve 435.000 sözcüğün

işaretlendiği derlem 1990'lı yıllara kadar kullanılan tek sözlü dile dayanan derlemdir. Bu derlem sözlü dilde kullanılan sözcükleri ve cümle yapılarını prozodik açıdan betimlediğinden özellikle söylem çözümlemesi çalışmalarında veri sağlayıcı olarak kullanılmıştır [28].

1984-1987 yılları arasında 11 kategoride yetişkinlerle yapılan görüşmelerden alınan ve 52.600 sözcüklük Lancaster/IBM Spoken English Corpus (SEC) 1992'de hazırlanmıştır. Bu çalışmada da vurgu, tonlama gibi konuşmaya bağlı özellikler işaretlenip, noktalama ve dil bilgisel sınıflandırma da yapılarak yoğun disk biçiminde kullanıma sunulmuştur. 900 konuşmacının 80 saatlik çeviriyazısı yapılmış konuşmalarından elde edilen ve yaklaşık 1 milyon sözcüklük bir toplama ulaşan Corpus of Spoken Amerian English (CSAE) de bir diğer önemli konuşma diline dayanan derlemdir. Bu derlem son zamanlarda yapılmış ve bilgisayarca okunur özellikli en büyük sözlü dil derlemidir. Yüz yüze görüşme, spontan konuşma ortamı, iş yaşamı konuşmaları, tartışmalar, haber yayınları, seromoniler ve radyodaki telefon konuşmaları gibi değişik türlerdeki ortamlar kaydedilmiştir. Bu özelliğiyle özellikle ağız çalışmaları için değerli veriler sağlamaktadır.

8. yüzyıl ve sonrası eski İngilizceyi derlemek üzere Toronto Üniversitesinde hazırlanan ve 1981 yılında yayınlanan Complete Corpus of Old English adlı derlem, 3.022 adet metnin bir araya getirilmesinden oluşmuştur. Elektronik olarak hazırlanan ilk tarihsel derlem Helsinki Üniversitesi'nde Helsinki Corpus of English Texts: Diachronic Part adlı derlemdir. Çalışma 1984-1991 yılları arasında tamamlanmış olup eski İngilizceden modern İngilizcenin ilk dönemlerini kapsamakta 400 örnekten ve 1.5 milyon sözcükten oluşmaktadır.

Birmingham Üniversitesi'nde hazırlanan ve Cobuild olarak bilinen Collins Birmingham University International Language Databese adlı derlem, 1960 ve sonrası dönemi kapsamaktadır. Bu derlemin oluşturulması sırasında gözetilen hedef kitle başta sözlük gereksinimi olan öğrenciler ve eğitmenlerdir. Ağız özelliklerinin dışında ölçünlü İngiliz İngilizcesinden %70, Amerikan İngilizcesinden %20 oranında metinler derlenmiştir. Derlemin % 25'lik bölümü sözlü dile ayrılmıştır. Metinler rastgele bir seçimle elde edilmemiş, daha çok popüler ve çok okunan eserlerden oluşmuştur. Üniversite arşivinde bulunan görüşme kayıtları ve ders konuşmalarıyla radyo yayınlarından elde

edilen kayıtlar sözlü bölümün malzemesini oluşturur. 1982'de derlemin sözcük sayısı 7.3 milyondur. Bu rakam 1987'de 13 milyona ulaşmıştır.

1990'da derlem projesinin yöneticisi John Sinclair, Cobuild'in sözcük sayısı bakımından daha kapsamlı bir derleme dönüşmesini sağlamak üzere The Bank of English adlı projeyi duyurmuştur. 1997'de 300 milyonu aşan derlemin günümüzde 500 milyon sözcüğü içerdiği bilinmektedir. Bu derlem bir monitör derlem özelliği taşımakta, yani sürekli artan bir içerik örüntüsü göstermektedir [28].

Derlem dil bilimi araştırmalarında adından sıkça söz edilen BNC, 1991-1995 yılları arasında ve İngiliz hükümetinin de proje maliyetinin yarısını desteklemesiyle tamamlanmıştır. 100 milyon sözcük içeren BNC, 1960-1975 arası ölçünlü İngiliz İngilizcesini temel almakta, %90'ı yazılı ve %10'u sözlü dilden derlenen malzemeyi içermektedir. Malzemenin %60'ı kitaplardan, %25'i dergilerden, %5'i broşürlerden ve %5'i de yayımlanmamış mektuplardan oluşturulmuştur. Sözlü malzeme çok çeşitli toplumsal katmanlardan derlenmiştir. Bunlar arasında ders konuşmaları, sunumlar, görüşmeler, seromoniler, politik konuşmalar, spor yorumları, söyleşi gösterileri sayılabilir. 15 ile 60 yaş arası farklı sosyo-ekonomik özellikteki 124 kişiyle yapılan 2000 saatlik görüşme kayıtları da derlemin sözlü dile dayanan içeriğinde yer almaktadır. Görüşmeciler yapılan görüşmenin kaydedildiğinden haberdar edilmiş, her bir görüşmeciyle yapılan kaydın ayrıntılı kayıt ortamları (zaman, yer, görüşmecinin mesleği, yaşı vb.) tutulmuştur. Derlemde yer alan metin örnekleri katmanlı örnekleme tekniğiyle elde edilmiş (yer, zaman, ortam ve düzey) mümkün olduğunca temsil gücü yüksek bir yapı elde etmek amacıyla kullanılmıştır. Burada amaç, yöntemin daha çok metnin kendi doğasına ve özelliklerine dayanmasıdır. SGML işaretleme diliyle işaretlenen derleme, internet üzerinde sorgu sistemiyle ulaşılabilmektedir [31].

3.6. Derlem Oluşturma Yöntemleri

Amacı belirlenmiş ve dilsel alt grupları iyi seçilmiş, temsil niteliği yeterli olan bir derlemin oluşturulmasında hangi örnekleme tekniklerinin kullanılacağı tartışmalı bir konudur. Derlemin;

- Makinece okunur olması,

- Otantik özellikli metinlerden oluşması,
- Temsil niteliğinin iyi olması bir derlem için gerekli niteliklerdir.

Bilgisayarca okunur dilsel verinin sağladığı yararlar arasında şunlar sıralanabilir:

- Verinin hızlı bir biçimde sıralanması, sorgulanması ve biçimlenmesi.
- İnsan sezgisine ve önyargısına bağlı yanlışlıkların en aza indirilmesi.
- Meta veri adı verilen verinin tanımını içeren işaretlemelerin çok daha hızlı ve otomatik yapılması.

Bilgisayarın derlem çalışmalarına sağladığı katkıyı Tognini-Bonelli dil bilim incelemesinde bir yöntemsel yön gösterici ve belirleyici olarak kabul etmektedir. Bilgisayarın belirleyici özelliğinden dolayı "bilgisayar destekli derlem dil bilimi" terimi de önerilmiştir [29].

Derlem çalışmalarında dil bilimcinin sezgisine dayanan incelemeler yerine gerçek, otantik, kullanımı temel alan dil verisinin tercih edildiği görülmektedir. Sezgiye dayalı yöntemde, araştırmacının kendi bireysel dil kullanımı, dış etkilenmeleri, önyargıları gibi incelemeyi etkileyecek değişkenler derlem tabanlı çalışmalarda sorun olmaktan çıkar. Araştırmacının sezgisine dayanan incelemede dil verisinin niceliksel yönlerinin çıkarımı da olanaklı değildir. Derlem tabanlı yöntemle veride yer alan farklılıklar hem niteliksel hem de niceliksel yönden kolay bir biçimde ortaya konulabilir. Derlem çalışmalarında geleneksel sezgiye dayalı yöntem "deneysel" bir yönteme dönüşmektedir [29].

Derlemde, *denge* (balance) konusu da derlemin sağlamlığı ve güvenirliği açısından önemlidir. Derlemin dengeli olması içerdiği metin sınıflarına (kategorilerine) ve derlemin kullanım amacına bağlıdır. BNC dengeli derlem örneği içinde en önemlilerindendir. Sadece sözlü dilden örnekler içeren CANCODE, alana özgü olan HKUST (bilgisayar bilimi derlemi) ve yalnızca yazılı dile özgü derlem olan Brown derlemi de diğer dengeli derlemler arasındadır. Bir derlemin hangi ölçütlere göre dengeli sayılacağı konusu günümüzde henüz tartışmalı konular arasındadır. BNC; American Ulusal Derlemi, Kore Ulusal Derlemi, Polonya Ulusal Derlemi ve Rusya Referans Derlemi gibi diğer ülke derlemlerinin hazırlanmasında kaynaklık etmiştir.

BNC'nin dengeli bir derlem olarak kabul edilmesi onun yapısal olarak geniş bir dil malzemesinden hazırlanması nedeniyledir. 100 milyon sözcüklük derlemin %90'ı yazılı dilden, %10'u çeviriyazısı yapılmış sözlü dilden oluşmuştur.

Bir dilin bütün yönlerini ayrıntılarıyla temsil edecek bir derlemin oluşturulmasının zorluğu, *örnekleme* konusunu derlem dil biliminin önemli bir sorunu durumuna getirmiştir. Örneklenecek malzemenin seçimi ve örnekleme yönteminin sağlamlığı tartışmalı konular arasındadır. Derlem ister durağan, ister sürekli geliştirilen bir yapıda olsun aslında genişçe bir dil nüfusundan elde edilmiş, seçilmiş örneklerdir. Temsil özelliği yeterli olan bir derlem için yapılması gereken, öncelikle dil ürününün sınırlarının belirlenmesi ve bu üründen *örnek biriminin* belirlenmesidir. Örnek birimdeki ögeler *örneklemenin çerçevesi*ni oluşturmaktadır. Örneklem birimi; kitap, dergi ya da gazete gibi metin türlerinden oluşur. Örneklem çerçevesi için örneğin Brown derlemini oluşturan Brown Üniversitesinde Kütüphanesinden yer alan kitap ve dergi koleksiyonlarının listesi verilebilir [29].

Derlem tasarımı ve örneklemesi hazırlandıktan sonra bilgisayar ortamında tutulan metin verisinin yapılandırılmamış, ham durumuyla bulunması, incelenen dil hakkında elde edilecek niteliksel ve niceliksel bilginin sağlamlığı ve güvenirliği konusunda kuşkulara yol açmaktadır. Derlemdeki dilsel verinin sağlam bir bilgiye dönüşebilmesi için bilgisayar ortamında tutulan verinin işaretlenmesi gerekmektedir.

İşaretleme, "elektronik form biçiminde depolanan belgenin içine, belgenin kendisiyle ilgili bilgilerin standart bir kodlamayla girilmesidir." [31].

Derlemin işaretlemesi, derlemin işlenmesinde ilk aşamadır. Derlemdeki belgelerin sınırlarının belirlenmesi belge üzerinde daha sonra gerçekleştirilecek işlemler için kolaylık sağlamakla birlikte anlamlı bilgilerin elde edilmesi için gereklidir. Metinlerin işaretlenmesinde, belgede geçen tablolar, paragraflar varsa yabancı dilden yapılan alıntılar yerleşim yerlerine göre işaretlenir. Belge üzerinde editör yorumları da bu süreç içinde ayrıca yer alabilir. Sözlü dilden örneklenen metinlerin çeviriyazısı yapılırken de konuşmada geçen durak yerleri, bürünsel özellikler de ayrıca işaretlenir [31].

Derlem işlemede tekli veri akışı ve çoklu veri akışı olarak ikiye ayrılabilecek bir yöntemden söz edilebilir. Tekli veri akışında, derlem metinleri tek tek ele alınır ve

metinde geçen özellikler belirlenir. Bu yöntem bilgisayarın görece zayıf donanıma sahip olduğu zamanlardan kalmadır. Doğrusal metin girdisinin her bir biriminin tek bir zaman biriminde işlendiği bir yöntemdir. Çoklu veri akışı için verilen örnek The Bank of English adlı derlemde kullanılan yöntemdir. Bu derlemde geçen metinlerdeki her bir birimin (sözcük, noktalama işareti vb.) bir numarası ve bu numaranın ilişkili olduğu bir metni bulunmaktadır, dolayısıyla, derlemde arama yapacak sözlük bilimci bulduğu birimin metnine daha kolay ve hızlı bir biçimde ulaşmaktadır. Elde edilen sonuç SGML gibi işaretleme diline çevrilebilmektedir [32].

1981'de Avrupa Bilim Kurumu bilgisayarların sözlük yayımında kullanımı ve sınırlıklarıyla ilgili bir çalıştay düzenlemiştir. Bu çalıştayda ele alınan konulardan biri çok işlevli sözlükler konusudur. Çok işlevli sözlük konusunun gündeme gelmesinde daha önce de değinildiği gibi artan bilginin disiplinler arası nitelik kazanmış olmasıdır.

Kruyt, makinece okunur sözlük ve derlemleri dil kaynakları adı verilen bir yapının içinde ele alır. Dil kaynakları içindeki dilsel veri tabanlarını da derlemler, sözlükler ve kavram sözlükleri olarak düşünen Kruyt'a göre, dilsel veri tabanları yapılandırılmış bir veri düzeni değil bir çok ilşevli kullanılabilen dil kaynaklarıdır [33].

Sözlükler, dil hakkındaki bilginin düzenli bir biçimde tutulduğu rezervuarlardır. Makinece okunur sözlükler, insanın kullanımı için elverişliyken bir bilgisayar destekli sözlük sistemi için yeterince yapılandırılmış değildir. Bilgisayarca değişik uygulamalarda insan dilinin anlaşılabilmesi için çok daha açık ve algoritmik, daha yapısal bir sözlük yapısı olmalı ve bir makinece okunur sözlüğün (MOS) bu duruma getirilmesi de uzun ve zor bir çalışma sürecini gerektirmektedir. MOS'ların bilgisayar uygulamalarında kullanılabilmesi için bu sözlüklerden bilgi çıkarımıyla ilgili bir yçalışmayı 1989'da Boguraev ve Briscoe yapmıştır. Yöntem için temel aldıkları sözlük Longman Dictionary of Contemporary English (LDOCE)'dir. Ayrıca tek bir MOS'tan bilgi çıkarımı yerine çoklu MOS'tan çıkarım için Byrd ve arkadaşları, yine Boguraev ve Briscoe ve Atkins çalışmışlardır. Bu çalışmaların tarihleri 1989-91 yılları arası olup, çalışmalardan büyük boyutlu derlemlerin önemi o zamanlar için daha iyi anlaşılmıştır. Church ve Mercer bilgisayar uygulamalarında kullanılacak sözlük yapılarının veri yönelimli olması gerektiğini belirterek kural tabanlı (rule-based) yöntemlerin, gelişen depolama

ortamları ve bilgi işlem gücünün artmasıyla daha da önem kazandığını savunmuşlardır [33].

LDOCE'den bilgi çıkarımı ve kullanımı amacıyla Alshawi 1989'da bir tanım çözümleyicisi geliştirmiştir. Kyurt'a göre, bu sistemin başarım oranı % 88'dir. Uygulama, sözlükteki tanımları anlamsal baş adı verilen bir yapıya dönüştürmektedir. Yani madde başı tanımları bir anlamsal düğüm olarak yorumlanmaktadır. Alshawi, bu uygulamada üstten-inme adı verilen bir bölümleme yöntemi kullanmıştır. Bu sözlük için bir başka uygulama örneği Pentherovdakie ve Vanderwande'nin 1993'te geliştirdiği uygulamadır. Uygulama, otomatik olarak sözlükte yer alan biçimbilgisel sözcüklerle anlamları arasındaki ilişki kurmaktadır. Türemiş madde başlarıyla bir ya da daha çok anlam arasındaki bağlantıyı ortaya çıkarmak amaçlanmıştır. Başarım oranı % 90 olarak verilmiştir.

Anlamsal ağ sözlükleri örnekleri olan WordNet ve EuroWordnet de otomatik bilgi çıkarımı çalışmalarına konu olmuştur. WordNet İngilizce için hazırlanan tek dilli, her bir sözcüğün bir kavram alanı ile ilişkilendirildiği yapısal bir sözlüktür. Psiko dil bilimi esaslarına göre hazırlanmıştır. WordNet üzerinde 2000 yılında Chai çalışmış, bu sözlüğü makine öğrenmesi yöntemiyle çözümlemeye çalışmıştır. Chai, çalışması sırasında önceden elle hazırlanmış bir etiketlenmiş eğitim setinin gerekliliğini bu çalışma için vurgulamıştır. EuroWordnet için benze bir çalışma diller arası bilgi çıkarımı amacılya Verdejo ve arkadaşları gerçekleştirmiştir. EuroWordnet İngilizce, İspanyolca gibi dillerin de bulunduğu diiler arası indeks sistemi kullanılan bir yapı özelliği gösterir [33].

Sözlük bilgisi ya da sözlüğe dayanan bilgi çıkarımı bu açıklamalarla birlikte sadece sözlük biliminin konmusu olmaktan çıkmaya başlamıştır. Sözlük, yeni yapılandırma anlayışıyla çeviriden, dil öğretimine ve derlem oluşturmaya kadar uzanan geniş bir sözlük tabanlı ortamın ana veri sağlayıcı nesnesi durumuna gelmektedir.

3.7. Derlem ve İstatistiksel Çıkarım

Metinsel çeşitliliği ve dağılımı yapılandırılmış, temsil niteliği yeterli bir derlemden istatistiksel tekniklerle dil bilimsel çözümlemelere kaynaklık edecek listeler, dağılımlar ve tablolar elde edilmektedir. Betimleyici ve yorumlayıcı istatistiksel yöntemler olarak

başlıca iki başlık altında verilen istatistik yöntemlerde ortalama, standart sapma, mod, medyan gibi merkezi dağılım eğilimleri var olan bütünün ya da veri setinde yer alan birimlerin genel olarak dağılımı hakkında bilgiler verir. Bir derlemden sadece sıklık listelerine dayalı çıkarımlar yapılması eksik sonuçlara neden olmaktadır. Birçok derlemin karşılaştırmalı bilgisine dayalı listeleri, sıklık bilgisine dayanan istatistik teknikler için girdi olabilmektedir. İstatistik çıkarımlar için hazırlanmış SPSS gibi profesyonel çözümler için hazırlanmış yazılımlar hem betimleyici hem de yorumlayıcı çalışmalar için raporlar üretebilmektedir [29].

Dilsel hipotezlerin test edilmesi için istatistiksel anlamlılık değerleri ölçüt olarak alınmaktadır. Elde edilen sonucun şansa dayalı olup olmadığı O(Olasılık) <0.05 gibi bir değerle test edilmektedir. Burada O olasılığı gösterir. Gözlemlenen değişkenlerdeki farklılıkların derecesi O <0.05 ya da kısaca 0.05'ten küçükse sonuçlar anlamlı kabul edilmekte, verideki farklılıkların şansa bağlı olmadığı belirlenmektedir. Derlem dil biliminde verinin test edilmesinde en çok kullanılan testlerden biri ki-kare testidir. Bu testte gözlemlenen değerle beklenen değer arasındaki fark ele alınır. Beklenen değerle gözlemlenen değer arasındaki fark ne kadar büyükse, farkın şans eseri olma olasılığı düşmektedir. Bu durumun tersinde de aradaki farkın şansa bağlı olduğu sonucuna ulaşılır. Sıkça kullanılan bir başka test log-likelihood testidir. Bu test, dağılımı normal olan veri üzerinde uygulanır. O değeri 0'a yaklaştıkça testin güvenirliği artmakta, böylelikle test edilen değişkenin şansa bağlı olmadığı anlaşılmaktadır [29].

Eşdizimli yapılar gibi sıklıkla bir arada bulunan yapıların istatistiksel otomatik yöntemlerle çıkarımı günümüzde oldukça önem kazanmıştır. Bu yapıların çıkarımında en çok tercih edilen test tekniği *mutual information* (MI) tekniğidir. Bu tekniğin dışında T testi ve Z testleri de özellikle T testi sıklıkla kullanılmaktadır.

MI tekniği bilgi teorisinden alınmıştır. Bu teknikte, sağında ve solunda birkaç birim bulunan merkez birimin dört birim sağı ve dört birim solundaki pozisyona bakılır. Buna *4:4 penceresi* (4:4 window) adı verilir. MI bir eşdizimlilik ölçüsü durumundadır. MI skoru yükseldikçe birimler arasındaki bağın gücü anlamlı kabul edilir ve iki birimin birlikteliğinin rastgele olmadığı sonucuna ulaşılır. Hunston, MI skorunun 3 ve daha

üzeri olmasının iki birimin eşdizimli kabul edilmesi için yeterli olacağını belirtmektedir [29].

