İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TÜRKÇE'NİN BAĞLILIK AYRIŞTIRMASI

DOKTORA TEZİ

Y. Müh. Gülşen ERYİĞİT

Anabilim Dalı : Bilgisayar Mühendisliği
Programı : Bilgisayar Mühendisliği

Tez Danışmanı : Prof. Dr. Eşref ADALI

EKİM 2006

ÖNSÖZ

Doktora çalışmalarımın başında yanımda olan, hayatımın en değerli kişilerinden bazıları şu an maalesef bu mutluluğu benimle paylaşamıyorlar. Öncelikle beni ben yapan babacığım Hasan Cebiroğlu'na her şey için çok teşekkür etmek istiyorum. Eminim ki bizi görüyor ve halen her adımımızla büyük gurur duyuyor. Hayat boyu hep yanımda olan ve gencecik yaşta kaybettiğim teyzeciğim İlker Görmez'e bana verdiği büyük destekten ötürü çok teşekkür ederim. Bu tezi onlara ithaf ediyorum.

Kendisini tanıdığım ilk günden beri, bir araştırmacı olmam için beni teşvik eden ve cesaretlendiren değerli hocam ve danışmanım Prof. Dr. Eşref Adalı'ya minnetlerimi sunarım. Doktora araştırmalarım sırasında engin bilgisini ve tecrübesini benden esirgemeyen değerli hocam Prof. Dr. Kemal Oflazer'e kendisi ile çalışmanın benim için büyük bir onur olduğunu belirtmek isterim. Ayrıca, doktora çalışmalarımın son senesinde ortak çalışma fırsatı bulduğum Prof. Dr. Joakim Nivre'ye ve araştırma grubuna, bana İsveç Växjö Ünivervisitesi'nde sağladıkları güzel ve sıcak çalışma ortamından ötürü teşekkürlerimi sunarım. Yine aynı nedenlerle, İTÜ Bilgisayar Mühendisliği Bölümündeki tüm çalışma arkadaşlarıma ve hocalarıma saygı ve sevgilerimi iletirim. Ayrıca bu araştırmaya yaptıkları destekten ötürü Tübitak Bilim Adamı Yetiştirme Grubu ve İstanbul Teknik Üniversitesi'ne teşekkürleri bir borç bilirim.

Son olarak, bugünlere gelmem için el birliği ile çalışan hayatımın en değerli kadınları annem ve halama, kardeşlerim Gülşah, Cihan ve Cansu'ya ve onlarla birlikte tüm aileme çok teşekkür ederim.

Ve sevgili eşim, hayat arkadaşım Cihat Eryiğit'e: "Sen olmasan yapamazdım".

Ekim 2006 Gülşen Cebiroğlu Eryiğit

İÇİNDEKİLER

KISALTMALAR

BT : Bellek Tabanlı
TB : Cümle Başarımı
CCG : Birleşenli Ulamsal Gramer
CFG : Bağlamdan Bağımsız Gramer
ÇK : Çekim Kümesi
ÇKB : Çekim Kümeleri arası Başarım
$\acute{C}KB_E$: Çekim Kümeleri arası Başarım (etiketli)
DDİ : Doğal Dil İşleme
GPSG : Kapsamlı Öbek Yapısal Gramer
HPSG : Baş-sürümlü Öbek Yapısal Gramer
LFG : Sözcüksel İşlevsel Gramer
LPCFG : Görünüm Bilgisi Eklenmiş PCFG
KDM : Karar Destek Makineleri
BD : Bilişimsel Dilbilim
PCFG : Olasılık Tabanlı Bağlamdan Bağımsız Gramer
TAG : Ağaç Birleştiren Gramer
TS : Türetim Sınırı
SB : Sözcükler arası Başarım
SDD : Sonlu Durumlu Dönüştürücü
Sİ : Sözcük İçi

TABLO LİSTESİ

ŞEKİL LİSTESİ

TÜRKÇE'NİN BAĞLILIK AYRIŞTIRMASI

ÖZET

Bağlılık ayrıştırması, bir tümce içindeki sözcükler arası ikili ilişkileri saptayarak o tümcenin çözümlemesini sağlayan yönteme verilen addır. Bir tümcenin anlamının çıkarılması doğal dil işlemenin ana hedefleri içindedir. Eğer tümcenin hedeflediği gerçek anlam çıkartılabilirse bu tümce makineler tarafından eyleme dönüştürülebilir ya da bu tümcenin başka dillerdeki karşılıkları bulunabilir. Anlam çıkarılması için öncelikle tümcenin çözümlenmesi gerekmektedir. Bir tümcenin çözümlemesi demek tümce içinde bulunan sözcüklerin görevlerinin belirlenmesi anlamındadır. Çözümleme yöntemlerinden biri de bağlılık ayrıştırmasıdır. Bu nedenle, bağlılık ayrıştırması doğal dil çalışmalarının temel konuları içinde sayılır.

Bir tümcenin çözümlemesi doğal olarak, tümcenin yapısına (tümce içerisindeki sözcüklerin dizilişine ve sözcüklerin yapısına) bağlıdır. Türkçe, bitişken ve tümce içi öğe dizilişleri serbest bir dildir. Bu nedenle tümce çözümlemesi Hint-Avrupa dillerine oranla daha karmaşıktır. Bu çalışmada geliştirilen yöntemin, Türkçe dil ailesi için kullanılabileceği gibi Türkçe'ye yakın dillere de uygulanabileceği düşünülmektedir. Örneğin Ural dil ailesinde yer alan Fince, Estonyaca, Macarca, Altay dil ailesinde yer alan Japonca ve Korece gibi.

Tümce çözümlemesi konusunda yapılmış çalışmalar incelendiğinde çalışmaların çoğunun Hint-Avrupa dil ailesi ve özellikle İngilizce üzerinde yapıldığı görülmektedir. Türkçe'nin de içinde yer aldığı Ural-Altay dil ailesi için yakın zamanda birçok araştırma başlatıldığı görülmektedir. Yukarıda değinildiği gibi, tümce çözümlemesi tümce yapısına bağlıdır. Hint-Avrupa dillerinin tümce yapısı ile Ural-Altay dillerinin tümce yapıları çok farklı olduğundan İngilizce için yapılmış olan çalışmaların Türkçe için kullanılabilmesi olanaklı değildir.

Bu tez çalışmasının hedefi, Türkçe tümcelerin çözümlemesini bağlılık ayrıştırması yöntemini kullanarak en yüksek başarımla gerçeklemektir. Bu amaçla:

- Türkçe'nin tümce yapısı bağlılık açısından incelenmiş,
- Türkçe'nin bağlılık yapısı modellenmiş,
- Farklı nitelikte ayrıştırıcılar geliştirilerek, ayrıştırıcıların ve modellerin başarımları karşılaştırılmış,
- Sonuç olarak ayırdedici öğrenmeye dayalı sınıflandırıcı tabanlı gerekirci ayrıştırıcının en iyi sonucu verdiği ortaya konmuştur.

Bu çerçevede, çalışmamızın bilime yaptığı katkılar aşağıda sıralanmıştır:

- Daha önce başka diller için gerçeklenmiş olan ayrıştırıcılar, bitişken olmayan dilleri ayrıştırmak üzere tasarlanmışlardır. Bu nedenle Türkçe için kullanıldıkları zaman başarımları düşük olmaktadır. Bu çalışmada geliştirilen ayrıştırma yöntemi, sözcüğün kökü, ekleri ve biçimbilimsel yapısını da dikkate alarak çalışmaktadır.
- Sözcüklerin ikili bağımlılıklarını araştırmada kullanılabilecek çok sayıda yöntemin varlığı bilinmektedir. Bu yöntemler bağlılıkları belirlerken sözcüklerin farklı özelliklerinden faydalanırlar. Bunlardan bazıları sözcüklerin metin içerisindeki *görünüm şekilleri*, *nitelikleri*, *biçimbilimsel özellikleri*, *komşu özellikleri*, *yakınlık durumları* gibi özelliklerdir. Bağlılık araştırması yaparken bu özelliklerin sayısının arttırılarak kullanılması halinde bulunan çözüm belirginliği artmakta ancak sonuca ulaşma olasılığı düşmektedir. Kullanılan özellik sayısı azaltıldığı durumda ise çözüm olasılığı yükselmekte ancak belirsizlik artmaktadır. Bu çalışmada Türkçe için bağlılık ayrıştırmasında hangi özelliklerin kullanılması halinde en iyi çözümün bulunacağı gösterilmiştir.

