Journal of the Faculty of Engineering and Architecture of Gazi University 32:2 (2017) 385-392

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Türkçe için ardışık şartlı rastgele alanlarla bağlılık ayrıştırma

Metin Bilgin[1*], Mehmet Fatih Amasyalı[2]
[1]Bursa Teknik Üniversitesi, Mekatronik Mühendisliği, Bursa, 16330, Türkiye
[2]Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği, İstanbul, 34220, Türkiye

Ö N E Ç I K A N L A R

- Şartlı rastgele alanlar ile sekans etiketleme
- Türkçe için bağlılık ayrıştırma
- Makine öğrenmesinin doğal dil işlemede kullanımı

Makale Bilgileri
Geliş: 04.03.2016
Kabul: 15.08.2016

DOI:
10.17341/gazimmfd.300607

Anahtar Kelimeler:
Optimizasyon,
sekans etiketleme,
bağlılık ayrıştırması,
şartlı rastgele alanlar,
makine öğrenmesi,
doğal dil işleme

ÖZET

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. 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. 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. Gerçekleştirilen bu çalışma ile Türkçe için en yüksek bağlılık ayrıştırması sonuçlarına ulaşılmıştır.

Dependency parsing with stacked conditional random fields for Turkish

H I G H L I G H T S

- Sequence labeling with Conditional Random Fields
- Dependency Parsing for Turkish
- Use of Machine Learning in Natural Language Processing

Article Info
Received: 04.03.2016
Accepted: 15.08.2016

DOI:
10.17341/gazimmfd.300607

Keywords:
Optimization,
sequence labelling,
dependency parsing,
condition random fields,
machine learning,
natural language processing

ABSTRACT

In the most general form Sequence Labelling is the production of an output sequence in response to an input sequence. Many of natural language processing problems such as (entity name recognition, machine translation, morphological analysis, separation of the elements of sentence etc.) can be defined as a sequence labelling. Dependency parsing is to determine the relationship and the type of the relationship between words within a sentence and it is essential to perform semantic analysis of a sentence. When dependency parsing is defined as a sequence labelling problem, production of two outputs (relationship type, related words) is required. Our recommendation is to use the Conditional Random Fields (CRF) which is commonly used in sequence labelling problems. However CRF is a method that produces a single output. To overcome this difficulty we propose to divide Dependency Parsing which is a problem with two outputs into two parts. The overall solution is provided by combining the results of these parts. With the performed operation we reached the best dependency parsing results for Turkish language.

*Sorumlu Yazar/Corresponding Author: metin.bilgin@btu.edu.tr / Tel: +90 224 300 3518

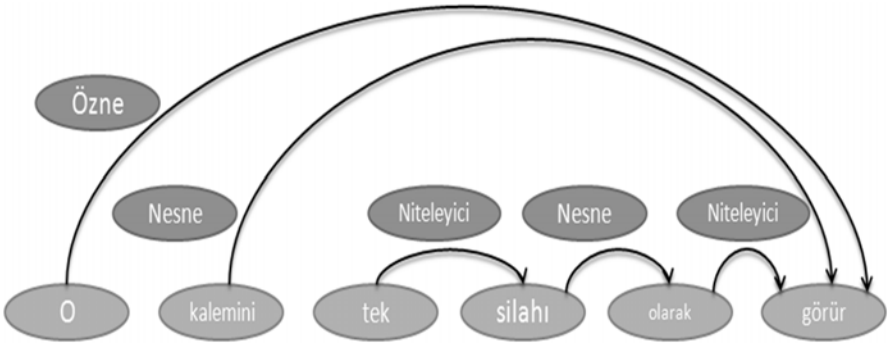
1. GİRİŞ (INTRODUCTION)