3.8. İnternet Üzerindeki Derlem Çalışmaları

Tüm dünya üzerinde haberleşmeden basit veri iletimine kadar birçok alanı kapsayan internet, dil çalışmalarının sayısallaştırılması çalışmalarının sonuçlarının da anında kullanıcısına sunulduğu bir araştırma-geliştirme ortamı özelliğini son beş yılda kazanmaya başlamıştır.

Derlemlerin basılı metinlerden elde edilmesinin getirdiği güçlükler karşısında internette yayımlanan özellikle HTML sayfaları, internet tarayıcısının (browser) kopyalama özelliği ile kolayca bilgisayarda saklanabilen veriler olarak kullanılmıştır.

Son yıllarda giderek artan oranda makale, kitap formu standardı PDF başta olmak üzere MS Word ve diğer belge formları da derlemlere ham veriler sağlamaktadırlar. Günlük gazete ve dergilerin veri tabanına dayalı yapıları internet tarayıcısında çoğunlukla kopyalanabilir özellikte görüntülenmektedir. Bu nitelik, derlem araştırmacısı için sayfaların tek tek kopyalayabilmesini sağlamakla birlikte geçmişe dönük gazete metinlerinin toplanması ve büyük boyutlu verinin toplanmasının amaç edildiği projelerde kullanışsız duruma gelmektedir. HTML olmayan diğer belge biçimlerinin derlenmesi içinse Google gibi arama motorlarının gelişmiş arama özelliklerinin kullanıcı tarafından bilinmesi gerekmektedir. Google'da " aranacak ifade filetype:pdf", aranacak ifade "filetype:doc" gibi bazı arama motoruna özel deyimlerin girilmesi aranan belgeyi bulabilmektedir. İnternet ortamında bilimsel makalelerin yayımlanmasında PDF belgeleri tercih edilmektedir. Bu durumun yaygınlaşmasında karakter sorunlarının daha az yaşanması, belge biçiminin korunması ve işletim isteminden bağımsız bir yapısının olması rol oynamaktadır. Derlemin bu türden bilimsel belgeye dayanan verilerden oluşturulması planlanmışsa tam metin yayın yapan özgül bilimsel siteler seçilmektedir.

Derlem verisinin internet üzerinden derlenmesinde kullanılan kopyalama ve belge indirme biçimlerinde kolaylıklar sağlayan yazılımlar bulunmaktadır. Toplu sayfa indirme, sayfadaki belgeleri sitede yer alan bağlantı ağına göre site dizinine göre

kullanıcının bilgisayarına indiren yazılımlardan biri Teleport'tur. Bu yazılım aynı anda çok sayıda adresten istenilen bağlantı derinliğinde sayfa ya da belgeyi indirip depolayabilmektedir. Yazılım salt metin verisinden, diğer dosya uzantılarını (.htm, .html, .shtml vb.) desteklemektedir. Araştırmacılar için internetten metin verisi elde etmek, özellikle geniş bant bağlantılı bilgisayarlar ve depolama hacmi yüksek donanımlarla gittikçe kolaylaşmaktadır. İnternet üzerinde derlem oluşturma yöntemleri ve sorunları üzerine kuramsal bilgi içeren web siteleri yer almaktadır. Bu sitelerde genel olarak derlem tanımının hemen her zaman yer aldığı görülmekle birlikte bazı sitelerde derlem konusunda kullanıcının bilgisinin üst düzeyde olduğu kabul edilerek doğrudan uygulamaya yönelik konular anlatılmaktadır. Hazırlanan sitelerin büyük çoğunluğu İngilizce içerikli ABD ve İngiltere kaynaklı siteler olarak göze çarpmaktadır. Üniversitelere ait dil bilimi ve bilgisayar bilimi bölümleri derlem dil bilimi için ayrı merkezler kurarak bu merkezlerin web sitelerini yayına sokmuşlardır. Üniversite sitelerinde derlem dil bilimi için kurslar ve tanıtımları da bulunmaktadır. Üniversitelerde düzenlenen konuyla ilgili sempozyum duyuruları ve ilgili bağlantılar da güncel olarak sunulmaktadır [28].

Derlem hakkında kuramsal bilgiye dayanan site bölümlerinde derlemden nicelik ve nitelik araştırmalarının nasıl yapılacağına dair bilgiler özellikle İngilizce metinler üzerinden örneklerle açıklanmaktadır. Bu sitelerin hemen hepsinde derlem için kullanılan yazılım sayfalarına bağlantılar yer almaktadır.

Çevrim içi metin arşivlerinin ve koleksiyonlarının oluşturulması amacıyla İngiltere'de kurulan AHDS (Arts and Humanities Data Sevices) adlı kuruluş kendi bünyelerinde tutulacak derlemlerin oluşturulmasıyla ilgili 6 bölümlük bir kılavuz hazırlamıştır. Derlemde kullanılacak işaretleme ve etiketleme standartları, sözlü dil için malzeme toplanması ve derlemin kullanıcılar için dağıtımıyla ilgili konular bu kılavuzda ayrıntılı bir biçimde yer alıp kuruluşun sitesi üzerinden yayımlanmaktadır. Cobuild Concordance and Collocations Sampler adlı çevrimiçi yazılım, İngilizce hazırlanan ve 56 milyon etiketlenmiş sözcüğün standardı kendi içinde oluşturulmuş etiketleme sistemiyle sorgulanmasına olanak vermektedir. Sitede verilen örnekte örneğin fool|fools|fooling|fooled/VERB gibi bir sorguda fool sözcüğünün sözcükbiçimleri aranırken “|” karakteriyle bu biçimlerden herhangi birinin sonuç olarak döndürülmesi

ve fiil olarak etiketlenenler arasından seçilmesi istenmektedir. Fool sözcüğünün fiil olarak kullanılan bütün biçimleri içinse (fool|fools|fooling|fooled)/VERB. Deyimi kullanılmaktadır[27]. Bu çevrimiçi yazılımın collocation sampler bölümü istatistiksel eşdizimlilik çıkarımında çok kullanılan T testi ve mutual information yöntemi kullanılarak hazırlanmış, iki yöntemin seçenekli sorgu olarak kullanıcıya sunulduğu bir formdan oluşmaktadır. Aşağıdaki örnekte İngilizce "computer" sözcüğünün T testi ile hesaplanan eşdizimlilik sonucundan bir kesit yer almaktadır [28].

Google arama motoru gibi çalışan ancak interneti başlı başına bir derlem gibi kullanmayı amaçlayan WebAsCorpus, sorgulaması yapılan sözcüklerin geçtiği bağlamları dizinlemekte, aralarında Türkçenin de bulunduğu 34 farklı dildeki web sayfalarını işlemektedir. Yazılımla; mantıksal arama (or, and operatörleri), n-gram sorgulama, sorgulanan sözcüğün değişik arama motorlarında kaç kez geçtiği gibi oldukça gelişmiş sorgulama seçenekleri sunulmaktadır. Arama sonuçları sınırlanmıştır, en fazla 500 sonuç gösterilebilmektedir. Gelişmiş arama seçeneklerinde aranan sözcüğün bağlamında bulunması istenen sözcük, sonuçlarda görüntülenmesi istenmeyen sözcükler, işlenecek en fazla web sayfası gibi yazılımın işleme yeteneğini özgülleştiren yapılar bulunmaktadır. Çeşitli veri tabanlarından terim arama motoru da yazılıma eklenmiştir [28].

Türkçede derlem kavramı ve derlem dil bilimine yönelik araştırmaların çok yeni olması, bu alanın dil bilimcileri de kapsayan geniş bir araştırma alanı olmasının dünyadaki diğer örnekleri göz önüne alındığında zaman alacağı görülmektedir. İnternet üzerinde derlem ve derlem dil bilimine yönelik Türkçe kaynaklar oldukça sınırlı kalmaktadır [53].

3.9. Türkçe için Derlem Çalışmaları

Türkçenin bilgisayarla işlenmesi üzerine ilk çalışma olarak 1976'da Aydın Köksal'ın otomatik biçim birimsel çözümleme konusunda hazırladığı “Automatic Morphological Analysis of Turkish” adlı doktora tezi öne çıkmakta, 1981'de Sagay'ın İngilizce-Türkçe arasında, 1987'de Stoop'un Almanca ve Türkçe arasında otomatik çeviri sistemi hazırlama çalışmaları ilk çalışmalar olarak görülmektedir. 1990'lı yıllarla birlikte bilgisayar destekli dil bilimi çalışmalarının tüm dünyada internet'in gelişmesiyle Türkçe bilgisayar destekli çalışmalarının da proje düzeyinde yapılmaya başlandığı görülür. Bu

çerçevede 1994'te NATO projesi olarak Oflazer ve Bozşahin'in Türkçe doğal dil işleme inisiyatifiyle ilgili projeleriyle hızlanan çalışmalar, 2000 yılında Tür'ün Türkçe için ilk istatistiksel çıkarım yöntemine dayalı doktora çalışmasıyla bir başka açıdan ele alınmıştır. Biçim birimsel ve çeviri ağırlıklı çalışmaların yanında 2001 yılında Yöndem'in hazırladığı Türkçe için söylem düzeyinde bölümleme konusundaki doktora çalışmasıyla doğal dil işleme ve derlem dil bilimi alanı, tüm dünyada olduğu gibi Türkiye'de de öncelikle bilgisayar bilimciler ve dil bilimciler arasında tanınmaya başlamıştır. Yukarıda sözü edilen çalışmaların büyük bölümünde kural tabanlı yöntemler kullanılmıştır [34].

Geleneksel dil çalışmalarında metne dayalı çalışmalar öteden beri yapılmakla birlikte, Türkiye'de, tek tek metinlerden değil de metin bütünlerinden topluca yararlanmak, bütünler arası karşılaştırma yapmak gibi derlem çalışmalarında ayrı ilgi ve çalışma alanı konuları üzerine kapsamlı olarak çalışmalar yeni başlamıştır [27]. Dil bilimciler arasında yapılan bir ankette, araştırmacıların % 41'inin çalışmalarının derleme dayalı olduğunun ortaya çıktığını belirtmektedir. Derlem olarak nitelenen bu çalışmaların büyük çoğunluğu, kişisel çabalarla elektronik formlar dışında toplanmış ve paylaşıma açık olmayan yapıdadır.

Günümüzde derlem kullanım amaçlarından biri doğal dil işleme uygulamalarına gereksinim duyduğu veri tabanını sağlamaktır. Etiketlemesi gerçekleştirilmiş bir derlem NLP uygulamalarının hata oranını düşürmektedir. Türkçe NLP uygulama çalışmalarına kaynaklık edebilecek dengeli ve temsil gücü hesaplanmış ilk derlem Orta Doğu Teknik Üniversitesinde (ODTÜ) hazırlanan ve 1999-2003 yılları arasında tamamlanan ODTÜ Türkçe Derlemi'dir. Derlem, 1990 sonrasında toplanan yazılı kaynaklara dayalı metinlerden oluşmaktadır. Derlem genel amaçlı bir derlem olarak nitelenmiş, sözlü dile dayanan örnekler bir araya getirilmemiştir. Derlem dil bilimi literatüründe denge ve temsil gücüne yönelik hazırlama ölçütleri de bu derlemin hazırlanmasında göz önüne alınmış, böylece 14 metin türü her biri 2000 sözcüklük 999 örneklem (201 kitap, 87 dergi 3 günlük gazete) bir araya getirilerek toplam 2 milyon etiketlenmiş sözcüklük derlem oluşturulmuştur.

Çizelge 3.1 ODTÜ Türkçe Derlemi Metin Türleri Dağılımı [27]

Tür	Köşe Yazısı	Haber	Roman	Öykü	Makale	Deneme	Araştırma	Gezi	Söyleşi	Diğer
%	8	42	13	11	8	7	5	2	1	3

Derlem için metin toplama, işaretleme ve sorgulamaya yönelik yazılımın geliştirilmesi çalışma planı izlenmiştir. Metinler Unicode salt metin olarak kaydedilmiş, derlemde yer alan sözcükler için Oflazer'in 1994' te hazırladığı biçim birimsel çözümleyici kullanılmıştır. XEROX Araştırma Merkezi Sonlu Durum Araçları kural tabanlı çözümleyici olarak biçim birimsel çözümlemede kullanılmıştır. Derlem daha sonra Sabancı Üniversitesi'yle ortaklaşa yürütülen bir proje kapsamında ODTÜ-Sabancı Ağaç Yapılı Derlemi adını almıştır. Bu ortak proje kapsamında 7300 cümle ve 65.000 sözcük etiketlenmiştir.

Biçimbirimsel çözümlemede birden çok sonuç veren yapılar için insan kontrolüne dayalı bir etiketleme standardı oluşturulan işaretleme yazılımında yer alarak belirlenen XML tabanlı derlem etiketleme standardı yönergesi izlenerek etiketlemenin son olarak varsa hatalarının düzeltilmesi etiketleyici kişiye bırakılmıştır.

Dil bilimciler için sorgulama yazılımında düzenli ifadeler ve mantıksal sorgulama (ve, veya) seçenekleri oluşturulmuştur. Biçim birimsel çözümleme dışında cümle düzeyinde otomatik çözümleme için de bağlılık grameri adı verilen ve Türkçe için çekim öbeklerinin belirlendiği bir form göz önüne alınmıştır. Derlem ve yazılımının internet ortamında araştırmacılara ücretsiz olarak paylaşıma açık biçimi bulunmaktadır [28].

Türkçe için derlem geliştirme çalışmalarında Çebi ve Varlıkların oluşturduğu ve TurCo adı verilen işaretlenmemiş derlemdir. TurCo'nun internet üzerinden toplanan bölümü 10 farklı web sitesinin internet tarayıcı yazılımının salt metin kaydetme özelliği kullanılarak oluşturulmuştur. Bu işlemde en büyük sorun olarak site sayfalarında geçen bağlantı bilgileri (hyperlink), tekrarlanan menü adları gösterilmiştir [35].

Derlemdeki toplam sözcük sayısı 50.111.828 olarak verilmiştir. Farklı sözcük sayısı 686.804 olarak çıkarılmıştır. Derlemin % 90.40'ı internette yayımlanan gazete, dergi gibi güncel yayınlar ve Türkiye Büyük Millet Meclisi (TBMM)'nin internet sitesinde

yayımlanmış metinlerinden ve Devlet İstatistik Enstitüsü (DİE)'nün web sayfaları taranarak elde edilmiştir. Derlemin bilgisayar ortamındaki toplam büyüklüğü 362 MB'dır. TBMM tutanaklarının yazılı ve sözlü Türkçeyi temsil ettiği derlem bilgisi olarak belirtilmektedir.

Derlemde kirlilik yaratacak unsurların yer alması, derlem ekibini ilk olarak metinlerde cümle sonu bulma algoritmalarını yazmaya yöneltmiştir. Bunun dışında sözcük kök ve gövdelerinin bulunması, sözcük türlerinin belirlenmesi ve veri tabanlarının oluşturulması çalışmalarının derlem için düşünüldüğü de belirtilmektedir.

Derlem için veri tabanı yönetim sistemi seçiminde MS-SQL ve MySQL gibi popüler yazılımların, doğal dil işleme çalışmalarının kendi özel sorunları ve çözüm gereksinimleri gereğince yetersiz kaldıkları, XML gibi işaretleme standartlarının bu nedenle TurCO'nun işaretlenmesinde kullanılacağı vurgulanmıştır [35].

Derlem geliştirme sürecinde karşılaşılan güçlüklere karşı TurCo ekibinin önerdiği çözümlerin başında dil bilimcilerle bilişim araştırmacılarının iş birliğinin sağlanması gerektiğidir. Bu konuyla ilgili olarak Türk Dil Kurumu'nun 2005 yılında Türkiye'de ilk olarak bilişim bilimcilerle dil bilimcileri bir araya getiren Bilgisayar Destekli Dil Bilimi Çalıştayı düzenlenmiştir. Çalıştay süresince Türkçe üzerine bilgisayarda geliştirilen derlemlerle, genel olarak NLP uygulama alanları tartışılmıştır [28].

BÖLÜM 4

SEKANS ETİKETLEME

4.1. Giriş

Sekans etiketleme bir giriş dizisine karşılık bir çıkış dizisinin üretimidir [36]. Makine öğrenmesinde kullanılan örüntü tanımanın bir çeşididir. Ağırlıklı olarak POS Etiketleme, NER gibi uygulamalarında kullanılır. Bir diziyi etiketleme de kullanılan algoritmalar olasılığa dayanan istatistiksel çıkarsama algoritmalarıdır. Verilen bir giriş sekansına karşılık bir çıkış sekansının üretilmesine Dizi etiketleme de kullanılan en yaygın istatistiksel modeller HMM, MEMM ve CRF sistemleridir.

Kullanım Alanları:

- POS Etiketleme (Pos Tagging)
- Terim Çıkarma (Term Extraction)
- Yüzeysel Parçalama (Shallow Parsing)
- Konuşma Tanıma (Speech Recognition)
- Bölümleme (Segmentation)
- Varlık İsmi Tanımalama (Name Entity Recognition)
- Nesne Tanıma (Object Recognition)
- Biyomedikal Varlık İsmi Tanıma (Identify Biomedical Named Entities)
- İmza Çıkarma (Signature Extraction)
- Bilgi Çıkarımı (Information Extraction)
- Anlamsal Rol Etiketleme (Semantic Role Labeling)
- Biyoinformatik (Bioinformatics)

- Tablo Çıkarımı (Table Extraction)
- Gen Tahmini (Gene prediction)
- El yazısı Tanıma (Handwriting Recognition)
- Video Analiz (Video analysis)

4.2. Sekans Etiketlemede Kullanılan Yöntemler

4.2.1. Saklı Markov Model (HMM)

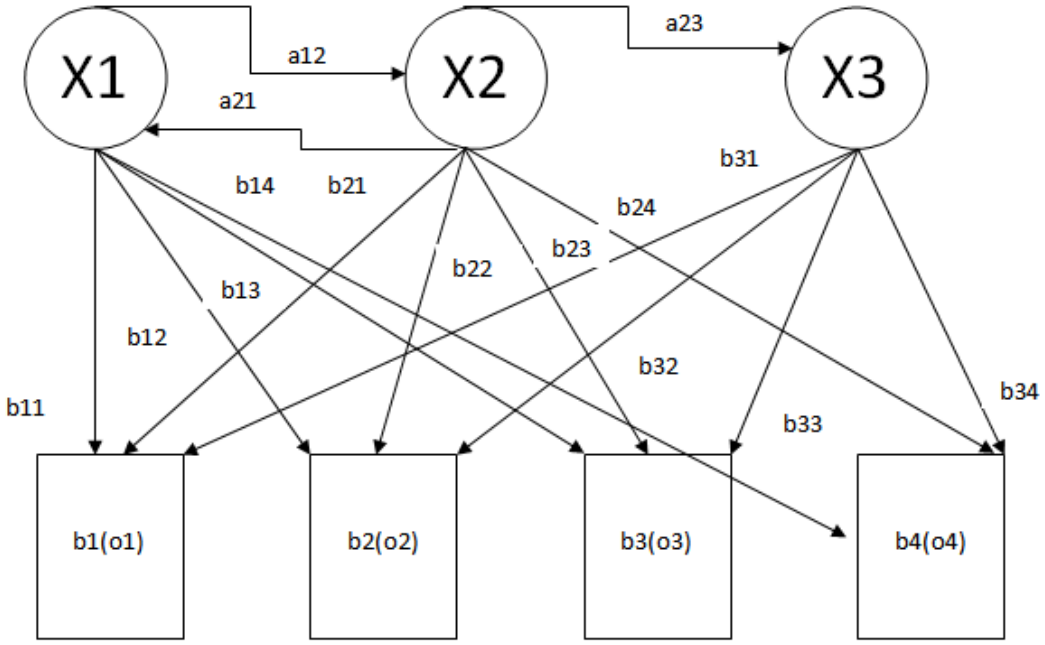
Markov analizi geçmişteki olaylardan bağımsız olarak, sadece mevcut süreç durumuna bağlı kalan sürecin, gelecekte nasıl gelişeceğini içeren olasılıkları bulunduran farklı bir özelliği vardır [37]. Saklı Markov Modeli (HMM) doğal dil isleme, ses tanıma, video isleme ve bunun gibi zamana bağlı değiskenlerin olduğu alanlarda kullanılan bağlantı tabanlı bir modeldir. SMM yapısı itibariyle Markov Model teorisine dayanmaktadır. Markov Model yapısı ardısık olarak gelen düğümlerin (çizge modeli) olasılığının bulunmasında kullanılır. Markov model yapısında düğümler sadece gözlemlenen öğeleri ifade eder.