Çalışmalarımız sırasında, yakın geçmişte yayınlanan Türkçe ağaç yapılı derlem kullanılarak, veri güdümlü ayrıştırıcılarda farklı tasarım yöntemlerinin kullanılmasının etkileri incelenmiştir. Bu incelemeler sırasında, temel model olarak alınan bazı kural tabanlı ayrıştırıcılar, olasılık tabanlı modele dayalı bir istatistiksel ayrıştırıcı ve ayırdedici öğrenmeye dayalı sınıflandırıcı tabanlı gerekirci bir ayrıştırıcı olmak üzere farklı ayrıştırma yöntemlerine sahip ayrıştırıcılar kullanılmış ve tasarım yöntemlerinin etkileri bunlar üzerinde değerlendirilmiştir. Daha sonra, ayrıştırmada çekim kümesi adı verilen biçimbilimsel birimleri, biçimbilimsel özellikleri ve görünüm bilgisi kullanmanın etkileri incelenmiştir. Ayrıştırıcıların sonuçları üzerinde incelemeler yapılmış ve başarımları ilgili yayınlardaki ayrıştırıcıların başarımları ile karşılaştırılmışlardır.

Sonuçlar, sözcükler yerine sözcüklerden daha küçük olan çekim kümelerinin tümce yapısının ana birimleri olarak kullanılmasıyla, Türkçe'de ayrıştırma başarımının arttırılabileceğini göstermektedir. Ayrıca biçimbilimsel özelliklerin ve görünüm bilgisi eklemenin, Türkçe'nin bağlılık çözümlemesinde çok önemli etkisi olduğu görülmüştür. Ancak, bu bilgileri tümüyle kullanmanın bazı ayrıştırıcıların başarımlarını kötü yönde etkilediği gösterilmiştir. Seçilen ayrıştırıcının niteliklerine bağlı olarak görünüm bilgisinin veya çekimsel özelliklerin kısmi olarak kullanılması önerilmiştir.

Bu tez çalışmasının sürdürüldüğü sırada benzer çalışmaların yapıldığı gözlemlenmiştir. Bu tezde geliştirilen yöntem ve aynı konuda yapılan diğer çalışmalar Haziran 2006 tarihinde CoNLL-X ortak çalışmasında aynı veri kümesi üzerinde sınanmıştır. Geliştirilen ayrıştırıcının diğer ayrıştırıcılara oranla en yüksek başarıyı verdiği gösterilmiştir.

DEPENDENCY PARSING OF TURKISH

SUMMARY

Dependency parsing is a syntax analysis method which aims to make the analysis of a sentence by determining the binary relationships between the words within that sentence. Understanding the meaning of a sentence is one of the main goals of natural language processing. If the real meaning of a sentence can be determined, this sentence can be translated into action by machines or its translation into other languages can be found out. The sentence should be analyzed first in order to resolve its meaning. The analysis of a sentence means to determine the roles of the words composing that sentence. One of the analysis methods is the dependency parsing. Therefore, it is considered within the main topics of natural language processing.

As might have been expected, the analysis of a sentence is related to its structure (the order of the words within the sentence and the structure of the words). Turkish is a language that is characterized by a rich agglutinating morphology, free constituent order, and predominantly head-final syntactic constructions. Therefore, its syntax analysis is more complex when compared to Indo-European languages. The method developed in this study can be used for the languages in Turkish language family and it is also thought to be suitable for the languages which are similar to Turkish such as Finnish, Estonian, and Hungarian in Ural language family and Japanese and Korean in Altaic language family.

When the studies on syntax analysis are investigated, it is seen that most of these are conducted for Indo-European languages and mostly for English. It is observed that there are many studies started recently for Ural-Altaic languages including Turkish. As stated above, syntax analysis depends on the sentence structure. Since the sentence structures of Indo-European languages and Ural-Altaic languages are very different from each other, it is not possible to apply the approaches developed for English to Turkish.

The aim of this thesis is to perform the syntax analysis of Turkish sentences with highest accuracy by using dependency parsing. With this aim:

- The syntax structure of Turkish sentences is investigated based on dependency relations,
- The dependency structure of Turkish is modeled,
- Parsers from different methodologies are developed and the performances of the parsers and parsing models are compared,
- It is shown that the best results are obtained with the classifier-based parser based on discriminative learning.

With this perspective, the contributions of this thesis are listed below:

- The parsers developed recently are designed to parse languages which are not agglutinative. Therefore, they are seen to perform lower when they are applied for Turkish. The parsing method developed in this study works by considering the stem, the suffixes and the morphological structure of the words.
- The methods to determine the binary dependencies between words get use of different features of the words such as their lexical information, part-of-speech category, inflectional features, neighbors' features and the distance between them. During the dependency analysis, the over-usage of these features causes sparse data problem. In this study, the necessary combination of the features to obtain the best dependency parsing accuracy is shown.

During our study, the impact of different design choices in developing data-driven parsers is investigated using data from the recently released Turkish Treebank (Metu-Sabancı Turkish Treebank). We first investigated the basic parsing methodology, including both parsing algorithms and learning algorithms by using some rule-based parsers as baselines and a statistical parser using a conditional probabilistic model and a deterministic classifier-based parser using discriminative learning. We then examined the impact of using morphological units, inflectional features and lexicalization in parsing. We made detailed analysis on the results and the success is compared with other works in literature.

The results showed that parsing accuracy in Turkish can be improved by taking morphologically defined units rather than word forms as the basic units of syntactic structure. In addition to this, it is seen that using inflectional features and lexicalization is crucial for the dependency parsing of Turkish. However, there are some evidence that the entire usage of these informations may be harmful for the parsing accuracy. Depending on the parser's characteristics, it is suggested to use partial lexicalization and inflectional features.

The method developed in this thesis and other methods in the literature about the same topic are tested on the same dataset in the Conll-X shared task (June 2006). It is shown that the most state-of-the-art results in the literature for dependency parsing of Turkish are obtained by using the parser introduced in this thesis.

1. GİRİŞ

Bağlılık ayrıştırması, bir tümce içindeki sözcükler arası ikili ilişkileri saptayarak o tümcenin çözümlemesini sağlayan yönteme verilen addır. Bir tümcenin anlamının çıkarılması doğal dil işlemenin ana hedefleri içindedir. Eğer tümcenin hedeflediği gerçek anlam çıkartılabilirse bu tümce makineler tarafından eyleme dönüştürülebilir ya da bu tümcenin başka dillerdeki karşılıkları bulunabilir. Anlam çıkarılması için öncelikle tümcenin çözümlenmesi gerekmektedir. Bir tümcenin çözümlemesi demek tümce içinde bulunan sözcüklerin görevlerinin belirlenmesi anlamındadır. Çözümleme yöntemlerinden biri de bağlılık ayrıştırmasıdır. Bu nedenle, bağlılık ayrıştırması doğal dil çalışmalarının temel konuları içinde sayılır.

Bir tümcenin çözümlemesi doğal olarak, tümcenin yapısına (tümce içerisindeki sözcüklerin dizilişine ve sözcüklerin yapısına) bağlıdır. Türkçe, bitişken ve tümce içi öğe dizilişleri serbest bir dildir. Bu nedenle tümce çözümlemesi Hint-Avrupa dillerine oranla daha karmaşıktır. Bu çalışmada geliştirilen yöntem Türkçe dil ailesi için kullanılabileceği gibi Türkçe'ye yakın dillere de uygulanabileceği düşünülmektedir. Örneğin Ural dil ailesinde yer alan Fince, Estonyaca, Macarca, Altay dil ailesinde yer alan Japonca ve Korece gibi.

Tümce çözümlemesi konusunda yapılmış çalışmalar incelendiğinde çalışmaların çoğunun Hint-Avrupa dil ailesi ve özellikle İngilizce üzerinde yapıldığı görülmektedir. Türkçe'nin de içinde yer aldığı Ural-Altay dil ailesi için yakın zamanda birçok araştırma başlatıldığı görülmektedir. Yukarıda değinildiği gibi, tümce çözümlemesi tümce yapısına bağlıdır. Hint-Avrupa dillerinin tümce yapısı ile Ural-Altay dillerinin tümce yapıları çok farklı olduğundan İngilizce için yapılmış olan çalışmaların Türkçe için kullanılabilmesi olanaklı değildir.

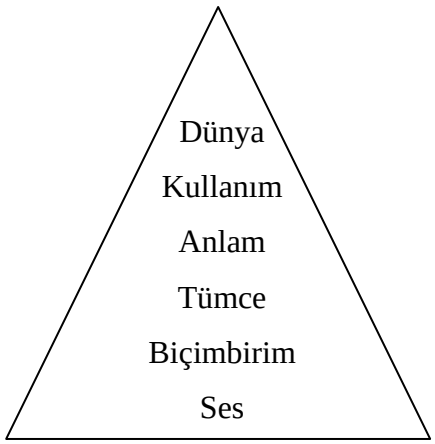
Aşağıdaki bölümlerde ilk olarak bağlılık çözümlemesi ve daha sonra bu ve benzer yaklaşımlar ile ilgili yayınlarda yapılan araştırmalar tanıtılacaktır. Tezin katkısı ve tezi oluşturan bölümlerin sunulmasıyla birlikte giriş bölümü sona erecektir.