Sekans Etiketleme (Sequence Labeling - SL) , giriş sekansı I(i1,i2,…,in) şeklinde olan sekansa karşılık çıkış için $O(o_1,o_2,...,o_n)$ sekansın üretilmesidir. O kümesine göre birçok 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. 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. Amacımız, SL işleminde sıklıkla kullanılan ve Markov tabanlı modeller içinde en başarılısı olan Şartlı Rastgele Alanlar (Condition Random Fields-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. Bağlılık Ayrıştırma teorisinin Tesnière'in 1959'daki çalışmasına dayandığı söylenebilir. Tesnière'e göre "Cümle, kendisini oluşturan öğeleri sözcükler olan düzenli bir topluluktur" [1]. Günümüzde Doğal Dil Anlama (DDA) alanında kullanılan Bağlılık Ayrıştırma bu ilişki bağımlı(alt terim)-sahip(üst terim) ilişkisi olarak tanımlanmaktadır [2]. Şekil 1'de bağlılık grafiği görülmektedir. Bucholz ve Marsi (2006), CoNLL-2006 veri seti üzerinde gerçekleştirilen 13 farklı dil için gerçekleştirilen DP çalışmasının sonuçlarını yayınlamışlardır [4]. Chen ve ark. (2007), CoNLL-2007 veri seti üzerinde 10 farklı dil için DP çalışması gerçekleştirmişlerdir. Yapılan çalışmada DP programı olarak Malt Parser kullanılmıştır [5]. 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 [6]. Cer ve ark. (2010), DP problemi için Penn Treebank üzerinden elde edilen verilerle çalışmışlardır. 4 farklı ayrıştırma modeline göre çalışma sonuçlarını yayınlamışlardır [7]. Eryiğit ve ark. (2011), Türkçe için bir DP çalışması gerçekleştirmiştir. Yapılan çalışmada METU-SABANCI Türkçe Treebank üzerinden elde edilen 5635 cümlelik bir set kullanılmıştır. Çapraz doğrulama (Cross-Validation) yöntemi kullanılarak set 10 parçaya bölünmüş ve her seferinde 1 parçası test olarak kullanılmıştır. [8]. Gerçekleştirdiğimiz çalışmada bizde bu çalışmada kullanılan veri setini ve çapraz-doğrulama yöntemini kullandık. Bu çalışmada Türkçe cümleler için bağlılık ayrıştırma Markov tabanlı CRF algoritması kullanılarak iki aşamada gerçekleştirilmiştir. 2. Bölümde CRF, Malt Parser ve kullanılan veri seti ile ilgili bilgiler verilmiştir. 3. Bölümde Türkçe dilinin yapısı ilgili bilgiler verilmiştir. 4. Bölümde yapılan uygulama çalışması ile ilgili bilgiler verilmiştir. 5.Bölümde elde edilen sonuçlar verilmiş. Tartışma bölümünde ise elde edilen sonuçlar yorumlanmıştır.

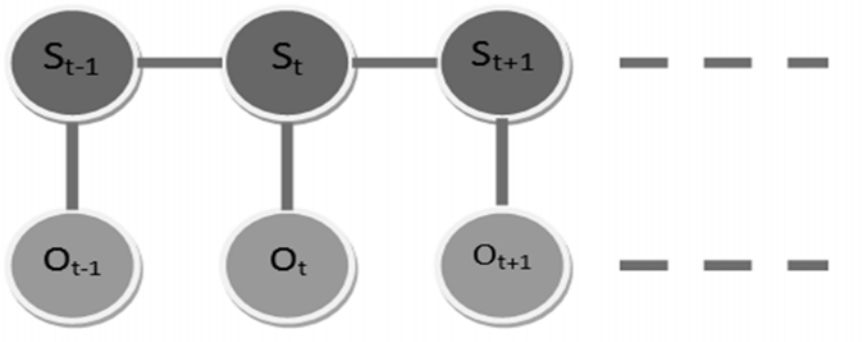
2. DENEYSEL METOT (EXPERIMENTAL METHOD)

2.1. Şartlı Rastgele Alanlar (Condition Random Fields)

CRF, Lafferty ve arkadaşları tarafından önerilen istatistiksel dizilim sınıflandırmasına dayanan bir makine öğrenmesi yöntemidir [9]. 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. CRF, NER, POS etiketleme, SP vb. gibi problemlerde sıklıkla başvurulan bir yöntemdir. CRF ile ilgili formül Eş. 1'de ve şekli Şekil 2'de verilmektedir.