Markov modelinde gözükmeyen bir islemin modelinden söz edilemez. Tüm Markov modeli gözlemlenen olayla bire bir örtüsür. Ancak HMM, gözlemleyemediğimiz ve varsaydığımız bağlantılar için saklı düğümler olusturarak bağlantıları bizim varsayımımıza göre yapar. Bir Markov Modeli ile bir denklemin matematiksel bir modeli çıkarılırken, HMM ile bu denkleme yakınsayan bir model çıkarılır [38]

HMM iki stokastik süreç içerir. İlk olan Markov süreci, zaman ile ilgili değişikliklerde kullanılır ve durumları içeren bir Markov zinciri üretir. Diğer süreç gözlemlenebilir olan özellik parametrelerini veya gözlemler denilen rastgele değişkenleri içerir.

HMM'in yapısı Şekil 4.1' degösterilmektedir. HMM bir durumlar zincirinden meydana gelir. HMM zinciri üzerindeki her durum kelimenin bir parçasına karsılık gelir. Her durum bir diğerine geçislerle bağlıdır. Geçisler, geçis olasılıklarına (a_{ij}) bağlı olarak durum değistirmeye imkân verir. Durumlara ilistirilen sürüm (Emission) olasılıkları (b_i) bir öznitelik vektörünün, referansın belirli bir zaman aralığıyla olan spektral benzerliğini gösterir. Sistem girdisine göre olusturulan öznitelik vektörleri dizisine bağlı olarak,

model üzerinde birinci durumdan baslayan farklı yollar izlenebilir. Bazı durumların tekrarı veya atlanması kullanıcının konusma hızındaki değişimlere sistemin adaptasyonunu sağlar. Bir kelimenin tanınabilmesi için referans olarak alınan durumdan itibaren izlenen yolun en son duruma kabul edilebilir bir olasılıkla ulaşması gereklidir [39].



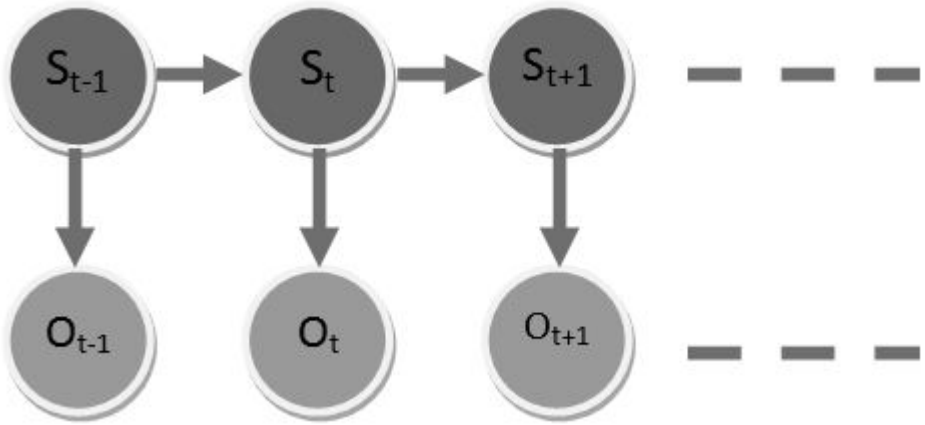
Şekil 4.1 Saklı Markov Model durum geçişleri [39]

Buna göre HMM'de gözlemler, gözlemlerden durumlara geçiş, Saklı durumlar, durumlar arası geçişler söz konusu olmaktadır.

HMM'i özetlersek [40];

- Yönlü Üretimsel (Genaratif) grafiksel model,
- Özellikler birbirinden bağımsız,
- P(s,o) maksimize etmeye çalışır,
- Gelecek gözlemleri dikkate alır,
- Gözlenen özellikler arasındaki birden fazla etkileşimi ve uzun menzilli bağımlılıkları gösteremez.

  HMM ile ilgili formül Eşitlik 4.1' de verilmektedir.



Şekil 4.2 Saklı Markov Model [40]

$$P(s,o) = \prod_{t=1}^{(\vec{o})} P\left(s_t \middle| s_{t-1}\right) P\left(o_t | s_t\right) \quad (4.1)$$

Ör: Secretariat is expected to race tomorrow.

"is" için ilgili hesaplama

x=NNP,VBZ,VBN,TO,VB,NR

$\boldsymbol{P}(is|x) = \boldsymbol{P}(is|Secretariat)\ \boldsymbol{P}(x|is)$

4.2.2. Maksimum Entropi Markov Model (MEMM)

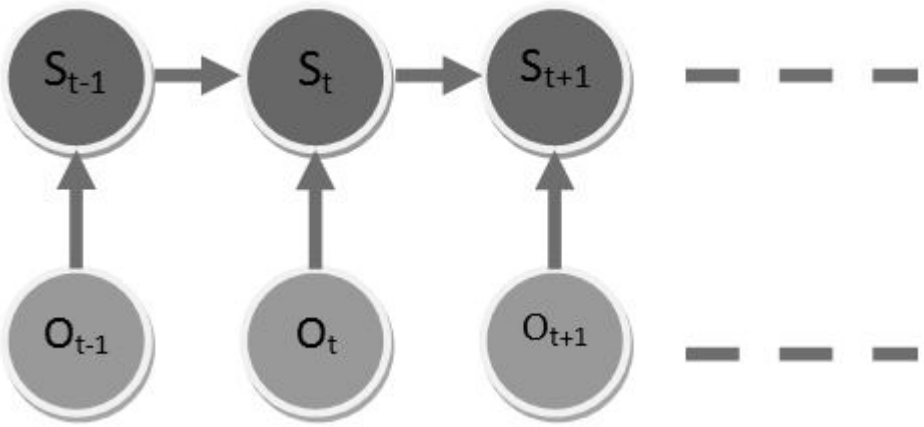
HMM'de durumlar gözlemci için saklıdır. Durumların bilinmesi çıkışların tahmin edilme başarısını artıracaktır. MEMM'de durumlar saklı değildir. Yönlü graf yapısını kullanarak durumlar arası geçiş şartlı olasılıkları ve durumların çıkışları üretme şartlı olasılıklarını birleşiminden çıkış etiketlerini bulmaya çalışır. Oluşturulan niteliklerin ağırlıklandırılması yerel yapıldığından ötürü düşük olasılıklı niteliklerde hesaba katılır. Bu durum etiket önyargısı probleminin oluşmasına ve yanlış etiketlemelere neden olmaktadır [41].

MEMM'i özetlersek [40];

- Yönlü Şartlı (Diskiriminatif) grafiksel model,
- Özellikler birbirinden bağımsız değil,
- Gelecek gözlemleri dikkate almaz,
- P(s|o) maksimizie etmeye çalışır,

- Etiket önyargı problemi vardır.

MEMM ile ilgili formül Eşitlik 4.2' de verilmektedir.



Şekil 4.3 Maksimum Entropi Markov Model [40]

$$P(s|o) = \prod_{t=1}^{(\vec{o})} \frac{1}{Z_{s_{t-1},o_t}} \exp\left(\sum_j \alpha_j f_j\,(s_t, s_{t-1}) + \sum_k \beta_k g_k\,(s_t, o_t)\right) \qquad (4.2)$$

Ör: Secretariat is expected to race tomorrow.

"is" için ilgili hesaplama

x=NNP,VBZ,VBN,TO,VB,NR

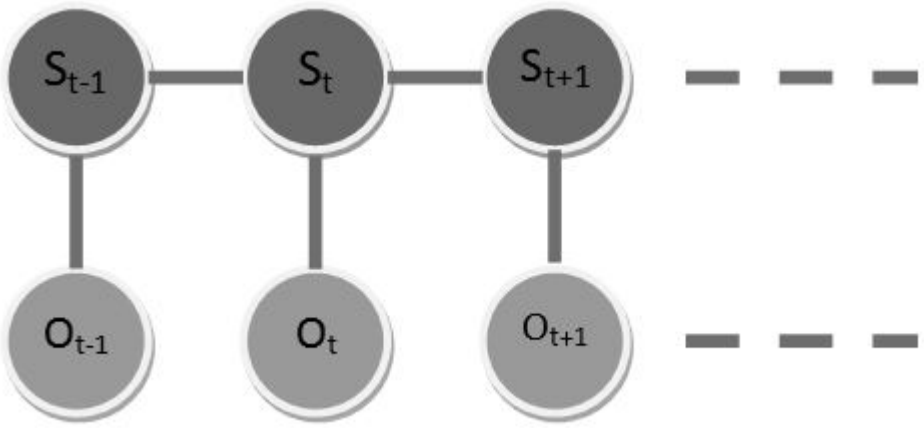
$$\boldsymbol{P}(is|x) = \exp\left(\sum_j \alpha_j f_j\,(is, Secretariat) + \sum_k \beta_k g_k\,(is, x)\right)$$

4.2.3. Şartlı Rastgele Alanlar (CRF)

CRF, Lafferty ve arkadaşları tarafından önerilen istatistiksel dizilim sınıflandırmasına dayanan bir makine öğrenmesi yöntemidir [42]. Dizilim sınıflandırıcıları bir dizilim içerisindeki her birime bir etiket atamaya çalışırlar. Olası etiketler üzerinde bir olasılık dağılımı hesaplar ve en olası etiket dizilimini seçerler. Buna göre CRF modeli p(o*|s*) olasılığını hesaplamak üzere geliştirilmiş bir olasılık modeli olarak tanımlanabilir. Burada o* =o_1,...,o_n olası çıktı etiketlerini belirtirken, s*=s_1,...,s_n giriş verilerini belirtir.

Grafiksel olarak HMM modelini ele alırsak, ard arda gelen düğümlerin birbiri ardından meydana gelme olaslıklarını etkilediği düşünülür. HMM ve MEMM gibi dizilim sınıflandırıcısı olan CRF, bir dizilim içerisindeki her bir birime etiket atamaya çalışır [43] . Olası etiketler üzerinden en olası etiket dizilimini seçer. CRF, NER, POS etiketleme, SP

vb. gibi problemlerde sıklıkla başvurulan bir yöntemdir. CRF ile ilgili formül Eşitlik 4.3' de ve şekli Şekil 4.4'de verilmektedir.



Şekil 4.4 Şartlı Rastgele Alanlar [40]

$$P(s|o) = \frac{1}{Z_{(\vec{o})}} \prod_{t=1}^{(\vec{o})} \exp\left(\sum_j \alpha_j f_j\,(s_t, s_{t-1}) + \sum_k \beta_k g_k\,(s_t, o_t)\right) \quad (4.3)$$

$Z_{(\vec{o})}$ tüm olası etiket dizileri için normalleştirme faktörüdür.

Eğitim derlemindeki her bir sözcük için nitelik fonksiyonları belirlenir. Eğitim kümesinde, nitelik fonksiyonları belirlenen sözcüklerin etiket bilgileri de mevcuttur. Buna göre nitelik fonksiyonları ve etiket dizilimleri belirlenen sözcüklerden faydalanalılarak her bir niteliğe ait ağırlık değeri hesaplanabilir. Bazı nitelikler o etiket türünü o sözcüğe atamak için yüksek ağırlıkta olabilirken, bazı niteliklerin o etiketi atamamak için ağırlığı düşük olabilir. Sistemi eğitmek sayesinde her bir nitelik için ağırlık değerlerini bulabileceğimiz bir CRF modeli oluşturulur. Eğitim sayesinde oluşturulmuş CRF modeli, daha önceden etiketlenmemiş sözcükleri etiketlemek için kullanılabilir. Her sözcüğün niteliği belirlendikten sonra, her bir niteliğin ağırlığının belli olduğu CRF modeli sayesinde, her bir sözcüğün her bir etikete atanma olasılığı hesaplanabilir [44].

Sonuç olarak en olası etiket dizilimine Y* dersek. Her bir sözcük dizilimi (o) için en yüksek olaslıklı etiket dizilimi denklem 4.4'de verildiği gibi en yüksek olasılığı seçerek bulunabilir.

$$Y^* = \text{argmax}(P(s|o)) \quad (4.4)$$

Ör: Secretariat is expected to race tomorrow.

"is" için ilgili hesaplama

x=NNP,VBZ,VBN,TO,VB,NR

$$\boldsymbol{P}(is|x) = \exp\left(\sum_j \alpha_j f_j\,(is, Secretariat) + \sum_k \beta_k g_k\,(is, x)\right)$$

CRF, MEMM'den farklı olarak etiket önyargı sorununu çözmektedir. Yani önceki sözcüklerden az bilgi taşıyanları hesaba katılmaz. MEMM'de nitelik fonsiyonlarının ağırlık değerleri normalize edilmezken, CRF'de nitelik fonksiyonlarının ağırlık değerleri normallize edilir ve bu sayede çok düşük ağırlıklı değerlerle uğraşılmamış olunur. CRF çok sayıda sekans etiketleme görevlerinde hem MEMM'ler hem de HMM'lerden daha üstündür [45][46][47].

CRF'yi özetlersek [40];

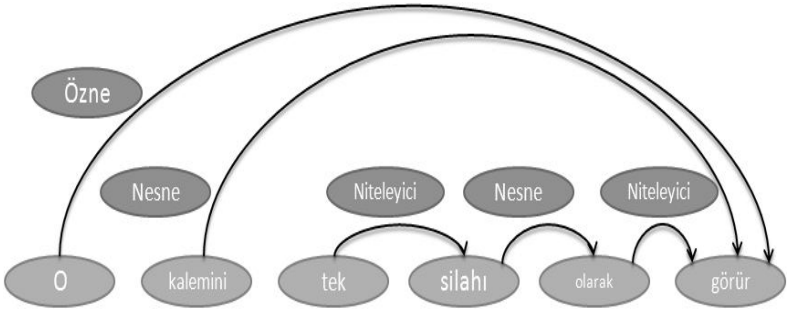
- Yönsüz Şartlı (Diskiriminatif) grafiksel model,
- Özellikler bağımsız değil,
- Gelecekteki gözlemleri dikkate alır,
- Parametre tahmini global optimumu bulmaya çalışır,
- Eğitim algoritması yavaş yakınsama sağlar,
- Parametre tahmininde büyük hesaplama maliyeti gerektirir.

CRF, hız olarak MEMM ve HMM'den yavaş olsa da doğruluk anlamında diğer iki yöntemden daha başarılıdır. CRF ve MEMM şartlı modeller olarak geçerken HMM üretimsel model olarak adlandırılır. CRF'de normalizasyon işlemi genel iken diğer iki yöntemde normalizasyon yereldir. MEMM'de HMM'den daha yavaş ama daha başarılı bir yöntemdir.

4.3. Malt Parser

Bağlılık Ayrıştırma teorisinin Tesnière'in 1959'daki çalışmasına dayandığı söylenebilir. Tesnière'e göre "Cümle, kendisini oluşturan öğeleri sözcükler olan düzenli bir topluluktur" [48]. "Zihin, cümleyi oluşturan sözcükler ve komşuları arasında ilişkiler bulur ve bu ilişkilerin bütünü cümlenin iskeletini oluşturur. Her bir ilişki bir alt terimi bir

üst terime bağlamaktadır." Günümüzde DDA alanında kullanılan Bağlılık Ayrıştırma bu ilişki bağımlı(alt terim)-sahip(üst terim) ilişkisi olarak tanımlanmaktadır [49].



Şekil 4.5 Bağlılık Grafiği [49]

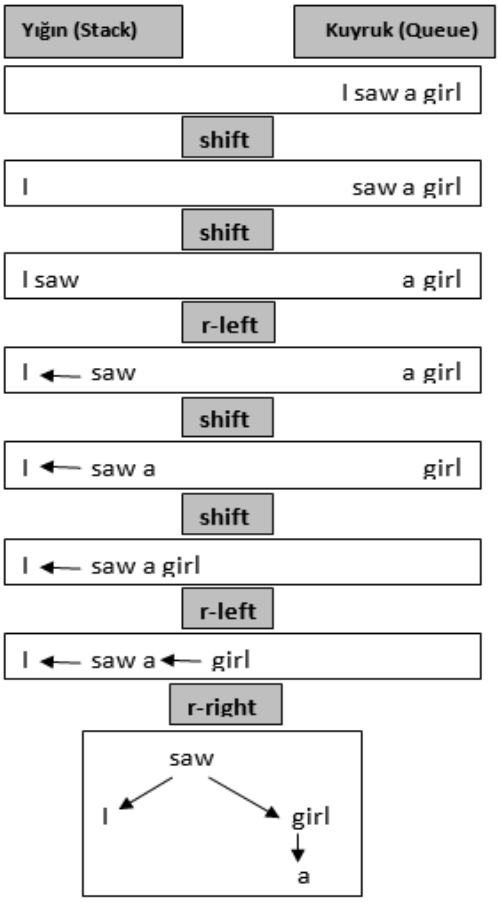
Bağlılık Ayrıştırması, metin içerisinde geçen her bir tümce için o tümceyi oluşturan sözcükler arasındaki alt terim-üst terim ilişkilerinin bulunmasıdır. Bu yöntem, üst düzey uygulamalar için anlamlı bilgi üretilmesine hizmet etmektedir. Bağlılık ayrıştırması, cümle içi sözcük dizilişleri serbest olan dillerde (Türkçe gibi) oldukça yetenekli bir yöntemdir. Cümle analizinde kullanılan bu yöntem Duygu Analizi (Sentiment Analysis-SA) uygulamaları için de bir alt basamaktır [50].

Ayrıştırcı, bir cümlenin bağlılık ağacını oluşturmak amacıyla olası bütün bağlılık kombinasyonları içerisinden optimum olanı bulmaya çalışır. Ayrıştırcıların genel olarak Ayrıştırma Algoritması ve Ayrıştırma Modelinden oluşmaktadır. Ayrıştırma algoritmaları, en iyi ağacı oluşturmak üzere ayrıştırma modelinden faydalanırlar. Son yıllara kadar, özellikle istatistiksel ayrıştırma alanında Ayrıştırma Algoritmalarında çeşitli dinamik programlama algoritmaları kullanılmıştır. Bu algoritmalar ayrıştırma modelinin kendilerine verdikleri ikili bağlılık olasılıklarını kullanarak, arama uzayında yer alan en yüksek olasılıklı ayrıştırma ağacını bulmaya çalışırlar [51].

Bu yöntemden tamamen farklı olan bir başka yaklaşım da gerekirci ayrıştırma algoritmaları kullanmaktır. Birçok çalışmada bu algoritmaların yüksek başarıları raporlanmıştır [52][53][54][55]. Gerekirci ayrıştırma algoritmaları, ayrıştırıcının her adımında bir sonraki hareketin ne olacağına ayrıştırma modeli yardımıyla karar verirler. Bu durumda ayrıştırma modeli herhangi bir makine öğrenimi sınıflandırıcısı olabileceği gibi kural tabanlı bir sınıflandırıcı da olabilir.

Malt Parser, Joakim Nivre tarafından İsveç'de Vaxjo Üniversitesinde gerçekleştirilmiş gerekirci ayrıştırma algoritması olarak Ötele-İndirge (shift-reduce) ve ayrıştırma modeli olarak Destek Vektör Makineleri (Support Vector Machine-SVM) kullanan bir programdır. DP problemlerinin çözümünde dilden bağımsız yapısıyla yüksek doğruluk değerlerine ulaşabilen popüler bir araçtır.