1.1 Bağlılık Çözümlemesi

Yapay zekanın bir alt dalı olan *Doğal Dil İşleme*[×][1] (DDİ, *Bilişimsel Dilbilim*[×]) insanların kullandıkları dili işlemeye yönelik teknikler geliştirmeyi amaçlar. Bu teknikler kullanılarak, insan makine iletişimini arttırma, makine ile çeviri yapma, hızlı bilgi çıkarımı[×] gibi bir çok konuda uygulamalar geliştirilmektedir. DDİ alanında yapılan çalışmalarda kullanılan bilgi düzeyleri altı ana başlık altında toplanabilir. Bunlar, *sesbilimi*[×], sözcüklerin yapısını inceleyen *biçimbilim*[×], sözcüklerin yapısal ilişkilerini inceleyen *sentaks bilgisi*[×] (tümce çözümlemesi), *anlamsal bilgi*[×], bir amaca ulaşmada dilin nasıl kullanılacağını inceleyen *kullanım bilgisi*[×] ve bir işlemin yürütülebilmesi için gerekli ön bilgiyi inceleyen *dünya bilgisi*dir[×]. Bu bilgi düzeylerinin herbiri veriyi işlerken Şekil 1.1'de kendisinden önce gelen bilgi düzeylerinin sonuçlarından yararlanırlar. Tümce çözümlemesi, sözcüklerin öğe olarak görevlerinin belirlenmesi (özne, yüklem vb...), tümce içerisindeki dizilişlerinin incelenmesi, sözcükler arasındaki ilişkilerin bulunması gibi konularda yapılan çalışmalara verilen isimdir. Bir tümcenin gramer yapısını belirlemek üzere incelenmesi işlemine ayrıştırma[×] denir. Tümce çözümlemesi ve ayrıştırma, ilgili yayınlarda eşdeğer olarak kullanılan terimlerdir. Ayrıştırıcı ise bu işlemi gerçekleştiren bilgisayar yazılımıdır. Ayrıştırma, DDİ alanında geliştirilen uygulamaların bir çoğu (anlam çıkarma, özetleme, soru cevaplama, bilgisayarlı çeviri vb...) için önemli bir bileşendir.

Tümce çözümlemesi yöntemlerinden biri olan Bağlılık Çözümlemesi[×] yöntemi, çok eskiden beri bilinen bir yöntem olmasına karşın Doğal Dil Ayrıştırması[×] (DDA) alanında ancak son yıllarda yoğun olarak kullanılmaya başlanmıştır. Özellikle farklı özellikteki diller için bağlılık gramerleri kullanılarak oluşturulan derlemlerin (Arapça (Hajic ve diğ., 2004), Çekçe (Hajič ve diğ., 2001), Danca (Kromann, 2003), İtalyanca (Bosco, 2004), Japonca (Lepage ve diğ., 1998), Slovakça (Dzeroski ve

[1] Arkasından "×" işareti gelen terimlerin İngilizce karşılıkları Ek E'de verilmektedir.

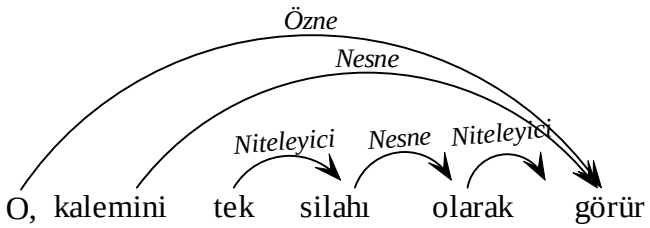


Şekil 1.1: DDİ Bilgi Düzeyleri

diğ., 2006), Türkçe (Oflazer ve diğ., 2003)) sayısının artması ile birlikte, bu konuda yapılan araştırmalar da hız kazanmıştır. Bunlara ek olarak birçok çalışmada, diğer gramer yöntemleri kullanılarak işaretlenmiş derlemler (Almanca (Brants ve diğ., 2002), Bulgarca (Simov ve diğ., 2002), Çince (Huang ve diğ., 2000), Hollandaca (van der Beek ve diğ., 2002), İngilizce (Marcus ve diğ., 1993), İsveççe (Nilsson ve diğ., 2005), İspanyolca (Civit Torruella ve Martí Antonín, 2002), Japonca (Kawata ve Bartels, 2000), Portekizce (Afonso ve diğ., 2002)) bağlılık gramerine uygun bir yapıya dönüştürülmüş ve bağlılık çözümlemeleri yapılmıştır.

Güncel bağlılık grameri kuramının, Tesnière'in 1959'daki çalışmasına dayandığı söylenebilir. Tesnière'e göre "Tümce, kendisini oluşturan öğeleri sözcükler olan düzenli bir topluluktur" (Tesnière, 1959). "Zihin, tümceyi oluşturan sözcükler ve komşuları arasında ilişkileri bulur ve bu ilişkilerin bütünü tümcenin iskeletini oluşturur. Her bir ilişki bir alt terimi bir üst terime bağlamaktadır." Günümüzde DDA alanında kullanılan bağlılık gramerlerinde bu ilişki *uydu (alt terim) - iye (üst terim)*$^{\times}$. ilişkisi olarak tanımlanmaktadır. Bağlılık grameri tabanlı metin ayrıştırmasının amacı metin içerisinde geçen her tümce için tümceyi oluşturan sözcükler arasındaki uydu-iye ilişkilerini bulmaktır. Şekil 1.2'de Türkçe bir tümcenin bağlılık durumu gösterilmektedir. İlgili yayınlarda, bağlılık oklarının yönü ile ilgili iki farklı yaklaşım benimsenmektedir. Bunlar bağlılık okunu; 1° uydu birimden çıkarak iye birime doğru çizmek, 2° iye birimden çıkarak uydu birime doğru çizmektir. Bu tez çalışmasında,

Şekil 1.2'de görüldüğü gibi, birinci yaklaşım benimsenmiştir; sözcükler arasında çizilen oklar uydu sözcükten iye sözcüğe doğru olan bağlılığı belirtmektedirler.



Şekil 1.2: Bağlılık Grafiği

Okların üzerlerine yazılan etiketler ise iki sözcük arasındaki bağlılığın türünü belirtmektedir. Şekil 1.2'de birden fazla iye vardır ancak ana iye "görmek" eylemidir. Bu eyleme bir özne ("O"), bir nesne ("kalemini") ve bir niteleyici ("olarak") olmak üzere üç adet uydu sözcük bağlanmıştır. Eylemin niteleyicisi bir isim kümesinden ("tek silahı olarak") oluşmaktadır. Bu küme içerisindeki sözcükler de birbirlerine uydu-iye okları ile bağlanmışlardır.

Bir bağlılık ayrıştırıcısının hedefi *etiketli* veya *etiketsiz* bağlılıkları bulmak olabilir. Ana iye dışındaki tüm sözcükleri bir başka sözcüğün uydusu olan grafikler *bağlı*$^{\times}$ grafik olarak anılırlar. Bu koşula uymayanlar ise *kopuk* olarak anılırlar. İye sözcüğün uydu sözcüğün sağ tarafında bulunduğu bağlılıklara *sağa bağımlı*$^{\times}$ bağlılıklar denir. Şekildeki tüm bağlılıklar bu türde bağlılıklardır. İye sözcüğün bağımlı sözcüğün sol tarafında bulunduğu bağlılıklara ise *sola bağımlı*$^{\times 2}$ bağlılıklar denir. Bağlılık grafiği çizildiğinde herhangi bir bağlılık ile kesişmeyen, bir diğer deyişle Uydu → İye bağlılık oku altında bulunan tüm sözcüklerin iye sözcüğe doğrudan veya dolaylı olarak bağlı olduğu bağlılıklara *kesişmeyen*$^{\times}$ bağlılıklar[3] denir. Bu koşulun dışında kalan bağlılıklara ise *kesişen*$^{\times}$ bağlılıklar denir. Ayrıştırıcı, bir tümce üzerinde işlem yaparken sözcüklerin sözcük sınıfları (isim, sıfat vb...), komşuluk bilgileri ve görünüm bilgileri (o, tek, silahı) gibi birçok özelliklerinden faydalanır. Çalışmalar, sözcüklerin *görünüm bilgileri*nin$^{\times}$ kullanılıp kullanılmamasına göre *görünüm bilgisi eklenmiş*$^{\times}$

[2]Türkçe'de bu tür bağlılıklar daha çok devrik tümcelerde görülürler.