Şekil 1. Bağlılık Grafiği (Dependency Graph) [3]



Şekil 2. Şartlı Rastgele Alanlar (Conditional Random Fields) [3]

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

$Z_{(\vec{o})}$ 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 faydalanılarak 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 [10]. Sonuç olarak en olası etiket dizilimine Y* dersek. Her bir sözcük dizilimi (o) için en yüksek olasılıklı etiket dizilimi Eş. 2'de verildiği gibi en yüksek olasılığı seçerek bulunabilir.

$$Y^* = \text{argmax}(P(s|o)) \quad (2)$$

2.2. Malt Ayrıştırıcı (Malt Parser)

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. Bu hareketler; Öteleme, Soldan Sağa Bağla ve Sağdan Sola Bağla şeklindeki durumlardır. Öteleme işleminde kuyrukta bekleyen elemanın yığına itilmesi demektir. Bu önceki kelimeyle bir bağlantı oluşturulamadığı ya da 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 [11]. 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 ayrıştırma 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 [12]. 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 [13]. SVM, yapısal risk minimizasyonu temelinde çalışarak bu problemlerin üstesinden gelmiştir. SVM, yüksek boyutlu fakat az sayıda veri içeren uygulamalarda da başarılıdır [14]. Bu özelliklerinden dolayı SVM; veri madenciliği [15], müşterilerin dolandırıcılık tespiti [16] ve görüntü sınıflandırma [17] gibi birçok uygulama alanında kullanılmıştır. DNA diziliminin ekson ve intron sınıflandırmasında ayrık fourier yöntemi ile elde edilen sonuçlar SVM ile karşılaştırılmıştır [18]. Konuşmacı doğrulama sistemleri için SVM kullanarak yaş ve/veya cinsiyete göre sınıflandırma problemlerine uyarlanmıştır [19]. Meme kanserinin teşhisinde SVM ve centroid tabanlı sınıflandırıcıların performansı ölçülmüştür [20].

2.3. Veri Seti (Data Set)

Bitişken bir dil olan Türkçe'de, sözcüklerin sonlarına ard arda çekim ve türetim ekleri konularak yüzlerce farklı sözcük oluşturulabilir. Tümcelerin, sözcük dizilişleri itibari ile büyük çoğunlukla SOP (Özne-Nesne-Yüklem) veya OSP kalıbına uymasına rağmen, öğeler anlatılmak istenen içeriğe ve vurguya bağlı olarak tümce içerisinde serbestçe yer değiştirebilirler. Türkçe'nin biçimbirimi oldukça karmaşık bir yapıya sahiptir: bir sözcük içerisinde, birden çok türeme görülebilir ve bir isim veya eylem kökünden üretilebilecek

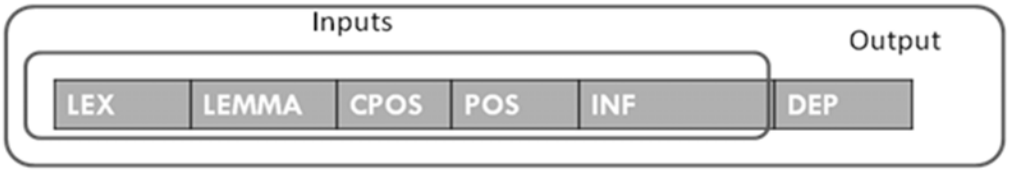
farklı sözcüklerin sayısı kuramsal olarak sonsuzdur. Türkçe'nin türetim sistemi çok üretkendir ve bir sözcüğün uydu veya iye olarak içerisinde bulunduğu tümce ilişkileri, sözcüğün içerdiği bir veya daha fazla türemiş yapının biçimbirimsel özellikleriyle belirlenmektedir [21]. Bağlılık Ayrıştırmasında özellik vektörlerinin oluşturulmasında kullanılan CoNLL formatındaki kalıp Şekil 3'de gösterilmektedir. Gerçekleştirilen çalışmada METU-SABANCI Türkçe Treebank üzerinden elde edilen 5635 cümlelik bir veri seti kullanılmıştır. Elimizdeki veri Şekil 3'de gösterilen formatta verileri içinde barındırmaktadır. Bu set içerisinde 65184 token bulunmaktadır [5].