Ötele-İndirge algoritması, genelde cümleyi soldan sağa doğru, iki farklı veri yapısından faydalanarak ayrıştırırlar. Yığında işlenmekte olan sözcüklerin tutulurken, kuyrukta işlenmek üzere bekleyen sözcüklerin tutulur. Ayrıştırıcı her adımda üç hareketten birini uygular. Bunlar Öteleme, Soldan Sağa Bağla ve Sağdan Sola Bağla şeklindeki durumlardır. Öteleme işleminde kuyrukta bekleyen elemanın yığına itilmesi demektir. Bu önceki kelimeyle bir bağlantı oluşturulamadığı yada yığının boş olduğu durumlarda gerçekleşir. Yığındaki eleman ile sırada bekleyen eleman arasında sağa doğru bir bağ varsa Sağa Bağlama işlemi gerçekleşirken, sola doğru bir bağ varsa Sola Bağla şeklindeki işlem gerçekleştirilir [56].



Şekil 4.6 Ötele-İndirge Algoritması [56]

Ayrıştırma modeli, yığının en üstündeki ve kuyruğun en başındaki elemana bakarak bir sonraki hareketin ne olacağına karar verir. Buna sözcük bilgisine, tip bilgisine ve o ana kadar yapılan işlemleri dikkate alarak yapar. Malt Parser ayrıştıma modeli olarak SVM kullanır.

SVM, güçlü istatistiksel teoriler üzerine inşa edilmiş bir makine öğrenmesi yöntemidir. İlk kez 1995 yılında Vapnik tarafından sınıflandırma ve regresyon tipi problem çözümleri için önerilmiştir [57]. Geleneksel makine öğrenmesi yöntemlerinde çok sayıda eğitim verisine sahip olma isteği, düşük yakınsama oranı, yerel minimuma takılma ve fazla uyum-eksik uyum (overfitting-underfitting) problemleriyle karşılaşılmaktadır [58]. SVM, yapısal risk minimizasyonu temelinde çalışarak bu problemlerin üstesinden gelmiştir. SVM, yüksek boyutlu fakat az sayıda veri içeren uygulamalarda da başarılıdır [59]. Bu özelliklerinden dolayı SVM; veri madenciliği [60], müşterilerin dolandırıcılık tespiti [61] ve görüntü sınıflandırma [62] gibi birçok uygulama alanında kullanılmıştır.

BÖLÜM 5

YAPILAN ÇALIŞMALAR

5.1. Giriş

Tezin bu bölümünde tez hazırlama süresince gerçekleştirilen uygulamalar hakkında bilgiler verilecektir.

5.2. NER için Gerçekleştirilen Çalışmalar

5.2.1. NER için Eğitim Seti Boyutunun Başarıya Etkisi

Bu çalışmada eğitim setinin büyüklüğünün test seti üzerindeki başarıya etkisinin olup olmadığını görmek için bir çalışma gerçekleştirilmiştir. Kullandığımız yazılım , L-BFGS yöntemi kullanarak CRF sistemlerinde etiketleme yapmaya yarayan bir yazılımdır. Yazılım Visual Studio platformunda C# kodlarıyla gerçekleştirilmiştir [63]. Eğitim setinde 1.24 milyon cümle ve Çizelge 5. 1' deki özellik çıkarım kuralları uygulandığında 1.2 milyar özellik oluşturulmaktadır.

Çizelge 5.1 Özellik Çıkarımı için kullanılan şablon

Şablon Kuralları - Unigram ve Bigram			
#Unigram		U08:%x[0,1]	U08:NNP
U01:%x[-1,0]	U01:New	U09:%x[1,1]	U09:VBP
U02:%x[0,0]	U02:York	U10:%x[-1,1]/%x[0,1]	U10:NNP/NNP
U03:%x[1,0]	U03:are	U11:%x[0,1]/%x[1,1]	U11:NNP/VBP
U04:%x[-1,0]/%x[0,0]	U04:New/York	U12:%x[-1,1]/%x[1,1]	U12:NNP/VBP
U05:%x[0,0]/%x[1,0]	U05:York/are	U13:%x[-1,0]/%x[-1,1]	U13:New/NNP
U06:%x[-1,0]/%x[1,0]	U06:New/are	U14:%x[0,0]/%x[0,1]	U14:York/NNP
U07:%x[-1,1]	U07:NNP	U15:%x[1,0]/%x[1,1]	U15:are/VBP
		#Bigram	

1.24 milyon cümleye sahip mevcut veri setinin eğitim süresinin mevcut bilgisayarımızda çok uzun sürmesinden ötürü, ilgili eğitim seti içinden belli bir kısmı (1.veri setimizde 1545 cümle- 2.veri setimizde 4006 cümle) eğitim için seçilmiş ve sistemin eğitimi tamamlanmış ve test setinde (1050 cümle) kullanılarak sonuçlar Çizelge 5.2' de gösterilmiştir.

Çizelge 5.2 Cümle (Sekans) Sayısına göre Etiket Başarı Oranları

1.Veri Seti	**S_PER**	**B_PER**	**M_PER**	**E_PER**	**S_LOC**	**B_LOC**	**M_LOC**
Hesaplanan	153	25	0	25	788	214	6
Gerçek	269	87	3	87	996	277	22
Başarı Oranı	56.87	28.73	0	28.73	79.11	77.25	27,27
2.Veri Seti	**S_PER**	**B_PER**	**M_PER**	**E_PER**	**S_LOC**	**B_LOC**	**M_LOC**
Hesaplanan	173	38	0	38	849	241	8
Gerçek	269	87	3	87	996	277	22
Başarı Oranı	64.31	43.67	0	43.67	85.24	87.00	36.36

S_PER- Single Person, B_PER-Begin Person,M_PER-Middle Person,S_LOC-Single Location,B_LOC-Begin Location,M_LOC-Middle Location

Yapılan çalışma sonucunda elde ettiğimiz veriler bize eğitim setindeki cümle (sekans) sayısının artmasının test seti üzerindeki başarıyı artırdığı göstermiştir.

5.2.2. Şablon Kurallarının Sayısının Özellik Sayısına Etkileri

Bu çalışmada, Çizelge 5. 1'de verilen özellik çıkarmak için kullanılan şablonun bazı kısımlarının elenmesi ile oluşturulacak olan özellik sayısındaki artış yada azalışın test seti üzerindeki başarısını görülmeye çalışılmıştır.

Çizelge 5. 1 ' deki özellik şablonu kullanılmış ve 5050 cümlelik eğitim seti için farklı özellik çıkarma şablonları kullanılarak sistem test edilmeye çalışılmıştır. İlk çalışmada U01-U06 arasında bulunan özellik çıkarma şablonları kullanılmış ve 322.816 özellik çıkarılmıştır. ikinci çalışmada ise U10-U15 arasında bulunan özellik çıkarma şablonları kullanılmış ve 154.856 özellik çıkarılmıştır. Test seti olarak 5000 cümlelik set kullanılmıştır.

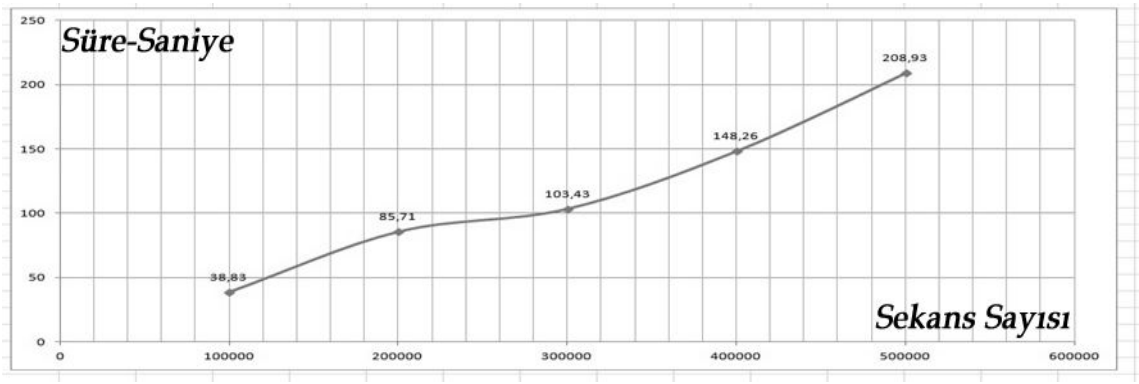
Çizelge 5.3 Şablon kural sayısına göre başarı oranları

Kural	Doğruluk Oranı
U01-U06	94.98
U10-U15	88.38

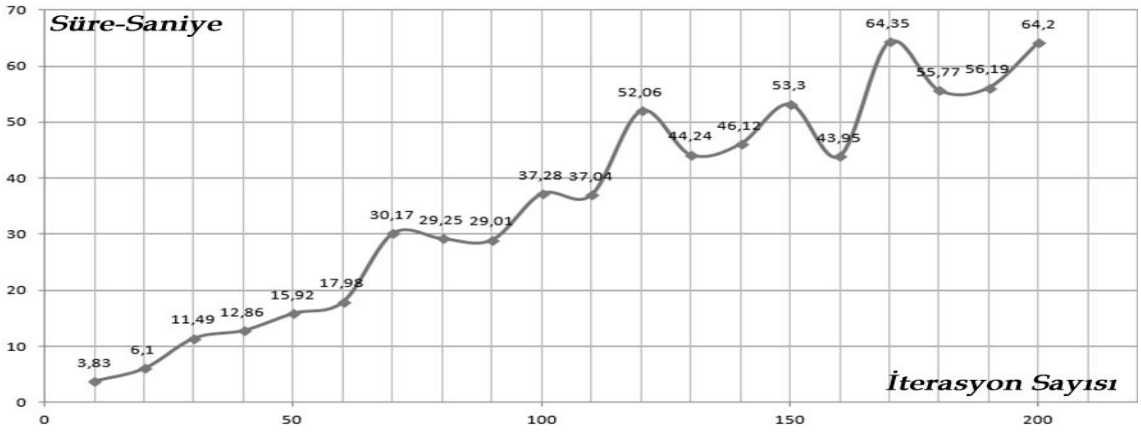
Bu çalışma sonunda, Çıkarılan özellik sayısının artırılması işlem zamanının artmasına neden olsa da test setindeki başarıyı önemli ölçüde artırdığı görülmüştür.

5.3. CRF' nin Ölçeklenebilirliği

Bu çalışmada, mevcut CRF sisteminin eğitim aşamasındaki eğitim sürelerinin hesaplanması üzerine çalışma gerçekleştirilmiştir. Burada oluşabilecek bir non-lineer durumdan yola çıkıp durumu daha lineer bir hale getirebiliriz fikri ile yola çıkılmıştır. Yaptığımız çalışmaya ait sonuçlar Şekil 5.1 ve Şekil 5.2' de gösterilmektedir.



Şekil 5.1 Sekans Sayısının Eğitim Sürelerine Etkisi



Şekil 5.2 İterasyon Sayısının Eğitim Sürelerine Etkisi

Yapılan çalışma sonucunda, mevcut CRF sisteminin yaklaşık bir lineer çalışma eğrisine sahip olduğunu görülmüştür.

5.4. CRF'nin Kollektif Öğrenme ile Kullanımı

5.4.1. Ayrık Alt Kümeler ile CRF' nin Performansı

Bu çalışmamızda, bir Kollektif Öğrenme yöntemi olan Ayrık Alt Kümeler yöntemi kullanılmıştır. Tek bir öğrenici yerine birden fazla öğrenicinin birlikte model oluşturmasına Kollektif Öğrenme adı verilmektedir. Tüm makine öğrenmesi problemlerinde (sınıflandırma, kümeleme ve regresyon) kullanılabilir.

Kollektif Öğrenme neden kullanılır sorusuna, bizlerde önemli kararlarımızı farklı zamanlarda tekrar tekrar düşünürüz, önemli kararlarımızı birçok bilene sorarız, bir hastalığın teşhisinde birçok doktordan görüş alırız vb. cevaplar verilebilir.

Veri kümemizin çok büyük olması durumunda, tek bir öğrenicinin tarafından modellenememesi gibi durumlarda veriyi alt kümelere bölerek ayrı öğreniciler tarafından modelleyerek sonuçları birleştirebiliriz.

Veri kümemiz çok küçük yada verinin karmaşıklığı yüksek ise mevcut veri kümesinden yeni alt kümeler oluşturarak, her alt kümeyi farklı öğrenici ile modelleyip sonuçları birleştirebiliriz. Veri kümemiz farklı türden bilgi(sayısal,kategorik vb.) içeriyorsa her veri türü için ayrı alt kümelere bölünüp ilgili öğrenici tarafından modellenerek sonuçları birleştirebiliriz.

Kısaca, Kollektif Öğrenme mevcut veri setini kullanarak daha doğru kararlar oluşturabilmek için kullanılır. Kollektif Öğrenmenin artılarının yanı sıra daha yavaş eğitim ve test aşamalarının olması ve daha zor anlaşılır bir model sunması eksiklikleri olarak sayılabilir.

Yaptığımız çalışmada 50.000 sekans sahip eğitim setini 2500 ve 5000 sekanslık alt Ayrık Alt Kümeler yöntemiyle öğrenmeyi gerçekleştirdik. Yaptığımız eğitim sonun da 5000 sekanslık test kümemiz üzerinde test işlemini gerçekleştirilmiş öğrenme sonuçları Çizelge 5.4' te gösterilmiştir.

Çizelge 5.4 Ayrık Alt Kümeler Doğruluk Oranları

Yöntem	Eğitim Kümesi Boyutu	Başarı Oranı
CRF	500 Cümle (10.000 sekans)	72.77
	2500 Cümle (50.000 sekans)	74.065
	5000 Cümle (100.000 sekans)	73.907
Ayrık Alt Kümeler CRF	Ayrık Alt Kümeler -5 (2500 cümlelik Eğitim Seti 500 cümlelik 5 alt küme)	74.02
	Ayrık Alt Kümeler -10 (2500 cümlelik Eğitim Seti 250 cümlelik 10 alt küme)	73.343
	Ayrık Alt Kümeler -20 (2500 cümlelik Eğitim Seti 125 cümlelik 20 alt küme)	74.057

Bu çalışma sonuncunda, Ayrık Alt Kümeler yönteminin test sonuçlarının doğruluğu azda olsa artırdığı görülmüştür.

5.4.2. Bagging ile CRF' nin Performansı

Bu çalışmada, bir Kollektif Öğrenme yöntemi olan Bagging yöntemini kullanarak CRF'nin performansını görmek istedik. Bagging (Bootstrap Aggregating) bir topluluk yöntemi olup sınıflama ve regresyon modelleri için uygulanmaktadır. Aşırı öğrenmeye karşı güçlü olan bu yöntem sınıflamada doğru sınıflama oranını arttıran ve varyansı düşüren bir yöntemdir. Veri setinde kayıp verilerin olduğu durumlarda da sınıflamada oldukça başarılıdır.

Bagging yöntemi, bir çok sınıflama modeline uygulanabilmekle birlikte daha çok karar ağaçları için kullanılmaktadır. Bagging yöntemi veri setinden sınıf yapısını bozmayacak şekilde rastgele örnekler seçilerek (bootstrap) oluşturulan çok sayıdaki karar ağacının yaptığı sınıf tahminleri oylamaya tabi tutularak en çok oyu alan sınıfı nihai sınıf tahmini olarak belirleyen öğrenme yöntemidir. Bagging yönteminde art arda oluşturulan ağaçlar önceden oluşturulan ağaçlara bağımlı değildirler ve ağaçlar orijinal veri setinden bootstrap örnekleme yapılarak oluşturulmaktadır.

Mevcut veri setindeki verilerden her defasında yerine koyarak farklı örnekler seçip yeni bir veri seti oluşturmak mümkündür. Bu şekilde yeni veri setleri oluşturmaya bootstrap metodu denilmektedir. Yapılan çalışmada 10 ve 20 farklı eğitim kümesi oluşturulup bunların performansı hesaplanmıştır. Yapılan çalışma ait bireysel başarılar Çizelge 5.5' te ve Bagging sonuçları Çizelge 5.6' da gösterilmektedir.

Çizelge 5.5 Eğitim kümelerinin bireysel başarıları

Eğitim Kümesi	2500 Sekans	5000 Sekans
1.Eğt.Kümesi	71,26	73,36
2.Eğt.Kümesi	71,98	73,24
3.Eğt.Kümesi	73,22	73,6
4.Eğt.Kümesi	72,56	73,7
5.Eğt.Kümesi	73,3	73,08
6.Eğt.Kümesi	72,92	73,68
7.Eğt.Kümesi	72,62	73,7
8.Eğt.Kümesi	73,14	73,36
9.Eğt.Kümesi	72,88	73,76
10.Eğt.Kümesi	73.1	73,6

Çizelge 5.6 Bagging İşlemi Sonunda Oluşan Başarı Oranları

Yöntem	Eğitim Kümesi Boyutu	Başarı Oranı	
CRF	50 Cümle (1000 Sekans)	71.98	
	100 Cümle (2000 Sekans)	74.28	
	500 Cümle (10.000 Sekans)	77.77	
	2500 Cümle (50.000 Sekans)	79.04	
	5000 Cümle (100.000 Sekans)	78.93	
Bagging CRF		10 Eğitim Kümesi	20 Eğitim Kümesi
	Bagging-100 (100 Cümlelik Eğitim setinden 100 cümle yerine koymalı rastgele seçim)	70.33	72.91
	Bagging-500 (500 Cümlelik Eğitim setinden 500 cümle yerine koymalı rastgele seçim)	76.37	77.42
	Bagging-2500 (2500 Cümlelik Eğitim setinden 2500 cümle yerine koymalı rastgele seçim)	78.57	78.66
	Bagging-5000 (5000 Cümlelik Eğitim setinden 5000 cümle yerine koymalı rastgele seçim)	78.83	78.78

Bu çalışma sonuncunda, bagging işlemi sonuçlarının normal eğitim sonucunda elde edilen test sonuçlarının doğruluğuna yaklaştığını ama altında kaldığı görülmüştür. Ayrıca eğitim kümesinin sayısının artırılmasının doğruluk üzerinde azda olsa başarıyı artırdığı görülmüştür.

5.5. Eksik Veri ile İlgili Çalışmalar

Bu çalışmada, sekans etiketleme problemlerinde sıkça karşılaşılabilen eksik veri oluşması durumlarında şimdiye kadar kullanılan ve önerdiğimiz yöntemlerin performanslarını görülmek istenmiştir. Veri kümelerindeki eksik değerlerle başa çıkabilmek için eksik veriyi göz ardı etmek, ilgili kaydı silmek, sıfır ile doldurmak, sık geçen ifade ile doldurmak veya ilgili kayıt ya da niteliğin satır, sütun ortalaması ile doldurulması gibi basit klasik yöntemler karmaşık hesaplama yöntemlerinin yerine kullanılmaktadır. Fakat bu gibi basit klasik yöntemler eksik gözlemleri yok sayarak verimi düşürmekte aynı zamanda var olan veriyi yanlılaştırarak sistematik anlamda kalitesizleştirmektedir.

Örneğin hava durumları kayıtlarının tutulduğu bir veri kümesinde yağmurlu havalarda ölçüm yapan metrekareye düşen yağmur miktarını ölçen ağırlık sensörü görevini yapmadığında tüm yağmurlu hava kayıtlarının silinmesi, hava durumu kayıtlarının tutulduğu bölgeye hiç yağmur yağmadığı gibi gösteren bir veri kümesi bırakabilmektedir. Yine buna benzer bir durum olarak bir ankette katılımcılardan siyasi görüş, yaş, kilo veya maaş durumları ile ilgili bilgiler istendiğinde örnek olarak katılımcılar çok yüksek ya da çok düşük maaş aldığının bilinmesinin istemediği için alakalı soruya yanıt vermekten çekinebilmektedirler. Böyle bir veri kümesinde eksik cevap yerine ortalama maaş değeri ya da nitelikte sık geçen ifadeyi (düşük, orta, yüksek) koymak yapılmak istenen anket çalışması sonuçlarına olan güveni azaltabilmektedir. Kaliteli veri madenciliği ancak kaliteli veri ile yapılabilmektedir. Bu ve benzeri durumlarda eksik olan değerin eksik olmayan diğer nitelik ve kayıtlara bakılarak hesaplanması ve böylece verinin kalitesinin artırılması daha uygun görülmektedir.

Bu çalışmada, anlatılan klasik yöntemlerin yanı sıra önerdiğimiz toplam 13 farklı eksik veri tamamlama yöntemlerini başarı oranları üzerine etkileri anlatılacaktır.