[3]Bağlılıkların kesişmesi dışında, bağlılık oku altında hiçbir yere bağlanmamış sözcükler bulunması da bu türden bağlılıklara yol açabilir. Bu durumda bağlılık yapısı aynı zamanda kopuk olacaktır. Anlaşılırlık açısından bu bağlılıklar *kesişmeyen* olarak adlandırılmışlardır.

ve *görünüm bilgisi eklenmemiş* olmak üzere iki farklı şekilde adlandırılacaklardır. Yukarıda anlatılanların ışığında, bağlılık ayrıştırıcısının amacı bir tümce için bağlılık grafiği oluşturmaktır. Bu amaçla, olası bütün bağlılık grafikleri içerisinden en uygun olanı bulmaya çalışır.

Yukarıda tanıtılan farklı bağlılık türlerine çeşitli dil ailelerinde rastlanmaktadır. Örneğin, birçok dil için kesişmeyen bağlılıklardan oluştukları varsayımı yapılırken, Çek dilindeki kesişen bağlılıkların çokluğu, yayınlarda vurgulanan bir nokta olarak dikkat çekmektedir. Şekil 1.2'deki Türkçe tümcenin İngilizce, Fransızca, Macarca, Fince ve Japonca karşılıkları Şekil 1.3'de verilmektedir. Aynı tümcenin Türkçe'sine ve bu dillerdeki karşılıklarına baktığımızda, şu yorumlar yapılabilir:

1. Bağlılık yapılarında kesişmenin olmadığı,
2. Ana iyeye doğru yönlenmenin olduğu,
3. Türkçe ve Japonca'da bütün iyelerin uydudan sonra geldiği (sağa bağımlı), İngilizce, Fransızca, Macarca ve Fince'deki örneklere baktığımızda ise bağlılıkların yönünün tümce içerisinde birkaç kez değiştiği (karma bağımlı) görülmektedir.

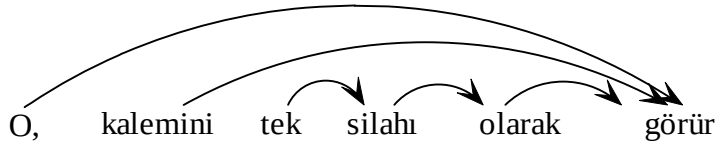
Türkçe'de devrik tümce yapıları dışarıda bırakıldığında tümce yapısının kesişmeyen ve sağa bağımlı olduğu görülmektedir. Türkçe'nin bu özelliği ileriki bölümlerde daha ayrıntılı olarak incelenecektir.

İlgili çalışmalarda, bir bağlılık yapısının düzgün olarak kabul edilmesi için aşağıda sıralanan üç kısıt getirilmiştir. Bu listedeki ilk iki kısıt bağlılık grafiğinin köklü bir ağaç yapısında olduğu varsayımına dayanmaktadır.

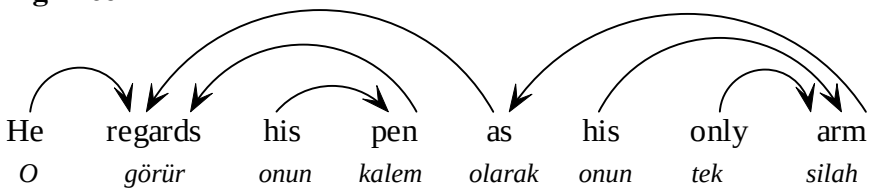
- Döngülere izin vermeyen
- Bağlı
- Kesişmeyen

DDA konusunda yapılan çalışmalar, kullandıkları gösterim yönteminden bağımsız olarak *gramer güdümlü*$^{\times}$ ve *veri güdümlü*$^{\times}$ olmak üzere iki kümeye ayrılabilirler.

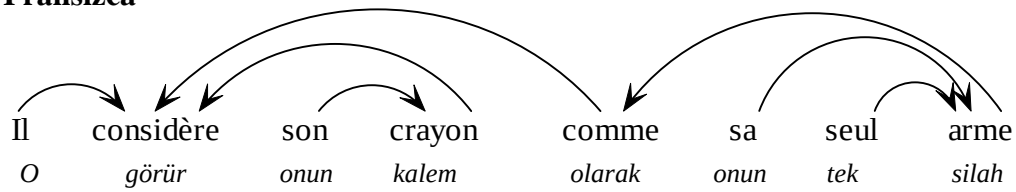
Türkçe



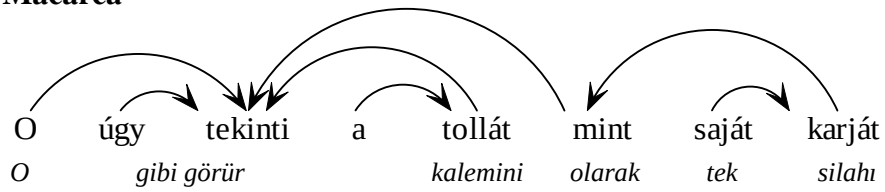
İngilizce



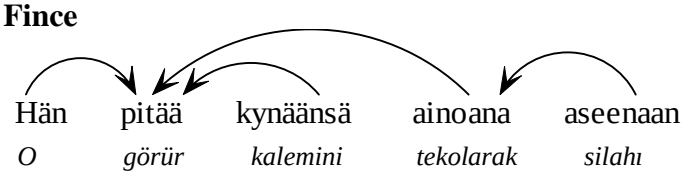
Fransızca



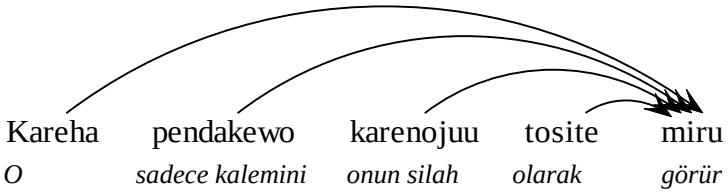
Macarca



Fince



Japonca



Şekil 1.3: Farklı Dillerde Bağlılık Yapıları

Gramer güdümlü yaklaşımlar, düzenli bir gramer kullanılarak bir dilin ayrıştırılmasını hedeflemektedirler. Buna karşın, veri güdümlü yaklaşımlar ise düzenli bir gramere gereksinim duymadan, çoğunlukla önceden çözümlemesi insan tarafından yapılmış bir derlemdeki verilerden *tümevarımsal çıkarım*[×] yaparak tümce çözümlemesi yapmayı hedeflerler. Bu yöntem üç ana bileşenden oluşur (Nivre, 2006a):

1. Bağlılık grafiğini oluşturan *Ayrıştırma Algoritması*
2. Tümce için oluşturulabilecek olası sonuç çözümlemelerine değer atayan *Ayrıştırma Modeli*
3. Eğitim verisi üzerinden tümevarımsal çıkarım yapan *Öğrenme Yöntemi*

Tümevarımsal çıkarım yapan ayrıştırıcılar olasılık tabanlı[×] ve sınıflandırıcı tabanlı olmak üzere ikiye ayrılabilirler. Olasılık tabanlı yöntemler, x tümcesi için oluşturulan y sonuç çözümlemelerini P(x,y) birleşik olasılığını (üretimsel[×]) veya P(y|x) koşullu[×] olasılığını hesaplayarak değerlendirirler ve bu olasılıkları enbüyükleyen çözümü bulmaya çalışırlar. Sınıflandırıcı tabanlı yöntemler ise ayrıştırma işlemini P(y|x) koşullu olasılığını enbüyükleyen[4] sonuç çözümlemesini seçmek üzere bir çeşit sınıflandırma problemine dönüştürürler. Bulunması hedeflenen sınıflar ayrıştırıcının hareketleri, iki birimin bağlanıp bağlanmayacağı veya belirlenen bağlılıkların türleri gibi iki veya daha çok sınıftan oluşan kümeler olabilir.

1.2 Ayrıştırma ile İlgili Çalışmalar

Gramer güdümlü yaklaşımlar ayrıştırılması hedeflenen dilin kurallı bir dil olduğu varsayımında bulunmaktadırlar. Özellikle derleyici tasarımı sürecinde programlama dillerinin ayrıştırılması amacı ile kullanılan bu yöntemler, doğal dillerin ayrıştırılmasında bazı sorunlar ortaya çıkarmaktadırlar. Kabullenilen kurallı dil varsayımından ötürü, gramer güdümlü yaklaşımlar, düzenli olmayan doğal dillerin ancak tanımladıkları düzenli bir alt kümesini ayrıştırabilirler. Geçtiğimiz son yirmi yıl içerisinde, gramer güdümlü birçok çalışmada karşılaşılan bu kısıt, veri güdümlü yöntemlerden faydalanılarak aşılmaya çalışılmıştır. Bu çalışmalara örnek olarak

[4] Hesaplayarak veya hesaplamadan (ayırdedici[×])

İngilizce için en yüksek başarımları veren Collins (1997)'in ve Charniak (2000)'ın çalışmaları gösterilebilir. Bu çalışmalarda düzenli gramer, parametrelerinin bir derlem kullanılarak hesaplandığı istatistiksel bir gramere[5] dönüştürülmüş ve bu şekilde kullanılmıştır.