3. SONUÇLAR VE TARTIŞMALAR
(RESULTS AND DISCUSSIONS)

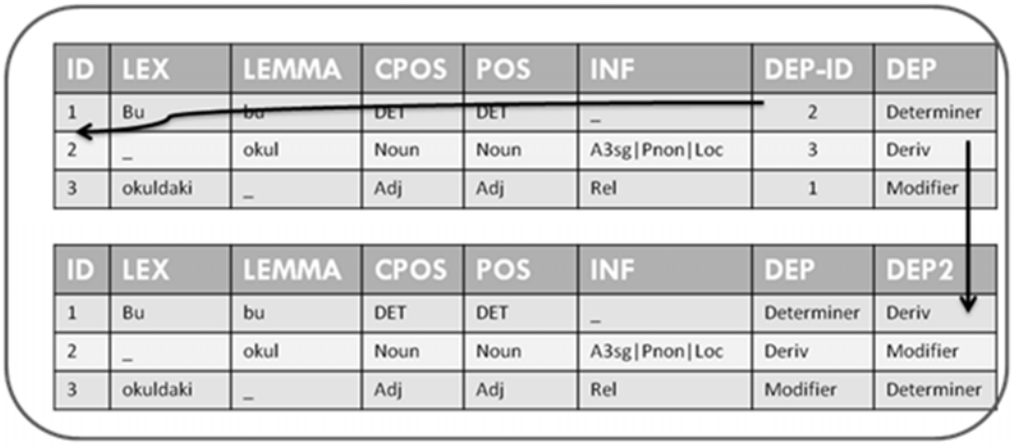
Gerçekleştirilen çalışmada AS_U, AS_L ve DEP metrikleri ölçülmüştür. AS_U metriği sözcüklerin çekim eklerine

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 3. CoNNL Veri Biçimi (CoNLL Data Pattern)



Şekil 4. 1.Aşama-DEP için girişler ve çıkış (First Stage- Inputs and Output for DEP)



Şekil 5. 1.aşama-DEP-ID yerine DEP atanması (First Stage- DEP-ID instead of DEP appointment)

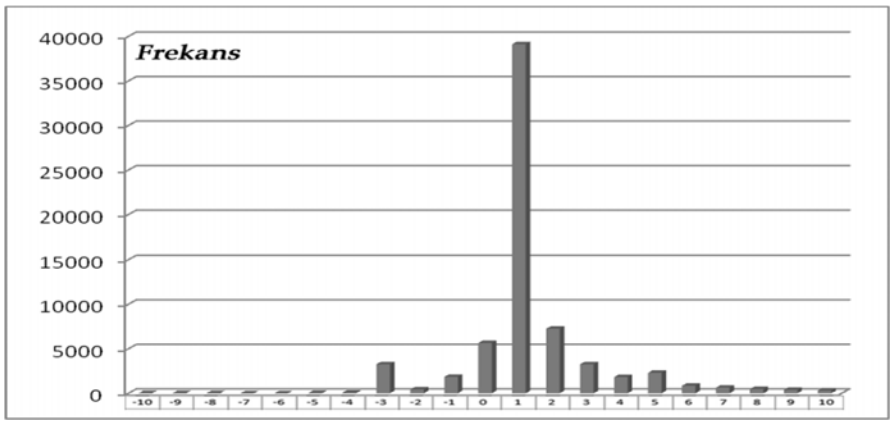
ayrılarak ve bağlantı türünün olmadığı yani doğru ID'ye (doğru çekim eki grubuna) bağlanma oranıdır. ASL metriği ise sözcüklerin çekim eklerine ayrılarak ve bağlantı türünün sisteme verildiği yani doğru ID'ye doğru etiketle (bağlılık türü) bağlanma oranını gösteren metriktir. DEP metriği doğru etiketi ile işaretlemeyi gösteren metriktir. AS_U, AS_L ve DEP %'lik olarak ifade edilmektedir. Bu çalışmada, özellikle bağlılık ayrıştırması uygulamalarında karşılaşılan Etiket ve ID bulma problemi için iki aşamalı bir sistem tasarlanıp, gerçeklenmeye çalışıldı. 5635 cümlelik veri setimiz 10 parçaya bölünmüş ve çapraz doğrulama (cross-validation) yöntemi ile her aşamada bir parça test seti olacak şekilde sistem eğitilip ardından test edilmiştir. Elde edilen sonuçların aritmetik ortalaması alınarak sonuçlar kayıt edilmiştir. Amacımız CRF'yi kullanarak Şekil 3'deki veri yapısında görülen 7. sütundaki DEP-ID ve 8. sütundaki DEP(Etiket) değerlerini 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 4'te görülmektedir. Eğitim aşamasının ardından test setimiz için DEP etiketlerimi üretilmiş ve 1.aşama sonuçlandırılmıştır. 2.Aşama da ise işimiz biraz daha karmaşıklaşmaktadır. Şekil 3'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' de görülmektedir. 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