Eğitimde kullanılmak üzere 50 cümlelik(1000 sekans) eğitim seti , 5000 cümlelik (100.000 sekans) test seti kullanılmıştır (Sonuç 1). Sonuç 1 için başarı oranı 67.8'dir. Eğitim setinden % 1, % 5, % 10, % 25 ve % 50' lik çıkışları boşaltılmıştır ardından çıkışları boş olan sekanslar eğitim kümesinden çıkarılmıştır (Sonuç 2). Çıkışları boş olan sekansların çıkışına eğitim setinde en fazla olan çıkış atanmıştır (Sonuç 3). Çıkışları boş olan sekansların çıkışına ilgili cümledeki en fazla olan çıkış atanmıştır (Sonuç 4). Çıkışları boş olan sekansların çıkışına eğitim setindeki girişlere en fazla atanan çıkış atanmıştır (Sonuç 5). Çıkışları boş olan eğitim kümesinin sekansların çıkışına k koyulup eğitilip, aynı eğitim kümesi ile test edilmiş ve olasılığı yüksek olan değer çıkış olarak atanmıştır (Sonuç 6). Sonuç 6'da yapılan çalışmada çıkışa atanan k değeri arttıkça test kümesinin 1.,2.,3.,4. ve 5. öncelikli çıkışlarında hala k görülmeye devam ettiğinde bu çıkışlar rastgele çıkış değeri üretilerek eğitim yapılıp test edilmiştir (Sonuç 7). Eğitim kümesindeki her sekanstan sırasıyla birbirini takip eden 1,2,3,4,5 satır silinip eğitim kümesi oluşturulup test edilmiştir (Sonuç 8). Eğitim kümesinden elde edilen çıkışların birbirini takip etme olasılıkları çıkarılarak sırasıyla eksik veriye sahip eğitim kümelerine eklenmiş, eğitilip test edilmiştir (Sonuç 9). Eksik veriye sahip eğitim kümesinden elde edilen çıkışların birbirini takip etme olasılıkları çıkarılarak sırasıyla eksik veriye sahip eğitim kümelerine eklenmiş, eğitilip test edilmiştir (Sonuç 10). Gerçekleştirdiğimiz 9 farklı veri tamamlama yöntemlerine ait sonuçlar Çizelge 5.7' de gösterilmektedir.

Çizelge 5.7 Eksik Veri Tamamlama Çalışmaları-1

	Sonuç 1= 67.8				
	Silinen Çıkış Oranı				
Yöntem	%1	%5	%10	%25	%50
Sonuç 2	67.48	67.21	66.04	61.82	50.01
Sonuç 3	67.13	65.71	63.13	55.47	39.54
Sonuç 4	67.31	65.41	62.27	58.81	41.26
Sonuç 5	67.16	66.97	65.51	63.17	56.36
Sonuç 6	68.34	68.03	66.85	60.37	27.36
Sonuç 7	68.17	67.92	67.03	61.27	46.48
Sonuç 8	65.4	64.97	64.91	63.65	63.3
Sonuç 9	67.07	65.53	62.6	54.38	40.66
Sonuç 10	67.02	65.53	63.01	55.09	38.41

Gerçekleştirdiğimiz çalışma sonucunda, boş olan çıkışlara eğitim kümesinde en fazla bulunan çıkışın atanmasının %50'i silinmiş veri seti üzerinde en yüksek doğruluk oranına ulaşan Sonuç 5'i temel çıkış noktamız olarak kabul ederek yöntemler üzerinde değişiklik yapmaya çalışıldı. Ayrıca çıkış kümesi eleman sayısını da 3 ve 5 kabul ederek bunun sonuçlarını da görmeye çalıştık.

Gerçekleştirdiğimiz çalışmayı çıkış kümesinin eleman sayısını 3 ve 5 olacak şekilde 4 farklı yöntem için tekrarladık. Sonuç 1 için 3 çıkışta başarı oranı 61.89 iken 5 çıkış için başarı oranı 67.8 olmaktadır. Diğer sonuçlar Çizelge 5.8' de görülebilmektedir.

Çizelge 5.8 Eksik Veri Tamamlama Çalışmaları-2

Yöntem	**3 Çıkış (x,y,z) - Sonuç1 =61.89**					**5 Çıkış (x,y,z,t,u) - Sonuç1=67.8**				
	Silinen Çıkış Oranı									
	%1	%5	%10	%25	%50	%1	%5	%10	%25	%50
Sonuç 5	62.12	61.93	61.26	61.31	59.65	67.16	66.97	65.51	63.17	56.36
Sonuç 6	62.76	62.36	61.96	58.33	26.01	68.34	68.03	66.85	60.37	27.36
Sonuç 7	62.76	62.44	62.14	60.72	46.76	68.17	67.92	67.03	61.27	46.48
Sonuç 8	61.4	60.78	61.13	60.05	59.85	65.4	64.97	64.91	63.65	63.3

Yapılan bu çalışma sonucunda boş olan çıkışlara ait sekansın silinmesi eğitim başarımızı artırmış olsa da bunun elimizdeki eksik verilerin eğitim setinden çıkarılması sonucu oluştuğundan bize olumlu bir katkı sunmayacağına karar verilmiştir.

Aynı çalışmanın devamında, çıkışlara (x=1, y=2, z=3,t=4 ve z=5) değerlerini atadık. Ardından Sonuç 5'de yaptığımız gibi girişlere göre çıkış sayılarını hesaplayıp, bunları ilgili çıkış değeri ile çarpıp toplam adet sayısına bölerek ilgili çıkışı atadık (Sonuç 11). Sonuç 11'deki işlemi tüm eğitim kümesinin çıkış olasılıklarına göre hesaplayıp gerçekleştirdik (Sonuç 12). Sonuç11'deki işlemi cümle bazlı düşünerek gerçekleştirdik (Sonuç 13). Gerçekleştirdiğimiz veri tamamlama yöntemlerine ait sonuçlar Çizelge 5.9' da gösterilmektedir.

Çizelge 5.9 Eksik Veri Tamamlama-3

Yöntem	3 Çıkış (x,y,z) - Sonuç1 =61.89					5 Çıkış (x,y,z,t,u) - Sonuç1=67.8				
	Silinen Çıkış Oranı									
	%1	%5	%10	%25	%50	%1	%5	%10	%25	%50
Sonuç 5	62.12	61.93	61.26	61.31	59.65	67.16	66.97	65.51	63.17	56.36
Sonuç 11	61.79	61.32	60.13	56.91	49.67	67.07	66.18	63.45	57.19	47.5
Sonuç 12	61.68	60.98	59.57	56.09	42.41	67.1	65.68	63.01	55.5	39.63
Sonuç 13	61.68	60.98	59.57	57.92	43.94	67.27	65.88	63.03	54.44	37.72

Yapılan bu çalışma sonucunda, çıkışların ağırlıklandırılması yönteminin başarı oranlarının da temel hipotezimizden daha düşük başarı göstermesinden ötürü bu yöntemin kullanılamayacağı görülmüştür. 5 farklı çıkış için kurallarımız çalıştırıldığında oluşturulan veri setinin % 78'i oluşurken kalan % 22'lik kısım rastgele çıkışlar ile üretilmiştir. 3 Çıkış ise kurallar ile veri setinin % 71'i oluşturulurken kalan % 29'luk kısım rastgele atanmıştır.

5.6. CRF' nin Yarı-Eğiticilli Öğrenme Performansı

Bu çalışmada, CRF' nin performansını yarı eğiticili öğrenme ile eğiterek görmek istedik. Makine öğrenmesi içerisinde eğiticili sınıflandırma yaklaşımları, sınıflandırma modeli oluşturmak için verilere ilişkin özellikleri (features) ve sınıf etiketlerini kullanmaktadır. Ancak sınıf etiketlerinin toplanması; konuşma tanıma, istenmeyen e-posta tespiti, medikal teşhis gibi gerçek dünya problemleri için zor, pahalı ve zaman alıcı bir süreçtir. Bu problemi çözmenin bir yolu, öğrenme süreci sırasında etiketsiz veriler ve etiketli verilerden yararlanabilecek sınıflandırma algoritmalarını geliştirmektir. Bu tür algoritmalar, makine öğrenmesi içerisinde "yarı-eğiticili sınıflandırma yaklaşımları" olarak adlandırılmaktadır.

Araştırmacılar, modelleme için yeterince etiketli veri olmadığı durumda etiketli verilerle birlikte etiketsiz verilerden de yararlanmak için oldukça iyimserdir. Problem

alanı için etkin bir yarı-eğiticili sınıflandırma modeli oluşturmak, düşünüldüğünden çok daha zordur. Birçok çalışmada etiketsiz veriler kullanıldığı durumda; sınıflayıcıların performansında iyileşmeler sağlandığını raporladığından, makine öğrenmesi topluluğunda mevcut eğilim, etiketsiz verilerden sınıflayıcılar için mümkün olan her zaman yararlanma şeklindedir. Etiketsiz veriler kullanıldığı zaman, sınıflayıcı performansında azalmaları gösteren deneysel kanıtlar mevcut olduğu için, etiketsiz verileri kullanma konusunda bu iyimser yaklaşımın değişmesi gerekmektedir.

Yaptığımız ilk çalışmada yapay eğitim ve test seti üzerinde çalışma gerçekleştirdik. Elimizde bulunan etiketli 50 cümle ile sistemi eğitip 5000 cümlelik test seti ile çalıştırdık. Ardından elimizde bulunan 500 cümlelik etiketsiz veri setinin etiketlerini CRF ile tahmin edip mevcut etiketli 50 cümle ile sistemi eğitip ardından test işlemini gerçekleştirdik. İlgili sonuçlar Çizelge 5.10' da görülmektedir.

Çizelge 5.10 CRF'nin Yarı-Eğiticili Öğrenme Sonuçları-1

Cümle Sayısı	**Başarı Oranı**
50 (Etiketli)	67.812
50(Etiketli)+500(Etiketsiz)	68.006

Yarı eğiticili öğrenme sonucunda sistemin doğruluk oranının azda olsa arttığını görebilmekteyiz.

Yaptığımız çalışmada eğitim kümesinde bulunan 50 adet etiketlenmiş veri ile 500 adet etiketlenmemiş cümleyi kullanarak bir eğitim kümesi oluşturmaya çalıştık. 500 adet etiketlenmemiş eğitim kümesinin çıkışlarının tahmini için CRF yönteminden faydalanarak en yüksek olasılığa karşılık gelen çıkışları atayarak sistemi test ettik. Ardından yaptığımız çalışmada 500 etiketsiz veriyi etiketleyip kullanmak yerine CRF sonrası oluşan cümle başarı yüzdelerine göre bazı cümleleri eğitim kümesi dahil ettik yada çıkarttık. Böylelikle normalde elde ettiğimiz başarıyı aynı seviyeye çekmeye yada geçmeye çalıştık. Sonuçlar Çizelge 5.11' de görülmektedir. Test Kümesi 2840 cümlelik bir veri kümesidir.

Çizelge 5.11 CRF'nin Yarı-Eğiticili Öğrenme Sonuçları-2

Cümle Sayısı		**Başarı Oranı**	
Etiketli	Etiketsiz	S Dahil	S Hariç
50	50	97.73	33.89
550	-	93.67	61.1
50	500	98.08	33.53
50	151	97.81	31.26
50	335	98.08	35.41
50	471	97.98	33.81
50	524	98.08	33.51
50	349	98.18	31.5
50	175	97.81	28.59
50	5000	97.66	33.16
5050	-	93.91	76.12
50	1190	97.76	30.94
50	3906	97.8	33.59
50	4707	97.72	33.44
50	3810	97.68	33.29
50	2720	97.63	32.93
50	1094	97.65	32.48
50	293	97.65	30.54

Yapılan bu çalışma sonuncunda, yarı eğiticili öğrenme ile sistemin doğruluk oranının azda olsa arttığı görülmüştür. Ayrıca önerilen cümlelerin başarı oranlarına göre eğitim setine eklenmesi ya da çıkarılması sayesinde sistemin doğruluk oranı aynı şekilde artış görülmüştür.

Bu kısımda yaptığımız bir diğer çalışmada ise çıkışlara mevcut girişlere göre frekansı en yüksek olan çıkışı atamak ile bir önceki yada bir sonraki durumdaki çıkışların atamanın bize bir katkısı olabileceğini düşünerek sistemi bir de bu şekilde çalıştırdık. Bir cümlede geçen bir kelimenin türünün bir önceki cümlede yada bir sonraki cümlede de benzer görevde geçebileceği varsayımından hareket ederek sistemi test edilmiş ve sonuçlar Çizelge 5.12' de gösterilmiştir.

Çizelge 5.12 CRF'nin Yarı-Eğiticili Öğrenme Sonuçları-3

Yüksek Çıkış Ataması					
	%1	%5	%10	%25	%50
S Dahil	90.51	90.44	90.33	90.18	89.38
S Hariç	33.06	31.13	28.91	26.88	12.11
1 önceki Çıkışın Atanması					
	%1	%5	%10	%25	%50
S Dahil	90.51	90.44	90.16	90.18	89.38
S Hariç	33.06	31.13	28.91	26.88	12.11
1 sonraki Çıkışın Atanması					
	%1	%5	%10	%25	%50
S Dahil	90.51	90.44	90.33	90.18	89.38
S Hariç	33.06	31.13	28.91	26.88	12.11

Yapılan çalışma sonucunda, önerilen yöntemin sistemin doğruluğunu artırıcı yada azaltıcı bir etkisinin olmadığı görülmüştür.

5.7. CRF'nin ML Yöntemleri ile Karşılaştırılması

5.7.1. Yapay Veri Seti için Yapılan Çalışma

Bu çalışmamızda, CRF' nin sınırlarını belirleyebilmek için eğitim setini ve test setini kendi kurallarımızla oluşturacak bir yazılım gerçekleştirdik. Sistemi 2 Giriş ve 1 Çıkış

olacak şekilde planlayarak elimizdeki hazır olan CRF sistemi üzerinde çalıştırabilmeyi hedefledik. Bu çalışmada kullandığımız girişler ve çıkış kümelerimiz,

Giriş1 Kümesi E{a, b, c, d, e}

Giriş2 Kümesi E{f, g, h, i, j}

Çıkış Kümesi E{x, y, z, t, u} ' dir.

Kural1-Kural5 arası kuralların çalıştırılması sonucunda bir çıkış değeri ataması gerçekleştirilemiyorsa Kural11 tetikleniyor ve mevcut çıkış kümemizden bir çıkış değeri çıkış olacak şekilde atanıyor. Ardından Kural6-Kural10 çalıştırılarak uygun çıkışlar üzerinde güncelleme işlemi gerçekleştiriliyor. 20 sekans oluşturulduktan sonra cümle sonuna geldiğimizi kontrol edebilmek için ". PUN Nokta" sekansı cümlenin sonuna ekleniyor.

- Kural 1- Eğer $\mathbf{x_{1(t)}=b}$ ve $\mathbf{x_{2(t)}=h}$ ise **Çıkış=x**
- Kural 2- değilse Eğer $\mathbf{x_{1(t)}=c}$ ve $\mathbf{x_{2(t)}=g}$ ise **Çıkış=y**
- Kural 3- değilse Eğer $\mathbf{x_{1(t)}=a}$ ve $\mathbf{x_{2(t)}=f}$ ise **Çıkış=z**
- Kural 4- değilse Eğer $\mathbf{x_{1(t)}=b}$ ve $\mathbf{x_{2(t)}=g}$ ise **Çıkış=t**
- Kural 5- değilse Eğer $\mathbf{x_{1(t)}=c}$ ve $\mathbf{x_{2(t)}=f}$ ise **Çıkış=u**
- Kural 6- Eğer $\mathbf{x_{1(t-1)}=c}$ ve $\mathbf{x_{2(t)}=h}$ ise **Çıkış=u**
- Kural 7- değilse Eğer $\mathbf{x_{1(t-1)}=a}$ ve $\mathbf{x_{2(t-1)}=g}$ ise **Çıkış=t**
- Kural 8- Eğer $\mathbf{x_{1(t+1)}=a}$ ve $\mathbf{x_{2(t)}=g}$ ise **Çıkış=y**
- Kural 9- Eğer $\mathbf{x_{1(t+1)}=b}$ ve $\mathbf{x_{2(t+1)}=f}$ ise **Çıkış=x**
- Kural 10- Eğer $\mathbf{x_{1(t+1)}=c}$ ve $\mathbf{x_{2(t)}=h}$ ise **Çıkış=z**
- Kural 11- değilse Çıkış=(Rastgele üretilen (x,y,z,t,u)'dan biri)
- Kural 12-20 sekansta bir ". PUN Nokta" sekansı üretiliyor

Yukarıda verilen kuralları çalıştırarak 5000 sekanslık bir eğitim seti oluşturabilmek için yazılımın kuralları çalıştırma yüzdeleri Çizelge 5.13' de verilmektedir.

Çizelge 5.13 Kural Oluşturma Yüzdeleri

Kural 1	Kural 2	Kural 3	Kural 4	Kural 5	Kural 6	Kural 7	Kural 8	Kural 9	Kural 10	Kural 11
5.96	6.17	6.34	7.19	7.3	8.21	7.08	10.64	10.32	9.58	21.3

5 elemanlı Giriş kümeleri kullanıldığında Kural 11(bir başka ifade ile rastgele değer atama)'in çalıştırılma oranları yaklaşık %50'e olmaktaydı. Biz girişleri 3'e çekerek sistemin ürettiği çıkışların rastgeleliğini azaltmak istedik. Bu çalışmada rastgelelik yaklaşık % 21 civarında gerçekleştirildi. Bir başka ifade ile CRF ile sistem çalıştırıldığında % 79 'a yakın bir doğruluk elde edilmesi gerekmekteydi. Yapılan çalışmada 5 girişe sahip ve 3 girişe sahip eğitim setleri 5 kural ve 10 kural halinde çalıştırılarak sonuçlar Çizelge 5. 14' te gösterilmiştir. Eğitim ve Test Kümesi sekans sayıları ve frekansları Çizelge 5. 15' de verilmiştir. Eğitim Seti 5000 cümle ve Test seti 50 000 cümleliktir.

Çizelge 5.14 Kural ve Giriş Sayısının Doğruluğa Etkileri

Sekans Sayısı	**7 Kural-Giriş Kümeleri 5 elemanlı**	**12 Kural-Giriş Kümeleri 5 elemanlı**	**12 Kural-Giriş Kümeleri 3 elemanlı**
1000	31.2	35,.17	66.95
5000	32.49	40.13	71.37
10000	34.69	42.01	72.89
20000	34.56	43.23	74.16
30000	35.06	44.03	74.14
40000	35.14	44.61	74.04
50000	35.48	45.16	74.42

Çizelge 5.15 Eğitim ve Test Kümesi Frekansları

1.Giriş	Eğitim Seti	Test Seti
a	1713	16675
b	1655	16654
c	1632	16671
2.Giriş		
F	1649	16606
G	1653	16796
H	1698	16598
Çıkış		
x	1093	10587
y	1011	10524
z	1219	12127
t	880	8671
u	797	8091

Bu çalışma sonucunda, kural sayısı arttığında bir bakıma rastgelelik azaltıldığında sistemin doğruluk oranı artış göstermektedir. Giriş sayısının azalmasında aynı şekilde olasılıksal olarak sistemin belirsizliğini düşürdüğünden başarıyı önemli ölçüde artırmıştır.

Yapay verisetleri üzerinde yaptığımız bir diğer çalışmada, CRF ve Klasik Makine Öğrenmesi (KMÖ) yöntemlerinin karşılaştırılması yapılmıştır. Çalışmada Yapay Veri seti üzerinde CRF ve KMÖ yöntemlerinin performansı ölçülmüştür. KMÖ yöntemleri için WEKA programı [64], CRF için CRFSHARP programı [63] kullanılmıştır.

Çalışmanın bu aşamasında yapay veri setleri oluşturmak için kurallar tanımlanmıştır. 4 farklı kural tanımlama şekli belirlenerek 4 farklı yapay veri seti oluşturulmuştur. Çalışmanın bir sonraki araştırma alanı gerçek veri setleri üzerinde Varlık İsmi Tanımlama belirlendiğinden veri setleri 2 girişli ve 1 çıkışlı olacak şekilde tasarlanmıştır.