Özellikle tümce içi sözcük dizilişleri serbest olan doğal diller için düzenli bir gramer yazılması, İngilizce gibi sözcük sıralanışları katı dillere oranla çok daha zor ve emek yoğun bir işlemdir. Bu tür dillerde, sözcükler tümce içerisinde serbestçe yer değiştirebildikleri için oluşturulan gramerlerin kural sayısı oldukça yüksek olmakta ve belirsizlik miktarı artmaktadır.

Geçmiş on yıl içerisinde, veri güdümlü ayrıştırma alanında çok sayıda araştırma yapılmış ve bu yöntemin başarısında önemli artışlar sağlanmıştır. Bu tür ayrıştırıcıların eğitimi için yüzbinler mertebesinde tümce içeren ve genelde tümce çözümlemeleri insan tarafından yapılmış derlemlere gereksinim duyulmaktadır. Bu nedenle araştırmalar ilk olarak geniş kapsamlı bir derleme[6] sahip olan İngilizce üzerine yoğunlaşmıştır. Ancak son yıllarda diğer diller için oluşturulan derlemlerin sayısının artması ile beraber, araştırmalar farklı dillerin ayrıştırılmasına ve bunların ortaya koyduğu farklı sorunların incelenmesine yönelmiştir. Üzerinde yoğun olarak çalışılmış diller için geliştirilmiş modellerin az çalışılmış dillere uyarlanmasında sorunlarla karşılaşılmıştır. Birçok çalışmada, İngilizce için geliştirilmiş yüksek başarımlı istatistiksel ayrıştırıcıların başka dillere uygulanması sonucunda başarımda önemli oranda düşüşler gözlendiği raporlanmıştır. Bunlara örnek olarak Çekçe için Collins ve diğ. (1999)'nin, Çince için Bikel ve Chiang (2000)'in, Almanca için Levy ve Manning (2003)'in ve İtalyanca için Corazza ve diğ. (2004)'nin çalışmaları gösterilebilir. Collins (1999) doktora tezinde ayrıştırıcısının Çekçe üzerinde başarı gösterememesinin (Collins ve diğ., 1999) nedeninin, bu modelin Çekçe gibi sözcük dizilişleri serbest ve sözcük çekimliliğinin yüksek olduğu dillerde sorunlarla karşılaşması olduğunu belirtmiştir. Belirli bir dile özel bir ayrıştırıcının, özellikle

[5]Bu ayrıştırıcılarda kullanılan *görünüm bilgisi eklenmiş istatistiksel bağlamdan bağımsız gramer* yöntemi Bölüm 1.2.1'de açıklanacaktır.

[6]"Penn Treebank" (Marcus ve diğ., 1993) isimli İngilizce derlem yaklaşık üç milyon etiketlenmiş sözcükten oluşmaktadır.

kendisinden tamamen farklı özelliklere sahip başka bir dilde başarı sağlayamamasının iki önemli nedeni şu şekilde açıklanabilir (Nivre, 2006b):

- Ayrıştırıcının geliştirildiği dile özgü özelliklere aşırı uyum sağlayamaması,
- Kullanılan ayrıştırıcının yüksek başarım gösterebilmesi için yüksek miktarda eğitim verisine gereksinim duyması ve düşük boyutta derlemler üzerinde başarı gösterememesi.

1.2.1 Çözümleme Yöntemleri

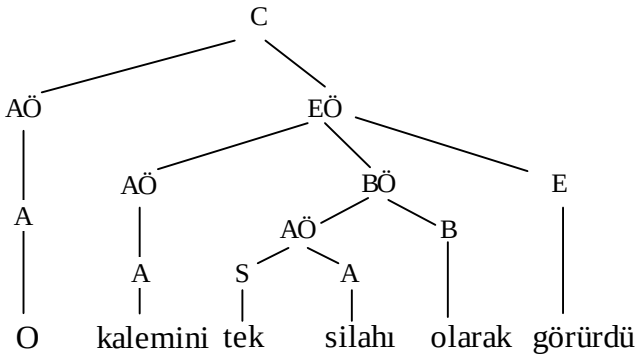
Bu bölümde, Bölüm 1.2.2'de bağlılık çözümlemesi yöntemi ile ilgili çalışmaların ayrıntıları verilmeden önce, diğer çözümleme yöntemleri ile ilgili özet bilgi verilecektir.

Ayrıştırma alanında en yaygın tanınan yaklaşımlardan biri *öbek yapısal gramerler*[×] olarak da anılan *bağlamdan bağımsız gramerler*dir[×]. Bu yaklaşım Chomsky (1957)'nin *üretimsel dönüşümlü dilbigisi*[×7] kuramına dayanır. Tümceleri öbeklere bölerek işlemeyi hedeflemektedir. Farklı kişiler tarafından yaklaşık aynı zamanlarda önerilen bağlamdan bağımsız gramerler (CFG) doğal dil işleme alanında birçok çalışmada kullanılmışlardır (Jurafsky ve Martin, 2000). Bu yaklaşıma göre, dilin temel ve kurucu birimi tümcedir. Tümce, ad öbeği ve eylem öbeği olmak üzere iki temel yapıdan oluşur. Bu öbekler de kendilerini oluşturan daha küçük öbeklerden kurulurlar. Dilin grameri bu öbekleri tanımlayan *yeniden yazım kurallarından* oluşurlar. Şekil 1.4'de Şekil 1.2'de verilen örnek tümcenin bu yaklaşıma göre gösterimi verilmiştir. Buradaki tümce aşağıdaki sekiz kural uygulanarak ayrıştırılmıştır:

C → AÖ EÖ , EÖ → AÖ BÖ E , BÖ → AÖ B , AÖ → S A | A ,
A → O | kalemini | silahı , S → tek , B → olarak , E → görürdü

CFG'lerin belirsizlik gidermedeki yetersizlikleri ve konuşma tanıma sistemleri için dilin modellenmesinin gerekliliği *olasılık tabanlı bağlamdan bağımsız gramerlerin* (PCFG) ortaya çıkmasına neden olmuştur. Ayrıştırıcı başarımının artmasını sağlayan bir diğer etken de *geçmişe dayalı*[×] modeller kullanılmasıdır. İlk olarak Black

[7]Kuramın Türkçe adı Hengirmen (2005)'in kitabından alınmıştır.



C:tümce, AÖ:Ad Öbeği, EÖ: Eylem Öbeği, BÖ: Belirteç Öbeği,
S:Sıfat, A:Ad, E:Eylem, B:Belirteç

Şekil 1.4: Öbekli Yapı Örneği

ve diğ. (1992) tarafından ortaya atılan bu modellerde bağlılıklar bulunurken ayrıştırma sırasında daha önceden alınan kararlar da özellik olarak sisteme dahil edilmektedirler. Bu özellikler genelde ayrıştırma sırasında kısmi olarak oluşmuş bağlılık ağacı yapılarıdır. Bir diğer söylemle, ayrıştırma sırasında önceden olan olayların yeni bağlılıklar bulunurken dikkate alınması durumudur. Daha sonraki dönemlerde PCFG'lerin de tek başlarına yetersiz olduğu sonucuna varılarak, bunlara görünüm bilgilerinin de eklenmesine gerek duyulmuştur. Yapılan araştırmalarda sözcükler arası ikili bağlılıkların öneminin görülmesiyle, her farklı sözcük için kuralların birer kopyasını oluşturan ve bu kurallar için olasılıkları ayrı ayrı hesaplayan, *görünüm bilgisi eklenmiş PCFG* (LPCFG)'ler ortaya çıkmıştır. İngilizce için bu tür gramerler kullanılarak yüksek başarımlı ayrıştırıcılar geliştirilmiştir. Bunlardan en iyi iki tanesi Collins (1997)'in ve Charniak (2000)'ın ayrıştırıcılarıdır. Öte yandan, Klein ve Manning (2003) çalışmalarında basit PCFG'lerin düşünüldüğü kadar kötü sonuçlar vermediğini, İngilizce için geliştirilen en iyi sayılan ayrıştırıcılarda (Collins, 1997; Charniak, 2000) sözcüklerin ikili bağlılık, birbirlerine olan uzaklık ve alt sınıf bilgilerinden yararlanıldığını, eğer bu bilgiler basit PCFG'de de kullanılırsa bunların da başarı oranının LPCFG'lerinki kadar yüksek olduğunu belirtmişlerdir. Oluşturdukları ayrıştırıcının başarısı, ilgili çoğu ayrıştırıcının başarısını geçmesine karşın, Collins (1997) ve Charniak (2000)'ın başarısını geçememiştir.