Şekil 6. 2.Aşama-DEP2 için girişler ve çıkış (Second Stage- Inputs and Output for DEP2)

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 7. 2.Aşama-DEP2 değerinin ID'ye çevrilmesi (Second Stage- Translation from DEP2 to ID)



Şekil 8. Türkçe için Histogram Grafiği (Histogram Graph for Turkish)

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 6'da görülmektedir. 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 7'de bu durum görülmektedir. Dönüşüm işlemi yapılırken elimizdeki Türkçe veri setinin histogramı çıkarılmıştır. Histogramda 0 Root'a bağlanma oranını, 1 sağındaki 1. kelimeye, -1 solundaki 1. kelimeye bağlandığını göstermektedir. 2,-2,3,-3 vb. durumlarda aynı şekilde devam etmektedir. Şekil 8' deki

Tablo 1. Türkçe için sonuçlar (Results for Turkish)

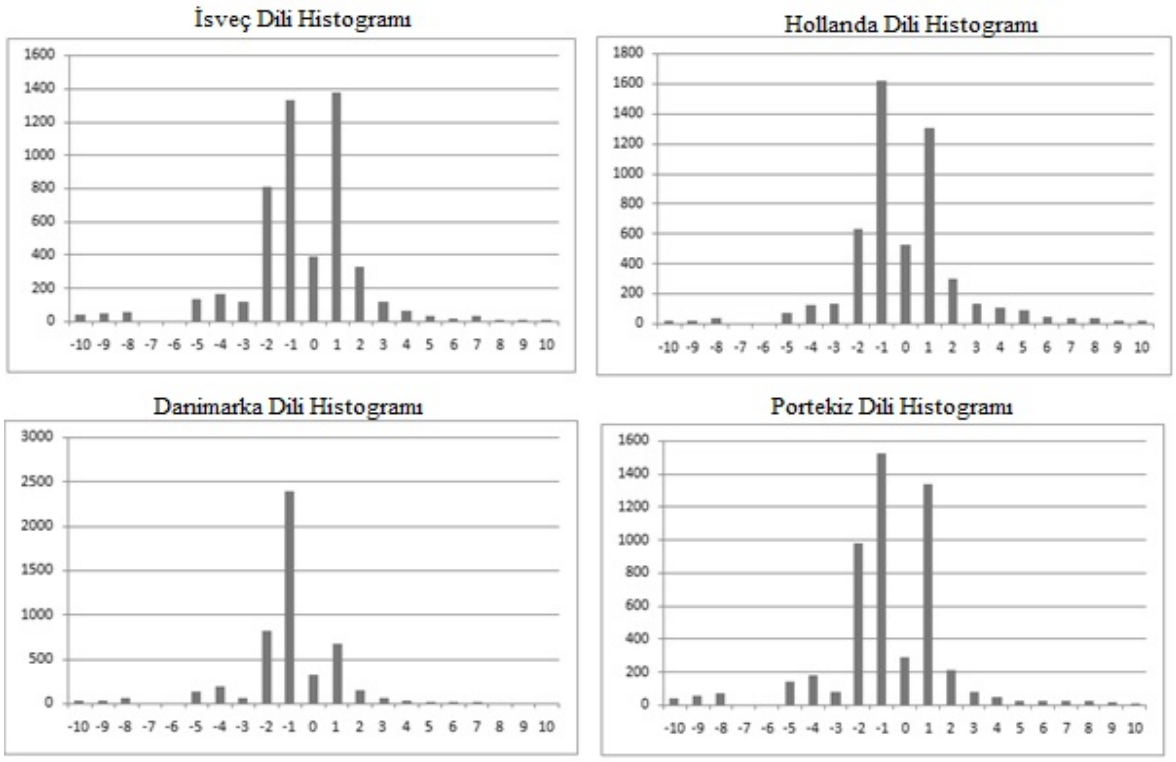
	AS_U	AS_L	DEP
Eryiğit ve ark. [8]	76,1	67,4	-
Malt Parser	77,16	67,23	75,55
Önerilen Yöntem - CRF	77,5	70,64	85,06