1.Veri setini oluşturmak için elimizde ;

Giriş Kümesi-1 {g1[0], g1[1], g1[2]} , Giriş Kümesi-2 {g2[0], g2[1], g2[2]}

Çıkış Kümesi E{x, y, z, t, u}, şeklinde kümelerin olduğu varsayımından yola çıkarak aşağıdaki kural oluşturma şablonu kullanılarak veri seti oluşturulmuştur.

- Kural 1- Eğer $\mathbf{x_{1(t)}=g1[1]}$ ve $\mathbf{x_{2(t)}= g2[1]}$ ise **Çıkış=x**

- Kural 2- değilse Eğer $\mathbf{x_{1(t)}}$**= g1[2]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=y**
- Kural 3- değilse Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=z**
- Kural 4- değilse Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=t**
- Kural 5- değilse Eğer $\mathbf{x_{1(t)}}$**= g1[2]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=u**
- Kural 6- Eğer $\mathbf{x_{1(t-1)}}$**= g1[2]** ve $\mathbf{x_{2(t)}}$**= g2[2]** ise **Çıkış=u**
- Kural 7- değilse Eğer $\mathbf{x_{1(t-1)}}$**= g1[0]** ve $\mathbf{x_{2(t-1)}}$**= g2[1]** ise **Çıkış=t**
- Kural 8- Eğer $\mathbf{x_{1(t+1)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=y**
- Kural 9- Eğer $\mathbf{x_{1(t+1)}}$**= g1[1]** ve $\mathbf{x_{2(t+1)}}$**= g2[0]** ise **Çıkış=x**
- Kural 10- Eğer $\mathbf{x_{1(t+1)}}$**= g1[2]** ve $\mathbf{x_{2(t)}}$**= g2[2]** ise **Çıkış=z**
- Kural 11- değilse Çıkış=(Rastgele üretilen (x,y,z,t,u)'dan biri)

Bu çalışmada Çizelge 7.25' deki kurallar yardımıyla türetilen kurallar kullanılarak 50,100,500,2500 ve 5000 cümlelik eğitim ve 2500 cümlelik (52500 sekans) test seti kullanılmıştır. Yapılan çalışmaya ait sonuçlar Çizelge 5.16 ve Çizelge 5.17' de görülmektedir.

Çizelge 5.16 Çalışma sonuçları

Cümle Sayısı	CRF	NB	IBK	Bag.	J48
50	68.56	64.37	80.93	79.79	78.68
100	70.74	67.11	82.29	81.35	80.21
500	74.07	66.63	82.55	82.56	82.68
2500	75.28	66.72	82.78	82.78	82.83
5000	75.17	66.69	82.87	82.9	82.84

Çizelge 5.17 CRF Karışım Matrisi

	x	**y**	**z**	**t**	**u**	**Nokta**
x	1467	744	0	1040	0	0
y	0	1434	796	867	0	0
z	815	814	1644	0	0	0
t	605	0	750	1907	0	0
u	0	0	0	0	1617	0
Nokta	0	0	0	0	0	725

Çizelge 5.16' dan da görüldüğü üzere CRF, Naive Bayes dışındaki makine öğrenmesi yöntemlerinden daha başarısızdır.

CRF' nin sadece "u" çıkışı için %100 başarı göstermesinin nedeni, Eğer x1(t)= g1[1] ve x2(t)= g2[1] ise Çıkış=u kuralının dışında hiçbir kuralın u çıktısı üretmemesidir. Aynı durum Nokta çıktısı içinde söylenebilir. Dolayısıyla başka bir olasılık olmadığından girişler kuralı sağladığı durumda çıkış "u" olarak işaretlenir. Bundan sonraki yapılan denemeler bunu test etmek için yapılmıştır.

Oluşturulan 2. , 3. ve 4. setlere ait kural listesi aşağıda görülmektedir. Bu kurallar kullanılarak yapılan denemelere ait sonuçlar Çizelge 5.18' de görülmektedir.

- **2.Kural Kümesi Listesi**
    - Kural 1- Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=z**
    - Kural 2- Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=t**
    - Kural 3- Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=x**
    - Kural 4- Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=y**
- **3.Kural Kümesi Listesi**
    - Kural 1- Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=y**

- Kural 2- Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=x**
- Kural 3- Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=x**
- Kural 4- Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=y**

- **4.Kural Kümesi Listesi**
  - Kural 1- Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=x**
  - Kural 2- Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=y**
  - Kural 3- Eğer $\mathbf{x_{1(t)}}$**= g1[0]** ve $\mathbf{x_{2(t)}}$**= g2[2]** ise **Çıkış=y**
  - Kural 4- Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[0]** ise **Çıkış=x**
  - Kural 5- Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[1]** ise **Çıkış=x**
  - Kural 6- Eğer $\mathbf{x_{1(t)}}$**= g1[1]** ve $\mathbf{x_{2(t)}}$**= g2[2]** ise **Çıkış=y**

Çizelge 5.18 İkinci Çalışma Sonuçları

Kural Kümesi	CRF	NB	IBK	Bag.	J48
2	100	100	100	100	100
3	52	52	100	100	100
4	100	100	100	100	100

Çizelge 5.19 2.Kural Kümesi Oluşturma Matrisi

	g2[0]	g2[1]
g1[0]	z	t
g1[1]	x	y

Çizelge 5.20 3.Kural Kümesi Oluşturma Matrisi

	g2[0]	g2[1]
g1[0]	y	x
g1[1]	x	y

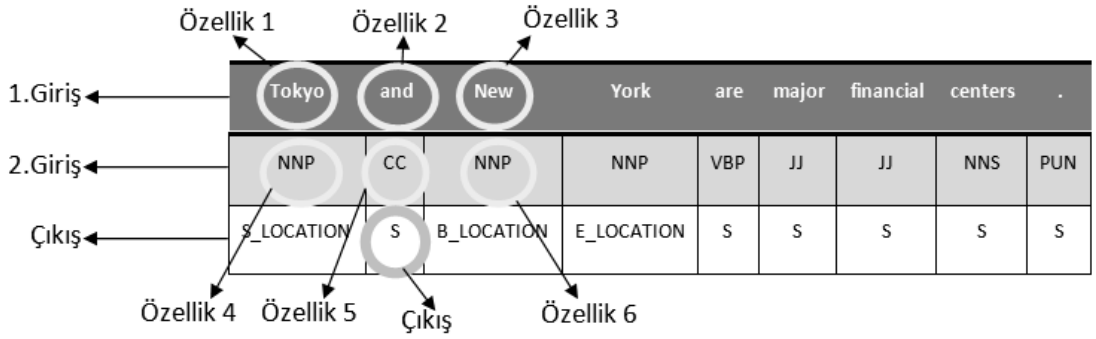
Çizelge 5.21 4.Kural Kümesi Oluşturma Matrisi

	g2[0]	**g2[1]**	**g2[2]**
g1[0]	x	y	y
g1[1]	x	x	y

Çizelge 5.19 ve Çizelge 5.21' de görüldüğü gibi çıkışlar tek bir dağılımdan geldiğinde CRF ve NB, KMÖ yöntemleriyle benzer başarı gösterirken, Çizelge 5.20' deki gibi çıkışların birden fazla dağılımdan geldiğinde CRF ve NB' nin başarıları düşmektedir.

5.7.2. Gerçek Veri Seti için Yapılan Çalışma

Bu çalışmada, CRFSharp sitesi üzerinden elde edilen veri seti kullanılmıştır [67]. Eğitim için 5050 cümle ve test için 2840 cümle kullanılmıştır. Yapay veri seti üzerinde yapılan çalışmada klasik makine öğrenmesi yöntemleri CRF' ye göre daha iyi sonuçlar üretmiştir. Gerçek veri seti ile yapılan uygulamada ise CRF klasik makine öğrenmesi yöntemlerinden daha başarılı sonuçlara ulaşmıştır. Bunda eğitim uzayının büyümesinin etkili olduğu sonucuna ulaşılmıştır. Eğitim uzayının geniş tutulması halinde CRF daha iyi bir optimizasyon sunmakta ve sonuçtaki başarı artım göstermektedir. Bunun için öncelikle elimizdeki eğitim ve test veri seti arff dosya formatına çevrilmesi gerekmektedir. İşlemin yapılabilmesi için pencere boyutu 3 olarak alınmış ve ona göre işlemler gerçekleştirilmiştir. Arff dosyasına çevirmek için kullanılan özellik şablonu Şekil 5.3' te ve arff dosya örneği Çizelge 5.22' de görülmektedir.



Şekil 5.3 Cümleye şablonun uygulanması

Çizelge 5.22 Arff dosyası formatı

Öz1	Öz2	Öz3	Öz4	Öz5	Öz6	Çıkış
?	Tokyo	and	?	NNP	CC	S_LOCATION
Tokyo	and	New	NNP	CC	NNP	S
and	New	York	CC	NNP	NNP	B_LOCATION

Yapılan çalışma ile ilgili sonuçlar Çizelge 5.23' te gösterilmektedir.

Çizelge 5.23 CRF ve Klasik Makine Öğrenmesi Yöntemlerinin Karşılaştırması

CRF	Bayes Net	Naive Bayes	NB Mult. Text	NB Updateable	IBK	KStar	LWL	Adaboost	At.Selected Classifier	J48	Cvparameter Sel.	Logitboost	Multischeme	OneR	Stacking	Vote	Decision Table
96,64	94,7	95,16	89,64	95,16	94,61	95,16	91,43	89,64	89,64	89,64	89,64	93,49	89,64	81,36	89,64	89,64	89,85

Gerçekleştirilen çalışma sonucunda CRF'nin KMÖ yöntemlerinden daha iyi sonuçlar verdiği görülmüştür.

5.7.3. Frekansı Düşük Olan Terimlerin Öğrenmeye Etkileri

5.7.2.'de yapılan CRF çalışmasında yaklaşık 50.000 sütunluk bir matris oluşmuştur. Her terim bu matrise bir sütun olarak eklenmektedir. Buda çalışma uzayını genişletmekte ve çalışma zamanını artırmaktadır. Bu çalışmada, mevcut eğitim ve test kümesinde frekansı düşük olan (1,10 ve 15'in altında) terimler "xxx" terimiyle değiştirilip çalışma süreleri ve başarıları ölçülmek istenmiştir. Yapılan çalışma sonuçları Çizelge 5.24' te görülmektedir.

Çizelge 5.24 Frekansı düşük olan terimlerin öğrenmeye etkileri-1

	CRF	Bayes Net	Naive Bayes	NB Mult. Text	NB Updateable	IBK	KStar	LWL	Adaboost	At.Selected Classifier	J48	Cvparameter Sel.	Logitboost	Multischeme	OneR	Stacking	Vote	Decision Table
Normal	96,64	94,7	95,16	89,64	95,16	94,61	95,16	91,43	89,64	89,64	89,64	89,64	93,49	89,64	81,36	89,64	89,64	89,85
K=1		95,3	95,05	89,64	95,05	94,59	95,1	91,37	89,64	89,64	95,08	89,64	93,41	89,64	81	89,64	89,64	93,91
K<=10		94,13	94,07	89,64	94,07	94,24	94,31	91,16	89,64	94,37	94,59	89,64	93,54	89,64	79,45	89,64	89,64	86,22
K<=15		93,72	93,81	89,64	93,81	94,07	94,04	91,12	89,64	89,64	94,31	89,64	93,6	89,64	79,09	89,64	89,64	87,27

Elimizdeki veri setinde S etiketi frekansı yüksek olduğundan onu çıkartıp eğitim başarı değerlerini tekrar hesapladığımızda Çizelge 5.25' deki sonuçlara ulaşılmıştır.

Çizelge 5.25 Frekansı düşük olan terimlerin öğrenmeye etkileri-2 (S Hariç)

	CRF	Bayes Net	Naive Bayes	NB Mult. Text	NB Updateable	IBK	KStar	LWL	Adaboost	At.Selected Classifier	J48	Cvparameter Sel.	Logitboost	Multischeme	OneR	Stacking	Vote	Decision Table
Normal	74,3	62,5	68,9	*	68,9	66,1	67	*	*	*	*	*	50	*	61,5	*	*	54,1
K=1		67,8	69,3	*	69,3	63,7	64,6	*	*	*	56,4	*	49,8	*	57,2	*	*	52,2
K<=10		63,3	66,2	*	66,2	58,4	55,7	*	*	37,02	53,7	*	49,5	*	41,5	*	*	37,7
K<=15		60,2	64,5	*	64,5	56,8	52,9	*	*	33,1	51,4	*	50,3	*	37,5	*	*	36,3

Bu çalışmada kullanılan 5050 eğitim cümlesi üzerinde uygulanan Çizelge 5.1'deki özellik şablonu ile 512421 özellik oluşturulmuştur.

Terim sayısı belli frekansın altında olan terimlerin değiştirilmesi sonucunda yeni oluşan eğitim ve test setindeki farklı terim sayılarını Çizelge 5.26' da görülebilmektedir.

Çizelge 5.26 Terim sayısı azaltılmış Eğitim ve Test Seti (CRF için)

	Eğitim			Test		
	1.giriş	2.giriş	Çıkış	1.giriş	2.giriş	Çıkış
Normal	9741	37	13	9809	37	13
K=1	4015	37	13	8285	37	13
K<=10	496	37	13	5854	37	13
K<=15	325	37	13	5688	37	13

Yapılan çalışmalara ait süre ve başarı sonuçları Çizelge 5.27' de gösterilmektedir.

Çizelge 5.27 Süre ve Başarı Oranları

	Başarı %	S hariç %	Süre	Özellik sayısı
Normal	96.64	74.31	4dk 15	512421
K=1	96.75	75.23	4dk 30	527813
K<=10	96.42	73.32	3dk 52	323154
K<=15	95.96	69.69	3dk 50	282139

Sonuç olarak frekansı az olan terimlerin eğitim setinden çıkarılmasının eğitim süresi üzerinde azalan yönde bir etki ettiği söylenebilir.

5.8. CRF ile Yan Cümleciklerin Tespiti ve Öğelerine Ayırmaya Etkisi

Cümle, bir duyguyu, bir düşünceyi, bir isteği veya bir yargıyı belirten söz dizisidir. Bu söz dizisinde bir veya daha fazla yargı bulunabilir. Yüklemi birden fazla da olabilir. O zaman cümledeki yargı cümlenin yapısıyla ilgilidir. Cümleleri yapı bakımından incelediğimizde temel olarak 2 gruba ayrıldığını görürüz.

İçinde tek bir yargı bulunan cümlelere basit cümle , içinde birden fazla yargı bulunan cümlelere bileşik cümle olarak isimlendirilmektedir.

Basit Cümle Örnekleri;

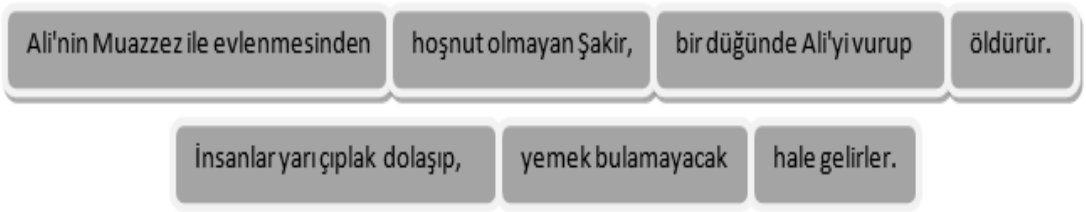
- Derslerine çok çalışmalısın.
- Ağaç sevgisi her insanda olmalıdır.

Bileşik Cümle Örnekleri

Şimdi memleketimde olmak (Yan Cümlecik) / varmış.(Temel Cümlecik)
Metni güzelce okuyarak (Yan Cümlecik) / anlamaya çalış. (Temel Cümlecik)

5.8.1. CRF ile Yan Cümleciklerin Tespiti

Biz bu çalışmamızda elle etiketlenmiş cümleleri eğitim aşamasında kullanarak etiketsiz cümleler üzerindeki yan cümlecikleri bulmayı hedefledik. Öncelikle çeşitli haber sitelerinden ve romanlardan edindiğimiz 1278 cümleyi elle yan cümleciklerine ayırdık. Şekil 5.4' te örnek bir gösterim görülmektedir.



Şekil 5.4 Yan cümleciklerine ayrılmış örnek cümleler

Ardından elimizdeki cümleleri Zemberek Doğal Dil İşlemi kütüphanesini kullanan FatihParser programı kullanılarak cümle içindeki kelimelerin çözümlemesi gerçekleştirilmiştir.

FatihParser Türkçe ve diğer Türkî diller için tasarlanmış bir sözdizimsel çözümleyicidir [65]. Yani cümleyi sözdizimsel bileşenlerine ayıran bir araçtır. Bu bileşenler öğeler, unsurlar, kelime grupları ve kelimelerdir. FatihParser dilbilimcilerin daha aşina oldukları bir ifade ile "cümle tahlili" yapan bir yazılımdır.

FatihParser programı ile elimizdeki 1278 cümledeki her kelimenin tek tek analizi yapılmıştır. Şekil 5.5' te örnek cümlelerdeki kelime çözümlemeleri görülebilmektedir.

Şekil 5.5 Kelime Analizi yapılmış örnek cümleler

1278 cümle sisteme verilerek otomatik olarak etiketlenmesi sağlanmıştır. Geliştirilen program aracılığıyla otomatik olarak etiketlenen cümleler, CRF sistemine verilebilecek formata çevrilmesi sağlanmıştır. Etiketletmede kullanılan terimlerin açıklamaları Çizelge 5.28' de gösterilmektedir. Otomatik olarak etiketlenen ve CRF sistemine verilebilecek hale getirilen cümle örnekleri Çizelge 5.29' da görülebilmektedir.

Çizelge 5.28 Etiketlerin Açıklamaları

Etiket	Tanımı
Başla	Yan/Temel cümlenin başladığını belirtir
Devam	Yan/Temel cümlenin devam ettiğini belirtir
Bitti	Yan/Temel cümlenin bittiğini belirtir
boş	Cümle içindeki boşlukları belirtir
Noktalama	noktalama işareti olduğunu belirtir

Çizelge 5.29 Otomatik Etiketlenmiş Cümle Örnekleri

Örnek Cümleler		
Giriş 1	**Giriş 2**	**Çıkış**
çocuk	isim	Basla
un	isim_tamlama-ın	Devam
bos	bos	bos
ad	isim	Devam
ı	isim_belirtme	Devam
bos	bos	bos
yusuf	Özel_isim	Devam
tur	isim_tanımlama_dır	Devam
bos	bos	bos
ve	conj	Devam
bos	bos	bos
öl	fiil	Devam
en	fiil_dönüşüm_en	Devam
ler	fiil_kişi_onlar_ler	Bitti
o	pron	Basla
nun	gen	Devam
bos	bos	bos
anne	isim	Devam
bos	bos	bos
ve	conj	Devam
bos	bos	bos
baba	isim	Devam
sı	isim_sahiplik_o_ı	Devam
dır	isim_tanımlama_dır	Devam
.	Nokta	Bitti

Yapılan çalışma sonucunda elimizdeki 1278 cümle önce elle etiketlenmiş, ardından FatihParser ile kelime analizi yapılmış ve son olarak da geliştirilen yazılım yardımıyla otomatik olarak etiketlenmiştir. Elimizdeki 1278 cümlenin 250'si test için 1028 tanesi eğitim için ayrılmıştır. CRF programı ile 1028 cümle eğitim için kullanılmış ve 104080 özellik çıkarılmıştır. Ardından da 250 cümle ile test aşaması gerçekleştirilmiştir. Elimizdeki eğitim setindeki etiketlerin frekansları Çizelge 5.30' da, Eğitim Cümle Sayısına göre başarı oranları Çizelge 5.31' de ve Etiket Başarı Oranları Çizelge 5.32' de gösterilmektedir.