Yukarıda anlatılan yöntemlere ek olarak, DDA alanında bir çok farklı kuram yine tümceleri öbeklere bölerek işleme düşüncesine dayalı olarak geliştirilmiştir. Bunların en bilinenleri *Sözlüksel İşlevsel Gramerler*$^{\times}$ (LFG), *Kapsamlı Öbek Yapısal Gramerler*$^{\times}$ (GPSG), *Ağaç Birleştiren Gramerler*$^{\times}$ (TAG), *Baş sürümlü Öbek Yapısal Gramerler*$^{\times}$ (HPSG)'dir.

Tümce çözümlemesinde bir diğer yaklaşım ise ulamsal gramerler kullanmaktır. Ulamsal gramerlerde, sözdizimsel ulamlar, anlambilim gösterimlerini kodlamakta kullanılan araçlardır (Bozşahin, 1996). Bu kodlama, mantıksal işlemler kullanılarak sözdizim ve anlam çözümlemesini birleştirmeyi hedeflemektedir. Bu yaklaşımı kullanan yöntemlere örnek olarak *Birleşenli Ulamsal Gramerler*$^{\times}$ (CCG) ve *Mantıksal Tipli Gramerler*$^{\times}$ gösterilebilir.

Yukarıda söz edilen ayrıştırma yöntemlerinden bazıları çeşitli çalışmalarda Türkçe'nin ayrıştırması sürecinde kullanılmaya çalışılmıştır. Bunlara örnek olarak Güngördü ve Oflazer (1994) ve Çetinoğlu ve Oflazer (2006)'in LFG tabanlı, Güngör (2004)'ün PCFG tabanlı, Hoffman (1995), Bozşahin (2002), ve Çakıcı (2005) 'nın CCG tabanlı, Şehitoğlu ve Bozşahin (1996)'nin HPSG tabanlı ayrıştırıcıları gösterilebilir.

Anlatılanlardan anlaşılacağı gibi bu çalışmalardaki temel yaklaşım her dilin bir grameri olduğu ve gramerin de tümce içerisindeki sözcük diziliş kurallarını belirlediğidir. Bu varsayımdan hareketle eğer dilin grameri bilgisayarın anlayacağı şekilde ifade edilebilirse tümce çözümlenmiş olur. Yine bilindiği gibi bazı dillerde tümce içerisindeki sözcükler tümcenin genel anlamını etkilemeden yer değiştirebilir. Bu durumda bütün tümceleri kapsayacak tek bir gramer oluşturmak karmaşık ve sorunlu bir işlem haline gelir. Katı diller için gramer güdümlü yaklaşımların iyi sonuç vereceği, esnek diller için ise başarımın daha düşük olacağı açıktır.

1.2.2 Bağlılık Çözümlemesi

Collins İngilizce için yüksek başarı gösteren ayrıştırıcısını (Collins, 1997) tasarlarken ilk aşama olarak sadece sözcükler arası bağlılıklara dayanan bir ayrıştırıcı (Collins 1996) geliştirmiştir. 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. Tümce içindeki sözcüklerin ikili ilişkilerine bakarak tümcenin çözümlenmesi yöntemi, dilbilimcilerin çok eskilerden beri kullandıkları bir tekniktir. Bölüm 1.2.1'de anlatıldığı gibi, bilgisayarla tümce çözümlemesi üzerinde uğraşanlar yakın zamana kadar öbek yapısal gramer yöntemi kullanmayı yeğlemişlerdir. Ancak günümüzde sözcükler arası ikili bağlılıkların ayrıştırmanın başarısındaki önemli etkisinin görülmesi ile birlikte bağlılık gramerleri kullanılmaya başlanmıştır. Bu konuda çalışan araştırmacıların ortak kanısı, bağlılık gramerlerinin en önemli üstünlüğünün, sözcük dizilişleri serbest dillerin ayrıştırmasındaki yetenekleri olduğudur (Jurafsky ve Martin, 2000).

Ayrıştırma algoritmaları, en iyi bağlılık ağacını oluşturmak üzere ayrıştırma modelinden faydalanırlar. Son yıllara kadar, özellikle istatistiksel ayrıştırma alanında ayrıştırma algoritması olarak çeşitli dinamik programlama algoritmaları kullanılmıştır. Bunlara örnek olarak Eisner (1996), Collins (1996), Haruno ve diğ. (1998), Uchimoto ve diğ. (1999), Sekine ve diğ. (2000), Kudo ve Matsumoto (2000)'nun çalışmaları gösterilebilir. Bu algoritmalar istatistiksel bir ayrıştırma modelinin kendilerine verdiği ikili bağlılık olasılıklarını kullanarak, arama uzayında yer alan en yüksek olasılıklı ayrıştırma ağacını bulmaya çalışırlar. Bu yöntemden tamamen farklı olan bir başka yaklaşım da gerekirci ayrıştırma algoritmaları kullanmaktır. Birçok çalışmada (Kudo ve Matsumoto, 2000; Yamada ve Matsumoto, 2003; Oflazer, 2003; Nivre, 2003) bu algoritmaların yüksek başarımı gösterilmiştir. 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.

Bu alanda gerçeklenen çalışmalarda genel olarak bazı varsayımlar kabul edilmektedir. Bu varsayımlardan herkes tarafından benimseneni, bağlılık grafiğindeki her düğümün en fazla bir adet düğüme bağlı olmasıdır. Bir diğer varsayım ise grafiğin bağlı ve döngülere izin vermeyen yapıda, diğer bir deyişle, köklü bir ağaç yapısında olmasıdır. Kesişmeyen bağlılık varsayımı ise doğal dillerde her zaman geçerli olmaması nedeniyle (Tapanainen ve Järvinen, 1997; McDonald ve diğ., 2005a; Nivre

ve Nilsson, 2005) teknik olarak sorgulanmasına rağmen, yüksek başarım gösteren birçok ayrıştırıcıda benimsenmiştir.

Eisner (1996)'in 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. Ayrıştırıcı, uydu sözcüğü bir iye sözcüğe bağlamadan önce, oluşacak bağlılık oku altında kalan tüm sözcüklerin bağlılıklarını kurmayı hedeflemektedir.

Haruno ve diğ. (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. Bu çalışmada Collins (1996)'in çalışmasına atıfta bulunularak, oradaki modelde olasıklıkların hesaplanması için belirli özellikler seçildiği ve parçacık türüne bakılmaksızın her parçacık için aynı özelliklerin kullanıldığı vurgulanmaktadır. Haruno ve diğ. (1998)'nin çalışmasında ise olasılıklar karar ağaçlarının kullanılması ile önceden belirlenen bir özellik kümesi içerisinden özelliklerin parçacık türüne göre otomatik olarak seçilmesi ile hesaplanır. Ayrıştırma algoritması olarak dinamik programlama kullanılmıştır.

Uchimoto ve diğ. (1999) ve Sekine ve diğ. (2000)'nin yaptığı çalışmalarda ise bağlılık olasılıkları hesaplanırken seçilen özelliklerin önem ağırlılıkları *en büyük bilgi değeri*[×] yöntemi yoluyla eğitim verisi içerisindeki sıklıklarına bakılarak hesaplanır. Bu iki çalışmada da ayrıştırma algoritması olarak *geriye doğru demetli arama*[×] algoritması kullanılmıştır. Daha önce de bahsedildiği gibi Japonca'da parçacıklar sadece sağ taraflarındaki parçacıklara bağlanabilirler. Bu nedenle çözümlemeyi, tümce sonundan başlayarak başa doğru (sağdan sola) yapmak, özellikle kesişen bağlılık olmayacağı varsayımının benimsenmesi ile, çözümlemeyi önemli ölçüde kolaylaştırmaktadır. Bu çalışmalarda yapılan deneylerin sonuçları Haruno ve diğ. (1998)'nin sonuçları ile

karşılaştırıldığında elde edilen başarının yaklaşık olarak aynı olduğu tespit edilmiştir. Buradan yola çıkarak, özellik seçiminin, bir diğer deyişle sınıflandırma yapılırken kullanılan özelliklerin hangileri olacağı seçiminin, hangi istatistiksel modelin (karar ağaçları, en büyük bilgi değeri vs...) seçileceğinden çok daha önemli olduğu vurgulanmıştır.