Tablo 2. Sık kullanılan bağlılık türleri için karışım matrisi (Confusion matrix for the frequent dependency types)

	Modifier	Deriv	Object	Sentence	Root	Subject	Notconnected	Coordination	Classifier	Determiner	Possessor	Dative.Adjunct
Modifier	10047	0	383	216	0	180	2	35	180	105	7	65
Deriv	0	11192	0	0	0	0	0	0	0	0	0	0
Object	375	0	6161	162	0	683	34	21	172	9	94	438
Sentence	192	0	96	6722	0	153	5	5	5	1	3	6
Root	2	0	1	0	5633	3	0	0	0	0	0	1
Subject	153	0	648	108	0	3098	1	2	136	34	222	39
Notconnected	0	0	50	0	0	0	2216	421	0	0	0	0
Coordination	28	0	10	4	0	4	577	1776	5	0	0	0
Classifier	187	0	161	14	0	137	2	1	1523	1	13	2
Determiner	61	0	4	35	0	15	0	0	0	1859	0	0
Possessor	19	0	58	1	0	172	0	0	22	0	1243	0
Dative.Adjunct	26	0	353	5	0	21	0	1	0	0	0	943

Tablo 3. Diğer diller için Sonuçlar (Results for Other Languages)

	Yöntem	DEP	AS_U	AS_L
İsveç Dili	Malt Parser	87,0	88,0	82,0
	CRF	84,0	54,0	51,0
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 9. Diğer dillere ait histogram grafikleri (Histogram graph for other languages)

Türkçe için hazırlanan histogramlar göstermiştir ki, veri setimizdeki 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. Bulunmaması halinde ise ilgili cümledeki ROOT etiketine bağlı ID ataması gerçekleştirilir. Bu şekilde oluşturulan sistemin Türkçe için sonuçları Tablo 1' de gösterilmiştir. Tablo 2'de bağlantı türü (DEP) için karışım matrisi görülmektedir. Karışım matrisinde en yüksek 12 frekans sahip etiket gösterilmiştir. Türkçe dışında Hollanda, Danimarka, Portekiz ve İsveç dilleri içinde sistem denenmiştir. Türkçe için gerçekleştirilen işlem basamakları bu diller içinde aynen uygulanmıştır. Tablo 3'de bu dillere ait bağlantı türü DEP, AS_U, AS_L başarıları verilmiştir. Ayrıca İsveç diline ait histogram, Hollanda diline ait histogram, Danimarka diline ait histogram ve Portekiz diline ait histogram Şekil 9'da görülmektedir.

4. SONUÇLAR (CONCLUSIONS)

İlk hipotezimiz , karmaşık bir problemi daha basit parçalara bölmenin problemin çözümünü kolaylaştıracağı ve bu durumun başarı oranı üzerinde olumlu etki yapacağı yönündedir. İkinci hipotezimiz ise 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. 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.

5. SİMGELER (SYMBOLS)

E	Eleman
$\propto$	Öğrenme Katsayısı
N	İsim (Noun)
V	Fiil (Verb)
ID	Tanımlama Kodu(Satır No)
DEP	Bağlantı Türü
AS_U	Doğru ID'ye Bağlanma Oranı
AS_L	Doğru ID'ye Doğru Etiketle bağlanma Oranı

TEŞEKKÜR (ACKNOWLEDGEMENT)

Veri setinin kullanımı konusunda desteklerinden ötürü Yrd. Doç. Dr. Gülşen Cebiroğlu Eryiğit'e ve çalışmaya olan desteklerinden ötürü Yrd. Doç. Dr. Murat Can Ganiz'e teşekkür ederiz.

KAYNAKLAR (REFERENCES)

1. Tesnière L., Introduction A la Syntaxe Structurale, Klincksieck, Paris, 1959.

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

3. Bilgin M., Ardışık Şartlı Rastgele Alanlarla Sekans Etiketleme, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2015.