Çizelge 5.30 Eğitim Seti Etiket Frekansları

Etiket	1026 Cümle	500 Cümle	250 Cümle	100 Cümle
Başla	2353	967	250	239
Devam	17423	7986	3778	1536
Bitti	2354	967	466	239
boş	7910	3701	1725	697
Noktalama	1500	698	336	133

Çizelge 5.31 Eğitim Seti Başarı Oranları

Eğitim Seti Cümle Sayısı	Çıkarılan Özellik	Başarı Oranı
100	21025	98.49
250	37355	98.46
500	59015	99.3
1028	104080	99.59

Çizelge 5.32 Etiketlerin başarı oranları

	Basla	**bos**	**Devam**	**Bitti**	**Noktalama**
100	91.72	100	99.94	85.42	100
250	91.38	100	100	84.83	100
500	96.11	100	100	93	100
1028	97.8	100	100	95.91	100

5.8.2. Yan Cümleciklerin Öğelerine Ayırmaya Etkisi

5.8.1' de cümleyi yan cümleciklerine ayırmayı gerçekleştirmiştik. Buradan hareketle cümleyi öğelerine ayırma işleminde cümleyi bir bütün olarak değil de yan cümleciklere bölünmüş halinin başarıya olumlu etki yapacağı düşünülmektedir. Bu çalışmada önce cümle yan cümleciklerine bölünmekte ardından cümle öğelerine ayrılmaktadır. Öğelerine ayrılan cümleler etiketlenmekte ve 1000 adet basit cümle ile eğitilmiş sistem üzerinde test edilmektedir. Test için 100 cümlelik bileşik cümle ve yan cümlelerine ayrılmış 225 cümle kullanılmıştır. Cümleleri etiketlerken kullanılan etiketlet Çizelge 5.33' te gösterilmektedir.

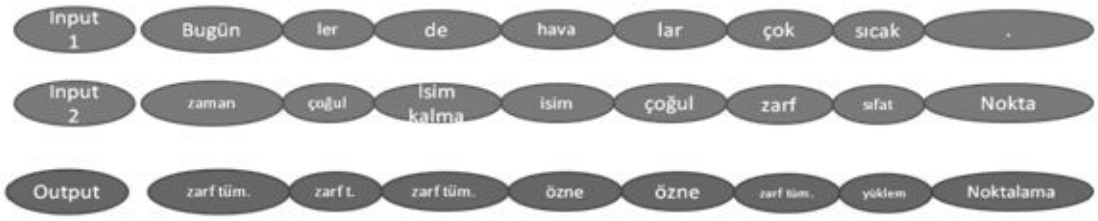
Çizelge 5.33 Kullanılan Etiketler

Etiket	Tanımı
o	Özne
bn	Belirtili Nesne
bsn	Belirtisiz Nesne
dt	Dolaylı Tümleç
zt	Zarf Tümleci
y	Yüklem
Noktalama	Noktalama işaretleri (.-, vb.)

Yan cümleciklere ayrılmış ve etiketlenmiş olan cümleler birleştirilerek 100 adet bileşik cümle haline getirilerek birinci test seti olarak kullanılmıştır. İkinci test seti olarak yan cümleciklerine ayrılmış ve etiketlenmiş veri seti kullanılacaktır. Şekil 5.6' da örnek bir etiketleme verilmiştir.

Şekil 5.6 Örnek Etiketleme

1000 adet basit cümle üzerinde yapılan morfolojik ayrıştırma işleminin ardından elle etiketleme yapılarak cümleler öğelerine ayrılmış ve CRF sistemine verilebilir bir hale gelmiştir. Aynı işlemler test setleri içinde yapılmış ve 2 ayrı test seti sisteme verilecek hale getirilmiştir. Şekil 5.7' de örnek bir eğitim seti cümlesi görülmektedir.



Şekil 5.7 Eğitim seti örnek cümle

Eğitim ve Test setleri üzerinde yapılan çalışmalar sonucunda Çizelge 5.34' teki sonuçlar elde edilmiştir.

Çizelge 5.34 Çalışma Sonuçları

Test Seti	Başarı Oranı
Zero Rule - R_0	36.15
Bileşik Cümle (Test Seti-1)	43.91
Yan Cümlecikli (Test Seti-2)	59.58

5.9. Ardışık CRF Yapısı Kullanarak SL İşlemenin Gerçekleştirilmesi

Bu çalışmada, özellikle bağlılık ayrıştırması uygulamalarında karşılaşılan Etiket ve ID bulma problemi için iki aşamalı bir sistem tasarlanıp, gerçeklenmeye çalışıldı. Bağlılık Ayrıştırmasında özellik vektörlerinin oluşturulmasında kullanılan CoNLL formatındaki kalıp Şekil 5.8' de gösterilmektedir.

ID	LEX	LEMMA	CPOS	POS	INF	DEP-ID	DEP
1	Bu	bu	DET	DET	_	2	Determiner
2	_	okul	Noun	Noun	A3sg|Pnon|Loc	3	Deriv
3	okuldaki	_	Adj	Adj	Rel	4	Modifier
4	öğrencilerin	öğrenci	Noun	Noun	A3pl|Pnon|Gen	8	Possessor
5	en	en	Adv	Adv	_	7	Modifier
6	_	akıl	Noun	Noun	A3sg|Pnon|Nom	7	Deriv
7	_	_	Adj	Adj	With	8	Deriv
8	akıllısı	_	Noun	Zero	A3sg|P3sg|Nom	14	Subject
9	şurada	şura	Noun	Noun	A3sg|Pnon|Loc	10	Locative.Adjunct
10	_	dur	Verb	Verb	Pos	11	Deriv
11	duran	_	Adj	ApresPart	_	13	Modifier
12	küçük	küçük	Adj	Adj	_	13	Modifier
13	_	kız	Noun	Noun	A3sg|Pnon|Nom	14	Deriv
14	kızdır	_	Verb	Zero	Pres|Cop|A3sg	15	Sentence
15	.	.	Punc	Punc	_	0	ROOT

Şekil 5.8 CoNNL Veri Biçimi

Öncelikle CRF'nin KMÖ yöntemlerinden daha iyi olduğunu kanıtlamak için 5635 METU-Sabancı derleminden 5000 cümle alınmıştır. 4500 cümle eğitim ve 500 cümle test seti olarak kullanılmak üzere ayrılmıştır. Girişlerimiz Şekil 5.8' deki LEX ve CPOS sütunları , çıkış ise DEP sütunudur. KMÖ yöntemleri için WEKA programı kullanılacağından elimizdeki veriler arff dosya biçimine dönüştürülmüştür. Arff dosya formatına çevirme işlemi için Şekil 5.3' teki özellik şablonu kullanılarak dosya formatı olarak Çizelge 5.22' deki hale getirilmiştir. WEKA ve CRFSHARP üzerinde sistem çalıştırılarak sonuçlar Çizelge 5.35' te gösterildiği gibidir.

Çizelge 5.35 DEP için CRF ve KMÖ karşılaştırması

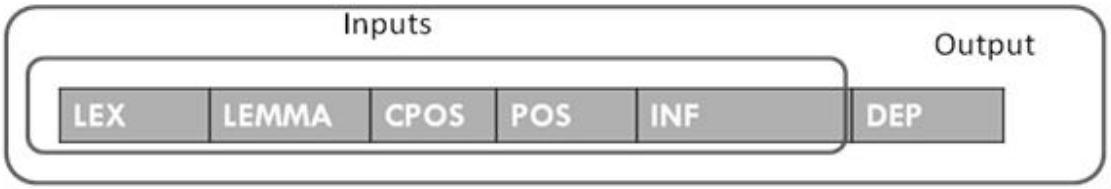
Yöntem	CRF	Naive Bayes	Bayes Net	IBK	KStar	LogitBoost	Decision Table
Başarı Oranı	79.51	73.53	66.83	73.78	74.27	71.66	61.55

DEP için yapılan çalışma sonunda CRF'nin KMÖ yöntemlerine göre daha yüksek başarı gösterdiği Çizelge 5.35' te görülmektedir.

CRF'nin KMÖ yöntemlerinden daha yüksek doğruluklara eriştiğini gördükten sonra çalışmamızda CRF'yi kullanmayı uygun gördük. Amacımız , CRF'yi kullnarak Şekil 5.8'

deki veri yapısında görülen 7. sütundaki ID ve 8. sütundaki Etiket değerleri doğru işaretleyebilmektir. Bu problem iki aşamada çözülmeye çalışılmıştır.

CRF'nin 1.aşamasında amacımız DEP yani bağlantı türünü bulabilmektir. Bunun için CRF'nin 1.aşamasındaki eğitim seti cümlemiz için girişlerimiz ve çıkışımızın Şekil 5.9' da görülmektedir.



Şekil 5.9 CRF 1.Aşama-DEP için girişler ve çıkış

Eğitim aşamasının ardından test setimiz için DEP etiketlerimi üretilmiş ve 1.aşama sonuçlandırılmıştır. 2.Aşama da ise işimiz biraz daha karmaşıklaşmaktadır. Şekil 5.8' deki DEP-ID sütununu bulmak için birkaç işleme daha ihtiyacımız olmaktadır. DEP-ID sütunundaki sayısal değerleri CRF'nin yüksek doğrulukla bulması zordur. Burada bir dönüşüm ihtiyacımız vardır. Gerekli olan bu dönüşüm Şekil 5.10' da görülmektedir.

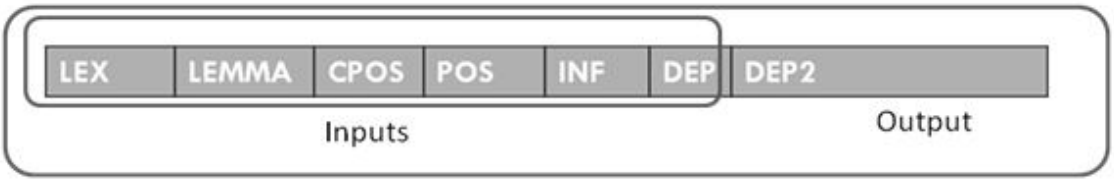
ID	LEX	LEMMA	CPOS	POS	INF	DEP-ID	DEP
1	Bu	bu	DET	DET	_	2	Determiner
2	_	okul	Noun	Noun	A3sg|Pnon|Loc	3	Deriv
3	okuldaki	_	Adj	Adj	Rel	1	Modifier

ID	LEX	LEMMA	CPOS	POS	INF	DEP	DEP2
1	Bu	bu	DET	DET	_	Determiner	Deriv
2	_	okul	Noun	Noun	A3sg|Pnon|Loc	Deriv	Modifier
3	okuldaki	_	Adj	Adj	Rel	Modifier	Determiner

Şekil 5.10 CRF 2.aşama-DEP-ID yerine DEP atanması

2.aşamada yapılan işlemler şu şekilde sıralanabilir. Öncelikle DEP-ID'ye karşılık gelen satıra gidilir. İlgili satırın DEP sütunundaki değer DEP2 sütunu olarak tanımlanır. Böylelikle DEP-ID sütunu yerine elimizde artık DEP2 adında yeni bir sütun olmuş olmaktadır. CRF ile bu aşamada elde edilen eğitim setimiz ile sistem eğitilip ardından

test işlemi uygulanarak sonuçlar elde edilmiştir. 2.Aşamada kullanılan eğitim seti giriş ve çıkışları Şekil 5.11' de görülmektedir.



Şekil 5.11 CRF 1.Aşama-DEP2 için girişler ve çıkış

2.aşama sonunda elimizde DEP ve DEP2 alanları hesaplanmış olmaktadır. Ama bizim DEP ve DEP-ID sütunlarına yani bağlantı etiketine ve ID'ye ihtiyacımız vardır. DEP sütunu elimizde olduğundan bağlantı türü için bir işlem yapmamız gerekmez. DEP2 için ise 2.aşamada yaptığımız dönüşümü tersine çevirmemiz gerekmektedir. Şekil 5.12' de bu durum görülmektedir.

ID	LEX	LEMMA	CPOS	POS	INF	DEP	DEP2
1	Bu	bu	DET	DET	_	Determiner	Deriv
2	_	okul	Noun	Noun	A3sg|Pnon|Loc	Deriv	Modifier
3	okuldaki	_	Adj	Adj	Rel	Modifier	Determiner

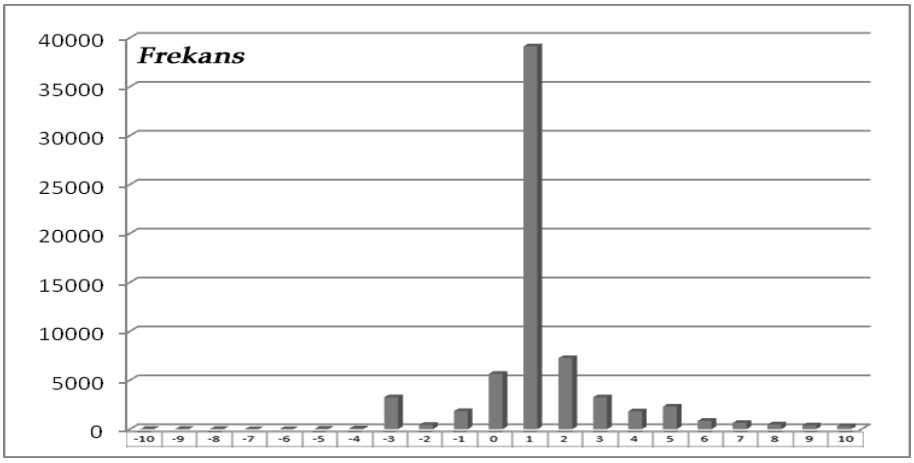
ID	LEX	LEMMA	CPOS	POS	INF	DEP	DEP2
1	Bu	bu	DET	DET	_	Determiner	2
2	_	okul	Noun	Noun	A3sg|Pnon|Loc	Deriv	3
3	okuldaki	_	Adj	Adj	Rel	Modifier	1

Şekil 5.12 DEP2 değerinin ID'ye çevrilmesi

Dönüşüm işlemi yapılırken elimizdeki Türkçe verisetinin histogramı çıkarılmıştır. Şekil 5.13' teki Türkçe için hazırlanan histogramlar göstermiştir ki, verisetimizdeki kelimeler ağırlıklı olarak sağa bağımlıdır. Bu yüzden yazılan bir program aracılığıyla mevcut kelimenin önce sağına sonra solana bakılmış ve etiket ile eşleşen satırın ID'si ilgili kelime için ID alanına eklenmiştir. Bu şekilde oluşturulan sistemin Türkçe için sonuçları Çizelge 5.36' da gösterilmiştir.

Çizelge 5.36 Türkçe için sonuçlar

Dil		AS_U	AS_L
Türkçe	Eryiğit ve Ark.[11]	76.1	67.4
	Malt Parser	77.16	67.23
	CRF	77.5	70.64

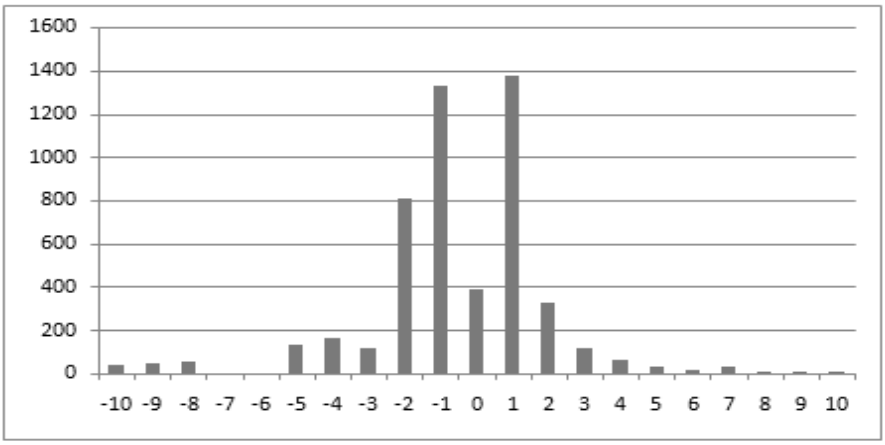


Şekil 5.13 Türkçe için Histogram Grafiği

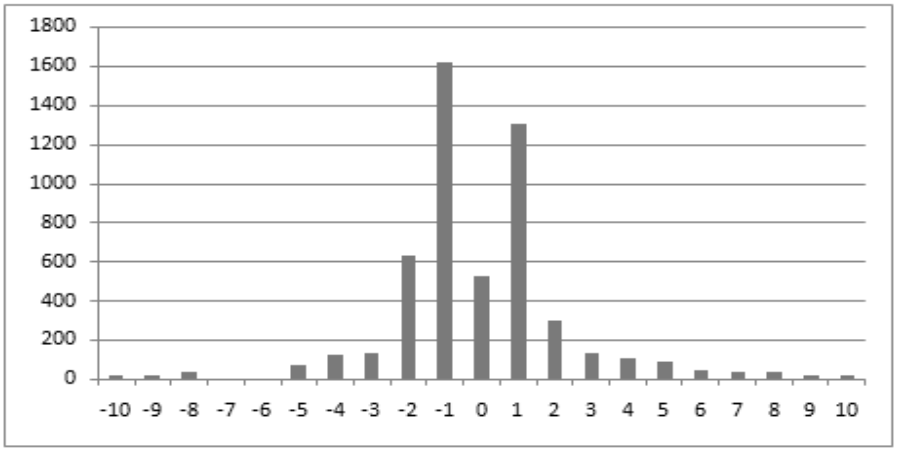
Türkçe dışında Hollanda, Danimarka,Portekiz ve İsveç dilleri içinde sistem denenmiştir. Türkçe için gerçekleştirilen işlem basamakları bu diller içinde aynen uygulanmıştır. Çizelge 5.37' de bu dillere ait bağlantı türü (DEP), AS_U, AS_L başarıları verilmiştir. Ayrıca İsveç diline ait histogram Şekil 5.14' te, Hollanda diline ait histogram Şekil 5.15' te, Danimarka diline ait histogram Şekil 5.16' da ve Portekiz diline ait histogram Şekil 5.17' de görülmektedir.

Çizelge 5.37 Türkçe Dışındaki Diller İçin Sonuçlar

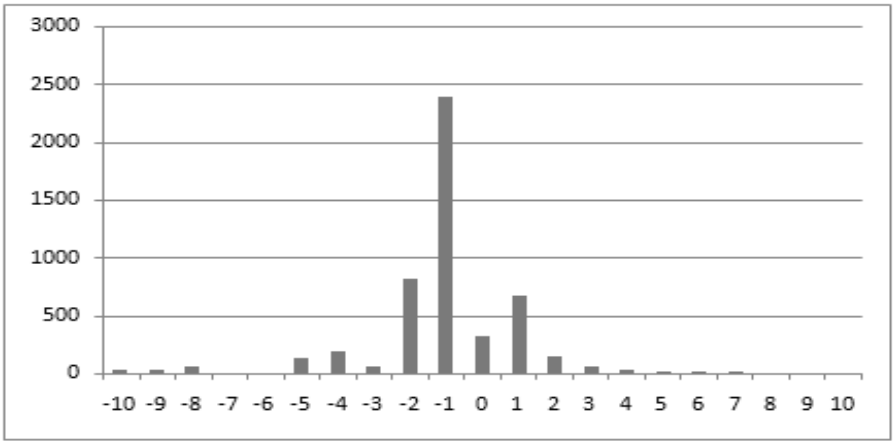
Dil	**Yöntem**	**DEP**	**AS_U**	**AS_L**
İsveç Dili	Malt parser	0.87	0.88	0.82
	CRF	0.84	0.54	0.51
Hollanda Dili	Malt parser	76.23	75.09	71.94
	CRF	81.79	42.57	39.35
Danimarka Dili	Malt parser	87.03	84.91	80.16
	CRF	87.57	37.66	35.44
Portekiz Dili	Malt parser	87.89	84.35	80.21
	CRF	88.63	44.92	42.56



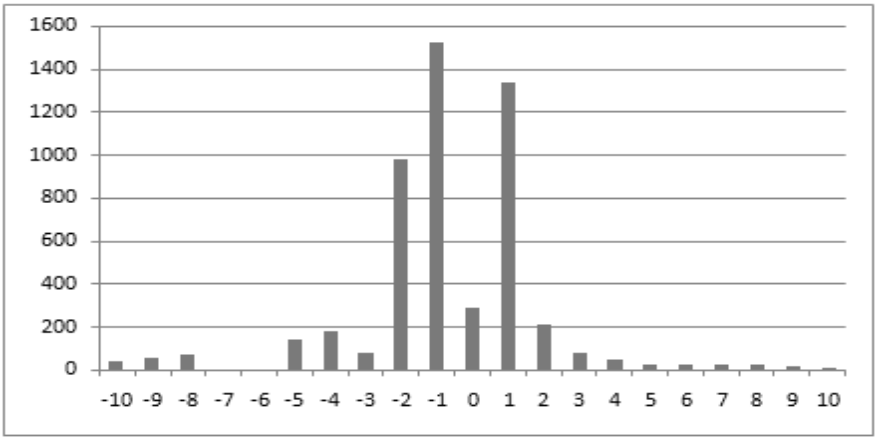
Şekil 5.14 İsveç Dili Histogramı



Şekil 5.15 Hollanda Dili Histogramı



Şekil 5.16 Danimarka Dili Histogramı



Şekil 5.17 Portekiz Dili Histogramı

Türkçe için CRF ile gerçekleştirilen çalışma sonucunda AS_U ve AS_L metriklerinde artış sağlanmıştır.