Makine öğrenmesi[x] çatısına dayanan geleneksel ayrıştırma algoritmaları sınıflandırma için gerekli özellikleri seçmede veya bu özelliklerin gerekli birleşimlerini oluşturmada başarısız kalmaktadırlar. *Karar destek makinelerinin*[x] (KDM) (Vapnik, 1995) bu konudaki yetenekleri, doğal dil işleme alanında sıkça kullanılmalarına neden olmuştur. KDM'ler çok yüksek boyutlu özellikler üzerinde bile yüksek başarım sergilemektedirler. Kudo ve Matsumoto (2000) yaptıkları çalışmada KDM'leri Japonca'nın bağlılık çözümlemesi alanında kullanmışlardır. Ancak bu çalışmada, KDM'leri sınıflar arasında karar vermede kullanmak yerine olasılık değerlerini hesaplamak için kullanmışlardır. Bu amaçla, örneklerin sınıfları ayıran düzleme olan uzaklıklarını sigmoid fonksiyonundan geçirerek olasılık değerleri olarak kullanmış ve yine Sekine ve diğ. (2000)'nin dinamik programlama algoritmasını kullanmışlardır. Elde edilen sonuçlar Uchimoto ve diğ. (1999) ve Sekine ve diğ. (2000)'nin sonuçlarından daha başarılıdır.

McDonald ve diğ. (2005a; 2005b) çalışmalarında bağlılık çözümlemesini yönlü bir grafikte maksimum kapsayan ağacı[x] bulma sorunu olarak işlemiş ve İngilizce ve Çekçe üzerinde denemiştir. Asıl olarak Eisner (1996)'in ayrıştırma algoritmasını kullanan bu ayrıştırıcı Chu-Liu-Edmunds (McDonald ve diğ., 2005a) algoritmasının kullanımıyla kesişen bağlılık içeren tümceleri de çözümleyebilmektedir. Bu çalışmada da KDM'lere benzer bir *aralık büyükleme sınıflandırıcısı* olasılık değerlerini hesaplamak amacıyla kullanılmıştır.

Yukarıda anlatılan istatistiksel modellerin tümünde öncelikle sözcükler arası bağlılıkların olasılıkları hesaplanmakta, sonrasında ise olası bütün bağlılıkların içerisinden en uygun bağlılık çözümü seçilmektedir. Bu işlem sırasında bağlılıkların birbirlerinden bağımsız oldukları varsayımında bulunulmaktadır. Bu tür bir modelde, bağlılıkların sadece soldan sağa doğru yönlendiği varsayımı yapılsa bile, n sözcüklü bir tümce için n(n-1)/2 adet eğitim verisine gereksinim duyulmaktadır. Kudo ve

Matsumoto (2002) çalışmalarında, Japonca için, bu modellere oranla daha az eğitim verisine gereksinim duyan, eğitim süresini kısaltan, bağlılıkların birbirlerine bağımlı olduklarını kabul eden, gerekirci[×] bir ayrıştırma algoritması (*basamaklı birleştirme*[×]) geliştirmişlerdir. Bu çalışmada elde edilen sonuçlar Japonca için, yukarıda bahsedilen ayrıştırıcıların tümünün sonuçlarından daha başarılıdırlar.

Yamada ve Matsumoto (2003), Kudo ve Matsumoto (2002)'nun basamaklı birleştirme ayrıştırıcılarına benzer bir çalışmayı İngilizce için yapmışlardır. Ancak bu çalışmada, bağlılık çözümlemesi için aşağıdan yukarıya işlem yapan gerekirci bir model geliştirmişlerdir. Bu modelde ayrıştırma iki aşamadan oluşmaktadır. İlk aşamada bağlılık çözümlemesi yapılan sözcüğün etrafındaki diğer sözcüklerin bilgileri çekilmekte, ikinci aşamada ise bu bilgilere dayanarak ayrıştırıcının tanımlı üç farklı hareketinden (ötele -S-, sağ -R-, sol -L-) bir tanesine sınıflandırıcı kullanılarak karar verilmektedir. Bu karar verilirken sınıflandırıcı olarak KDM kullanılmıştır. Üç sınıf arasında karar verilmeye çalışıldığı için L-R, L-S ve R-S arasında sınıflandırma yapmak üzere üç adet KDM tasarlanmıştır. KDM'nin öğrenme sürecinin hızını arttırmak için eğitim verisi, bağlılığın solundaki sözcüğün etiketine göre kümelere ayrılmış, her küme için ayrı KDM tasarlanmış, daha sonra sınama sırasında ilgili sözcüğün etiketine bakılarak gerekli KDM çalıştırılmıştır. Sonuçlar Collins (1997)'in ve Charniak (2000)'ın sonuçları ile karşılaştırılmış ve başarının bu çalışmalara göre çok az daha düşük olduğu görülmüştür. Bu çalışmada tümce yapısı ile ilgili hiçbir bilgi kullanılmadığı gözönünde bulundurulursa, sonuçların iyi sayılabileceği belirtilmiştir.

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. SDD'lerin birden çok kez uygulanması açısından, Kudo ve Matsumoto (2002)'nun basamaklı birleştirme yöntemine benzerlik göstermektedir. Ayrıştırıcı, bağlılıklar ile birlikte bağlılık türlerini de belirlemektedir. Bu çalışma Collins (1999)'in doktora tezinde bahsettiği yüksek çekimli dillerin ayrıştırılmasının nasıl bir yöntemle yapılabileceğine bir örnektir. Çok

çekimli diller ayrıştırma konusunda iki büyük soruna yol açmaktadırlar (Collins, 1999). Bunlardan birincisi sözcük etiketlerinin durum, kişi, sayı, cinsiyet gibi birçok bilgiyi taşımaları ve bu nedenle çok sayıda etiket oluşmasıdır. Seyrek veri sorununa yol açmadan, etiketlerin taşıdığı bilgilerin tümünü kullanabilmek için yöntemler geliştirilmelidir. İkincisi ise yine seyrek veri sorununu ortaya çıkaran sözcük çeşitliliğindeki çokluktur. Sınama verisinde, eğitim verisinde rastlanmayan birçok sözcük ortaya çıkmaktadır.

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. 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. Nivre ve diğ. (2004) yaptıkları çalışmada yine İsveççe için Nivre (2003)'nin çalışmasına benzer yığın yapılı bir ayrıştırıcı geliştirmişlerdir. Bu çalışmadan farklı olarak, bu sefer bağlılık etiketleri (türleri) de belirlenmeye çalışılmış ve insan tarafından yazılmış gramer kuralları yerine *bellek tabanlı*[↗] (BT) bir sınıflandırıcı kullanılmıştır. BT sınıflandırıcı bir çeşit *k en yakın komşu*[↗] algoritmasıdır. Sınıflandırmada kullanılan özelliklere farklı ağırlıklar verme, uzak ve yakın komşulara farklı ağırlıklar verme gibi yeteneklere sahiptir. 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. Bu yaklaşım Yamada ve Matsumoto (2003)'nun KDM'leri kullanarak yaptıkları çalışmalarına benzer bir yaklaşımdır. Burada BT kullanılmasının nedeni bağlılık etiketlerinin de belirlenmeye çalışılması ile birlikte çok sınıflı bir sınıflandırıcıya gereksinim duyulması olarak açıklanmıştır. Yamada ve Matsumoto (2003)'nun çalışmasından farklı olarak, ayrıştırma işlemini

veri üzerinden tek geçişte tamamlamaya çalışmışlardır. Nivre ve Scholz (2004) yaptıkları çalışmada aynı modeli İngilizce için "Penn Treebank" verisi üzerinde uygulamış ve sonuçların, bağlılık türleri belirlenmese bile, Yamada ve Matsumoto (2003)'nunkinden çok kötü olduğunu belirlemişlerdir. Bunun nedeninin kullandıkları tek geçişte ayrıştırma yönteminden kaynaklandığını ifade etmişlerdir. Başarı oranının düşük olmasının kullandıkları sınıflandırıcının KDM'lere kıyasla genelde başarısız bir yöntem olmasından kaynaklandığı yorumu yapılabilir.

Türkçe'nin veri güdümlü bağlılık çözümlemesi konusunda ilk çalışma Eryiğit ve Oflazer (2006)'nin çalışmasıdır. Bunun yanısıra son bir sene içerisinde birçok çalışmada Türkçe'nin bağlılık çözümlemesi yapılmıştır: Eryiğit ve diğ. (2006a), Eryiğit ve diğ. (2006b), Nivre ve diğ. (2006a), Attardi (2006), Wu ve diğ. (2006), Nivre ve diğ. (2006b), Bick (2006), Canisius ve diğ. (2006), Carreras ve diğ. (2006), Cheng ve diğ. (2006), Corston-Oliver ve Aue (2006), Chang ve diğ. (2006), Liu ve diğ. (2006), McDonald ve diğ. (2006), Johansson ve Nugues (2006), Riedel ve diğ. (2006), Schiehlen ve Spranger (2006), Shimizu (2006), Dreyer ve diğ. (2006), Wu ve diğ. (2006), Yüret (2006). Bu çalışmalar ile ilgili ayrıntılı incelemeler tezin ilerleyen bölümlerinde verilecektir. Çalışmalar içerisinde Türkçe ağaç yapılı derlem üzerinde en yüksek başarım Eryiğit ve diğ. (2006b)'in çalışmasında tanıtılan ayrıştırıcı ile elde edilmiştir.