4. Buchholz S., Marsi, E., CoNLL-X Shared Task on Multilingual Dependency Parsing, Computational Natural Language Learning (CoNLL), New York-Amerika Birleşik Devletleri, 149-164, 8-9 Haziran, 2006.

5. Chen W., Zhang Y., Isahara H., A Two-Stage Parser for Multilingual Dependency Parsing, Computational Natural Language Learning (CoNLL), Prague-Çek Cumhuriyeti, 1129-1133,28-30 Haziran, 2007.

6. Ambati B.R., Samar H., Sambhav J., Sharma D.M., Sangal R., Two Methods to Incorporate Local Morphosyntactic Features in Hindi Dependency Parsing, Statistical Parsing of Morphologically Rich Languages (SPMRL),Los Angeles-Amerika Birleşik Devletleri, 22-30, 5 Haziran, 2010.

7. Cer D., Marneffe M.C., Jurafsky D., Manning C.D., Parsing to Stanford Dependencies: Trade-offs Between Speed and Accuracy, Language Resources and Evaluation (LREC), Valletta-Malta, 1628-1632, 17-23 Mayıs, 2010.

8. Eryiğit G., İlbay T., Can O.A., Multiword Expressions in Statistical Dependency Parsing, Statistical Parsing of Morphologically Rich Languages (SPMRL), Dublin-İrlanda, 45-55, 6 Ekim, 2011.

9. Lafferty J., McCallum A., Pereira F., Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, International Conference on Machine Learning (ICML), Massachusetts-Amerika Birleşik Devletleri, 282-289, 28 Haziran - 1 Temmuz, 2001.

10. Kazkılınç S., Türkçe Metinlerin Etiketlenmesi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2012.

11. Eryiğit G., Adalı E., Oflazer K., Türkçe cümlelerin kural tabanlı bağlılık analizi, 15th Turkish Symposium on Artificial Intelligence and Neural Networks, Muğla-Türkiye, 17-24, 21-24 Haziran, 2006.

12. Vapnik V., The Nature of Statistical Learning Theory, Second Edition, Springer, New York, 1995.

13. Weizhen L., Wenjian W., Huiyuan F., Air Pollutant Parameter Forecasting Using Support Vector Machines, International Joint Conference on Neural Networks (IJCNN), Honolulu-Hawaii, 630-635, 12-17 Mayıs, 2002.

14. Shen J., Pei Z.J., Lee E.S., Support Vector Regression in the Analysis of Soft-Pad Grinding of Wire-Sawn Silicon Wafers, Cybernetics and Information Technologies Systems and Applications (CITSA), Orlando-Amerika Birleşik Devletleri, 19-24, 21-25 Temmuz, 2004.

15. Burbidge R., 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. Erişim Tarihi Eylül 18, 2014.

16. Kim H., Pang S., Je H., Kim D., Bang S.Y., Constructing Support Vector Machine Ensemble, Pattern Recognition, 36, 2757–2767, 2003.

17. Goh K.S., Chang E., Cheng K.T., SVM Binary Classifier Ensembles for Image Classification, Information and Knowledge Management (CIKM), Atlanta-Amerika Birleşik Devletleri, 395-402, 5–10 Kasım, 2001.

18. Daş B., Türkoğlu İ., Classification of DNA sequences using numerical mapping techniques and fourier transformation, Journal of the Faculty of Engineering and Architecture of Gazi University, 31 (4), 921-932, 2016.

19. Yücesoy E., Nabiyev V.V., Determination of a speaker's age and gender with an SVM classifier based on GMM supervectors, Journal of the Faculty of Engineering and Architecture of Gazi University, 31 (3), 501-509, 2016.

20. Takcı H., Diagnosis of breast cancer by the help of centroid based classifiers, Journal of the Faculty of Engineering and Architecture of Gazi University, 31 (2), 323-330, 2016.

21. Eryiğit G., Adalı E., Oflazer K., Türkçe'nin Olasılık Tabanlı Bağlılık Ayrıştırması, İstanbul Teknik Üniversitesitesi Mühendislik Dergisi, 7 (4), 106-117, 2008.