Türkçe dışındaki diller için yapılan çalışmalarda ise DEP metriği haricinde başarı düşük kalmıştır. Bunun sebebi olarak Şekil 5.14 - Şekil 5.15 - Şekil 5.16 ve Şekil 5.17' de görüldüğü üzere bu dillerin Türkçe için geçerli olan sağa bağımlılık ilkesinden farklı bir yapı göstermesidir.

BÖLÜM 6

SONUÇ VE ÖNERİLER

Sekans Etiketleme bir giriş dizesine karşılık bir çıkış dizesinin üretimidir. Giriş ve Çıkış dizesinin içeriklerine göre Doğal Dil İşlemenin birçok uygulaması (Varlık İsmi Tanımlama, Makine Çevirisi, Cümleyi Öğelerine Ayırma vb.) sekans etiketleme olarak tanımlanabilir.

Bu çalışmada Sekans Etiketleme için iki hipotez geliştirilmiş ve yapılan uygulamalarla bu hipotezler kanıtlanmaya çalışılmıştır.

Birinci Hipotezimiz , karmaşık bir problemi daha basit parçalara bölmenin problemin çözümünü kolaylaştıracağı ve bu durumun başarı oranı üzerinde olumlu etki yapacağı yönündedir. Bu hipotezimizi gerçekleştirmek için yaptığımız cümleyi öğelerine ayırma uygulamasında, bileşik bir cümleyi sisteme bir bütün haline vermektense alt parçalara (yan cümleciklere) bölmenin sistemin doğruluğunu olumlu yönde artırdığı görülmüştür.

İkinci Hipotezimizin temelinde de karmaşık bir problemi daha basit alt parçalara bölme fikri yatmaktadır. Problemi alt parçalarına bölmenin yanı sıra Bağlılık Ayrıştırması gibi 2 çıkışlı problemlerinin çözümü için ardışık olarak iki aşamada çözmenin sistemin başarısını artıracağı yönündedir.

Türkçe için yapılan çalışmalarda, Destek Vektör Makineleri (Support Vector Machine-SVM) tabanlı bir yapı kullanan Malt Parser ile belirli bir doğruluk oranlarına erişilmiştir. Malt Parser özellikle bağlılık ayrıştırması problemini çözmek için geliştirilmiş gerekirci ayrıştırma modeli olarak isimlendirilir. Gerekirci ayrıştırma algoritmaları, ayrıştırıcının her adımında bir sonraki adımı tahmin etmek için ayrıştırma modelleri kullanırlar.

Ayrıştırma Algoritması olarak ötele-indirge yöntemini kullanarak en iyi ağacı oluturmaya çalışır. En iyi ağacı oluşturmaya çalışırkende Ayrıştırma modeli olarak önce cümle içindeki kelimeler arasındaki ikili ilişkileri bularak işe başlamaktadır. Ardından bulunan ikili ilişkileri bağlantı türü ile gruplamaktadır. Sonraki aşamalarda ise bulunan her grup için olasılıksal ağırlıklar hesaplanmakta ve test kümesi üzerinde denenmektedir. Ayrıştırma modelinde sınıflandırıcı olarak SVM kullanılmıştır.

Bizim önerimizde, iki bilinmeyenli (Bağlılık Türü ve Bağlanılan Kelime-ID) bir problem olan Bağlılık Ayrıştırmasını iki parçaya bölerek çözmeye çalışılmıştır. Öncelikle sisteme kelimeler ve bağlılık türleri verilmiştir. Eğitimin ardından Test kümesi üzerinde bağlılık türleri bulunarak kaydedilmiştir. İkinci aşamada ise bağlanılan kelime (ID) yerine ilgili kelimenin bağlantı türü verilmiştir. Sistem bu yeni hali eğitildikten sonra ilk test aşaması sonucunda bağlantı türleri belirlenmiş olan test kümesi sisteme verilmiş ve ID karşılık bağlantı türü bulunmuş olmaktadır. Burada yapılması gerekenişlem bir önceki aşamada bağlanılan kelime (ID)'ye karşılık bağlılık türünün atanması işlemi terse çevrilmektdir. İkinci aşamada üretilen bağlantı türü aynı cümle içerisinde önce sağa sonra sola doğru aranır. Bulunduğu takdirde ilgili satırın ID'si sistem tarafından atanmaktadır. Bulunmaması halinde ise ilgili cümledeki ROOT etiketine bağlı ID ataması gerçekleştirilir.

Yapılan çalışmalar sonucunda sekans etiketleme problemlerinin çözümünde sıklıkla kullanılan CRF'nin ardışık sekans etiketleme problemlerinin çözümünde özellikle Türkçe için daha önce yapılmış çalışmalardan ve özellikle Bağlılık Ayrıştırması alanında sıkça kullanılan Malt Parser'a göre daha iyi sonuçlar elde etmiştir.

Türkçe dışında kullanılan Hollanda, Portekiz, Danimarka ve İsveç dilleri için CRF'nin performansının daha geride görülmesinin nedeni CRF'nin performansından ziyade test aşamasından sonra yapılan etiketten ID'ye çevirme işleminde kullanılan sağa bağımlı yapının bu diller için uygun olmamasıdır. Bu problemi çözmek için öncelikle bu dillere ait verisetlerin histogramları çıkarılarak çevirme işleminde kullanılan algoritmanın güncellenmesi ile bu verisetlerine ait başarı oranlarının artırabileceği düşünülmektedir.

Sistemin dile bağımlı yapıdan kurtarılması ve Türkçe dışındaki diller için başarının artırılması için gereken çalışmaların yapılması gelecek hedeflerimizdir. Bağlanılan

kelimenin bağlantı türü yerine sınıfının ya da alt sınıfını gibi ayırt ediciliği daha yüksek olması muhtemel alanların koyulması sistemin performansını artırabilir.

Yapılan çalışmalar sonucunda Malt Parser'ın bağlanılan kelimeyi (ID) bulma performansı, CRF'nin de Bağlantı Türü (DEP)'nü bulma performansındaki doğruluk değerleri daha yüksektir. Gelecek planları içerisinde bu iki yöntemin hibrit kullanabileceği bir yöntem geliştirmek sayılabilir.

Ayrıca Yan cümleciklere bölünen bir veri setinde cümleyi öğelerine ayırma işlemi daha başarılı olmuştur. Bunu bir adım ileriye götürerek yan cümleciklere bölünen bir cümleyi söz öbekleri,isim tamlamaları vb. alanlara bölerek sistemin performansının değerlendirilmesi yine gelecek hedefleri arasındadır.

KAYNAKLAR

[1] Sha, F. ve Pereira, F.,(2003). "Shallow Parsing with Conditional Random Fields", North American Chapter of the Association for Computational Linguistics (HLT-NAACL), May 31 and June 1 2003, Edmonton.

[2] Collins, M., (1996). "A New Statistical Parser Based on Bigram Lexical Dependencies", Association for Computational Linguistics (ACL), 24-27 June 1996, Santa Cruz.

[3] Collins, M., (1997). "Three Generative, Lexicalised Models for Statistical Parsing", Association for Computational Linguistics (ACL), 7-12 July 1997, Madrid.

[4] Jurafsky, D. ve Martin, J.H., (2000). Speech and Language Processing, Second Edition, Pearson Education International, New Jersey.

[5] Eisner, J., (1996). "Three New Probabilistic Models for Dependency Parsing: An Exploration", Computational Linguistics (COLING), 5-9 August 1996, Copenhagen.

[6] Haruno, M., Shirai, S. ve Ooyama, Y., 1998. "Using Decision Trees to Construct A Practical Parser", Machine Learning, 34:131-149.

[7] McDonald, R., Pereira, F., Ribarov, K. ve Hajic, J., (2005). "Non-Projective Dependency Parsing Using Spanning Tree Algorithms", Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP), 6-8 October 2005 , Vancouver.

[8] McDonald, R., Crammer, K. ve Pereira, F., (2005). "Online Large-Margin Training of Dependency Parsers", Association for Computational Linguistics (ACL), 25-30 June 2005 , Ann Arbor.

[9] Oflazer, K., (2003). "Dependency Parsing with An Extended Finite-State Approach", Journal Computational Linguistics, 29(4):515-544.

[10] Nivre, J., (2003). "An Efficient Algorithm for Projective Dependency Parsing", International Workshop on Parsing Technologies (IWPT), 23-25 April 2003, Nancy.

[11] Nivre, J. ve Nilsson, J., (2003). "Three Algorithms for Deterministic Dependency Parsing", Nordic Conference of Computational Linguistics (NODALIDA), Reykjavik, 30-31 May 2003, Reykjavik.

[12] Nivre, J., Hall, J. ve Nilsson, J., (2004). "Memory-Based Dependency Parsing", Computational Natural Language Learning (CoNLL), 6-7 May 2004, Boston.

[13] Buchholz, S. ve Marsi, E., (2006). "CoNLL-X Shared Task on Multilingual Dependency Parsing", Computational Natural Language Learning (CoNLL), 8-9 June 2006, New York.

[14] Chen, W., Zhang, Y. ve Isahara, H., (2007). "A Two-Stage Parser for Multilingual Dependency Parsing", Computational Natural Language Learning (CoNLL), 28-30 June 2007, Prague.

[15] Ambati, B.R., Samar,H., Sambhav,J., Sharma ,D.M. ve Sangal, R. , (2010). "Two Methods to Incorporate Local Morphosyntactic Features in Hindi Dependency Parsing", Statistical Parsing of Morphologically Rich Languages (SPMRL), 5 June 2010 , Los Angeles.

[16] Cer, D., Marneffe, M.C., Jurafsky, D. ve Manning, C.D., (2010). "Parsing to Stanford Dependencies: Trade-offs Between Speed and Accuracy", Language Resources and Evaluation (LREC), 19-21 May 2010, Malta.

[17] Eryiğit,G., İlbay,T. ve Can, O.A., (2011). "Multiword Expressions in Statistical Dependency Parsing", Statistical Parsing of Morphologically Rich Languages (SPMRL), 6 October 2011, Dublin.

[18] Orhan, Z., (2006). Türkçe Metinlerdeki Anlam Belirsizliği olan Sözcüklerin Bilgisayar Algoritmaları ile Anlam Belirginleştirmesi, Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[19] Covington M. A., (1994). Natural Language Processing for Prolog Programmers, Prentice Hall, New Jersey.

[20] Ergenç İ., (2002). Konuşma Dili ve Türkçe'nin Söyleyişi, Multilingual, İstanbul.

[21] Karadeniz,İ., (2007).Türkçe için Biçimbirimsel Belirsizlik Giderici, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[22] Weizenbaum, J., (1966), "ELIZA-A Computer Program for The Study of Natural Language Communication Between Man And Machine", Association for Computing Machinery (ACM), 9(1):36-45.

[23] Weaver, W., (1949). Translation, July 15 1949,New Mexico; Derleme: William N. ve Booth, A. D., (1955). Machine Translation Of Languages, John Wiley & Sons, New York.

[24] Chomsky, N., (1957). Syntactic Structures, Second Edition, Mouton, Berlin.

[25] Berger, A.L., Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Lafferty, J.D., Printz,H., Ures, L. ve Mercer, R.L., (1994). "The Candide System For Machine Translation", Human Language Technology (HLT), March 8-11 1994, New Jerey.

[26] Türk Dil Kurumu , Derlem, http://www.tdk.org.tr/index.php?option=com_gts&arama=gts&guid=TDK.GTS.563f237c195683.18771937, 11 Mart 2014.

[27] Say, B., (2006). "Türkçe İçin Bir Derlem Geliştirme Çalışması", Türk Dil Kurumu, 868:81-88.

[28] Tahiroğlu, B.T.,(2010). Bilgisayar Destekli Sözlük Bilimi Çalışmalarında Derleme Sözlüğü Veri Tabanı Örneği, Doktora Tezi, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü, Adana.

[29] McEnery,T., Xiao, R. ve Tono, Y., (2006). Corpus-Based Language Studies: An Advanced Resource Book, Routledge, Oxon.

[30] Ooi, V.B.Y., (1998). Computer Corpus Lexicography, Edinburgh University Press.

[31] Kennedy, G., (1998). An Introduction to Corpus Linguistics, Pearson Education Limited, New York.

[32] Sinclair, J., (2003). A Practical Guide to Lexicography- Corpora for Lexicography, John Benjamins Publishing Company, Amsterdam.

[33] Kruyt, T., (2003). A Practical Guide to Lexicography-Multifunctional Linguistic Databases: Their Multiple Use, John Benjamins Publishing Company, Amsterdam.

[34] Yöndem, M.T., (2006). "Bilişimsel Dil Bilimin Altyapı Ögeleri", Bilgisayar Destekli Dil Bilimi Çalıştayı Bildirileri, 9-10 June 2006, İstanbul.

[35] Çebi, Y., Varlıklar, Ö., (2006). "Türkçe Derlem Oluşturmada Karşılaşılan Sorunlar ve Çözüm Önerileri", Bilgisayar Destekli Dil Bilimi Çalıştayı Bildirileri, 9-10 June 2006, İstanbul.

[36] Erdogan, H.,(2010). "A Tutorial on Sequence Labeling Generative and Discriminative Approaches", International Conference on Machine Learning and Applications (ICMLA), 12-14 December 2010, Washington.

[37] Öztürk, A., (2004). Yöneylem Araştırması, Genişletilmiş 9. Baskı, Ekin Kitabevi Yayınları, Bursa.

[38] Manning, C.D. ve Schutze,H., (1999). Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge.

[39] Agun, H.V., (2008). Doğal Dil İşlemede Çizgesel ve Olasılık Tabanlı Bir Otomatik Öğrenme Uygulaması, Yüksek Lisans Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne.

[40] Cornell University Department of Computer Science, Sequence Tagging with HMMs-MEMMs, http://www.cs.cornell.edu/courses/cs4740/2011sp/lectures/memms-4pp.pdf, 11 Ekim 2015.

[41] Tür, G., Hakkani-Tür, D. ve Oflazer,K., (2003). "A Statistical Information Extraction System for Turkish", Natural Language Engineering, 9(2): 181-210.

[42] Lafferty, J., McCallum, A. ve Pereira, F., (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", International Conference on Machine Learning (ICML), 28 June-1 July 2001, Massachusetts.

[43] Wallach, H.M., Conditional random fields: An introduction, http://www.inference.phy.cam.ac.uk/hmw26/papers/crf_intro.pdf, 15 Ekim 2014.

[44] Kazkılınç,S., (2012). Türkçe Metinlerin Etiketlenmesi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[45] Rau, L.F., (1991). "Extracting Companu Names from Text", Artificial Intelligence Applications of IEEE, 24-28 February 1991, Miami.

[46] MacQueen, J. B., (1965). "Some Methods for Classification and Analysis of Mutivariate Observations", Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1:281-297

[47] Ekbal, A., Bandyopadhyay, S. ve Haque, R., (2009). "A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi", Linguistic Issues in Language Technology (LiLT), 2(1):1-44.

[48] Tesnière, L., (1959). Introduction A la Syntaxe Structurale, Klincksieck, Paris.

[49] Eryiğit, G., (2006). Türkçenin Bağlılık Ayrıştırması, Doktora Tezi,İstanbul Teknik Üniversitesi, İstanbul.

[50] Yıldırım, E., Bağlılık Ayrıştırması, http://www.nlpturk.org/?p=66, 12 Kasım 2014.

[51] Eryiğit, G., Adalı, E. ve Oflazer, K., (2006). "Türkçe Cümlelerin Kural Tabanlı Bağlılık Analizi, Turkish Symposium on Artificial Intelligence and Neural Networks, 21-24 June 2006, Muğla.

[52] Kudo, T. ve Matsumoto, Y., (2000). "Japanese Dependency Analysis Based on Support Vector Machines", Empirical Methods In Natural Language Processing and Very Large Corpora (EMNLP), 7-8 October 2000, Hong Kong.

[53] Yamada, H. ve Matsumoto, Y., (2003). "Statistical Dependency Analysis with Support Vector Machines", International Workshop of Parsing Technologies (IWPT), 23-25 April 2003, Nancy.

[54] Oflazer, K., (2003). "Dependency Parsing with An Extended Finite-State Approach", Computational Linguistics, 29(4):515-544.

[55] Nivre, J., (2003). "An Efficient Algorithm for Projective Dependency Parsing", International Workshop on Parsing Technologies (IWPT), 23-25 April 2003, Nancy.

[56] Graham N., NLP Programming Tutorial-Dependency Parsing, http://www.phontron.com/slides/nlp-programming-en-11-depend.pdf, 25 Kasım 2013.

[57] Vapnik, V., (1995). The Nature of Statistical Learning Theory, Second Edition, Springer, New York.

[58] Lu, W., Wang, W., Leung, A., Lo, S., Yuen, R., Xu, Z. ve Fan., H., (2002). "Air Pollutant Parameter Forecasting Using Support Vector Machines", International Joint Conference on Neural Networks (IJCNN), 12-17 May 2002, Honolulu.

[59] Shen, J., Pei, Z. J. ve Lee, E. S., (2004). "Support Vector Regression in the Analysis of Soft-Pad Grinding of Wire-Sawn Silicon Wafers", Cybernetics and Information Technologies Systems and Applications (CITSA), 21-25 July 2004, Orlando.

[60] Burbidge, R. ve Buxton, B., An Introduction to Support Vector Machines for Data Mining Technical Report, http://www.cs.ucl.ac.uk /staff/r.burbidge/pubs/yor12-svm-intro.html, 18 Eylül 2014.

[61] Kim, H., Pang, S., Je, H., Kim, D. ve Bang, S. Y., (2003). "Constructing Support Vector Machine Ensemble", Pattern Recognition, 36:2757–2767.

[62] Goh, K. S., Chang, E. ve Cheng, K. T., (2001). "SVM Binary Classifier Ensembles for Image Classification", Information and Knowledge Management (CIKM), 5–10 November 2001, Atlanta.

[63] CRF Program, CRFSharp, https://crfsharp.codeplex.com, 20 Ekim 2015.

[64] The University of Waikato, WEKA, http://www.cs.waikato.ac.nz/ml/weka, 20 Mart 2014.

[65] Zafer, H.R., Türkçe Cümle Çözümleyici, http://hrzafer.com/turkce-cumle-cozumleyici-fatih-parser, 20 Şubat 2014.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Metin BİLGİN
Doğum Tarihi ve Yeri	:19.06.1982 - Gemlik
Yabancı Dili	:İngilizce
E-posta	:metin_bilgin@hotmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Müh.Tamamlama	Bilgisayar Mühendisliği	Yalova Üniversitesi	2015
Y. Lisans	Elekt.ve Bilg.Sist.Eğt.	Selçuk Üniversitesi	2008
Lisans	Bilgisayar Sist.Öğrt.	Selçuk Üniversitesi	2005
Lise	Fen Bilimleri	Gemlik Lisesi	1999

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2005-Devam Ediyor	Milli Eğitim Bakanlığı	Teknik Öğretmen

YAYINLARI

Bildiri

1. Bilgin, M., Amasyalı, M.F., (2015). “Sekans Etiketleme Uygulamaları için Makine Öğrenmesi Yöntemlerinin Karşılaştırılması”, IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU), 16-19 Mayıs 2015, Malatya.

2.Bilgin, M., Amasyalı, M.F., (2015). “Semantic Role Labeling With Relative Clauses”,3rd Global Conference on Computer Science, Software, Networks and Engineering (COMENG), 19-21 Kasım 2015, İstanbul.

ÖDÜLLERİ

1. Selçuk Üniversitesi Teknik Eğitim Fakültesi Bölüm 2. si