1.3 Tezin Katkısı

Türkçe bitişken, ayrıca tümce içi öğe dizilişleri serbest bir dildir. Tümce içerisindeki sözcükler genelde sağa bağımlıdırlar. Sıralanan bu özellikleri ile, İngilizce'den ve ayrıştırma alanında üzerinde yoğun olarak çalışılmış birçok dilden farklılıklar göstermektedir. Bu niteliği ile benzer özellikler gösteren bir sınıf dilin temsilcisi olarak görülebilir. Bu dillere örnek olarak diğer Türki diller, Fince, Estonyaca, Macarca, Japonca ve Korece gibi diller gösterilebilir.

Tümce çözümlemesi konusunda yapılmış çalışmalar incelendiğinde çalışmaların çoğunun Hint-Avrupa dil ailesi ve özellikle İngilizce üzerinde yapıldığı görülmektedir. Türkçe'nin de içinde yer aldığı Ural-Altay dil ailesi için yakın zamanda birçok

araştırma başlatıldığı görülmektedir. Yukarıda değinildiği gibi, tümce çözümlemesi tümce yapısına bağlıdır. Hint-Avrupa dillerinin tümce yapısı ile Ural-Altay dillerinin tümce yapıları çok farklı olduğundan İngilizce için yapılmış olan çalışmaların Türkçe için kullanılabilmesi olanaklı değildir.

Bu tez çalışmasının hedefi, Türkçe tümcelerin çözümlemesini bağlılık ayrıştırması yöntemini kullanarak en yüksek başarımla gerçeklemektir. Bu amaçla:

- Türkçe'nin tümce yapısı bağlılık açısından incelenmiş,
- Türkçe'nin bağlılık yapısı modellenmiş,
- Farklı nitelikte ayrıştırıcılar geliştirilerek, ayrıştırıcıların ve modellerin başarımları karşılaştırılmış,
- Sonuç olarak ayırdedici öğrenmeye dayalı sınıflandırıcı tabanlı gerekirci ayrıştırıcının en iyi sonucu verdiği ortaya konmuştur.

Bu çerçevede, çalışmamızın bilime yaptığı katkılar aşağıda sıralanmıştır:

- Daha önce başka diller için gerçeklenmiş olan ayrıştırıcılar, bitişken olmayan dilleri ayrıştırmak üzere tasarlanmışlardır. Bu nedenle Türkçe için kullanıldıkları zaman başarımları düşük olmaktadır. Bu çalışmada geliştirilen ayrıştırma yöntemi, sözcüğün kökü, ekleri ve biçimbilimsel yapısını da dikkate alarak çalışmaktadır.
- Sözcüklerin ikili bağımlılıklarını araştırmada kullanılabilecek çok sayıda yöntemin varlığı bilinmektedir. Bu yöntemler bağlılıkları belirlerken sözcüklerin farklı özelliklerinden faydalanırlar. Bunlardan bazıları sözcüklerin metin içerisindeki *görünüm şekilleri*, *nitelikleri*, *biçimbilimsel özellikleri*, *komşu özellikleri*, *yakınlık durumları* gibi özelliklerdir. Bağlılık araştırması yaparken bu özelliklerin sayısının arttırılarak kullanılması halinde bulunan çözüm belirginliği artmakta ancak sonuca ulaşma olasılığı düşmektedir. Kullanılan özellik sayısı azaltıldığı durumda ise çözüm olasılığı yükselmekte ancak belirsizlik artmaktadır. Bu çalışmada Türkçe için bağlılık ayrıştırmasında hangi özelliklerin kullanılması halinde en iyi çözümün bulunacağı gösterilmiştir.

Çalışmalarımız sırasında, yakın geçmişte yayınlanan Türkçe ağaç yapılı derlem kullanılarak, veri güdümlü ayrıştırıcılarda farklı tasarım yöntemlerinin kullanılmasının etkileri incelenmiştir. Bu incelemeler sırasında, temel model olarak alınan bazı kural tabanlı ayrıştırıcılar, olasılık tabanlı modele dayalı bir istatistiksel ayrıştırıcı ve ayırdedici öğrenmeye dayalı sınıflandırıcı tabanlı gerekirci bir ayrıştırıcı olmak üzere farklı ayrıştırma yöntemlerine sahip ayrıştırıcılar kullanılmış ve tasarım yöntemlerinin etkileri bunlar üzerinde değerlendirilmiştir. Daha sonra, ayrıştırmada çekim kümesi adı verilen biçimbilimsel birimleri, biçimbilimsel özellikleri ve görünüm bilgisi kullanmanın etkileri incelenmiştir. Ayrıştırıcıların sonuçları üzerinde incelemeler yapılmış ve başarımları ilgili yayınlardaki ayrıştırıcıların başarımları ile karşılaştırılmışlardır.

Sonuçlar, sözcükler yerine sözcüklerden daha küçük olan çekim kümelerinin tümce yapısının ana birimleri olarak kullanılmasıyla, Türkçe'de ayrıştırma başarımının arttırılabileceğini göstermektedir. Ayrıca biçimbilimsel özelliklerin ve görünüm bilgisi eklemenin, Türkçe'nin bağlılık çözümlemesinde çok önemli etkisi olduğu görülmüştür. Ancak, bu bilgileri tümüyle kullanmanın bazı ayrıştırıcıların başarımlarını kötü yönde etkilediği gösterilmiştir. Seçilen ayrıştırıcının niteliklerine bağlı olarak görünüm bilgisinin veya çekimsel özelliklerin kısmi olarak kullanılması önerilmiştir.

Türkçe için geliştirilen ilk veri güdümlü bağlılık ayrıştırıcıları bu tez kapsamında yapılan araştırmalar sonucunda ortaya çıkmıştır. Bu tez çalışmasının sürdürüldüğü sırada benzer çalışmaların yapıldığı gözlemlenmiştir. Bu tezde geliştirilen yöntem ve aynı konuda yapılan diğer çalışmalar Haziran 2006 tarihinde CoNLL-X ortak çalışmasında aynı veri kümesi üzerinde sınanmıştır. Geliştirilen ayrıştırıcının diğer ayrıştırıcılara oranla en yüksek başarıyı verdiği gösterilmiştir. Türkçe ile ilgili araştırmaların sayısındaki hızlı artış aşağıdaki dört nedene bağlanabilir:

1. ODTÜ-Sabancı Türkçe Ağaç Yapılı Derlemi (Oflazer ve diğ., 2003; Atalay ve diğ., 2003)'nin yayınlanması,
2. Türkçe'nin bağlılık çözümlemesi ile ilgili yapılan örnek çalışmalar (Eryiğit ve Oflazer, 2006; Nivre ve diğ., 2006a),

3. Türkçe'nin bitişken yapısı ile benzer birçok dile örnek teşkil etmesi,
4. Bağlılık Çözümlemesi yöntemine olan ilginin giderek artması.

Yukarıda sıralanan bu gelişmeler sayesinde Türkçe, yukarıda sözü edilen ortak çalışmada işlenen dillerden biri olarak seçilmiş ve böylece birçok araştırma grubu tarafından çözümlemesi yapılmıştır. Tez sırasında, geliştirilen yeni modellere ek olarak Türkçe ağaç yapılı derlemin güncellenmesi ve ortak çalışmaya hazırlanması konusunda da çalışmalar yapılmıştır.

1.4 Tezin Bölümleri

Toplam altı bölümden oluşan tezin içeriği aşağıda açıklanmıştır.

- *Bölüm 2 - Türkçe'nin Özellikleri*

  Bu bölümde Türkçe'ye özgü bağlılık ayrıştırması ile ilgili özellikler kısaca özetlenmiştir. Ayrıca bu bölümde, tezde eğitim ve sınama verisi olarak kullanılan ODTÜ-Sabancı Ağaç Yapılı Derlemi tanıtılacak ve tez kapsamında bu derlem üzerinde yapılan iyileştirmeler anlatılacaktır.

- *Bölüm 3 - Türkçe'nin Bağlılık Ayrıştırması*

  Bu bölümde, Türkçe'nin bağlılık ayrıştırması konusunda incelemeler yapılmış ve bu incelemelerin sonuçları verilmiştir. Aynı bölümde, farklı ayrıştırıcılar üzerinde Türkçe'ye özgü tasarım modellerinin etkileri irdelenmiştir. Bölüm 3.1, veri güdümlü ayrıştırıcıların başarımlarını karşılaştırmak üzere geliştirilmiş olan üç temel ayrıştırıcıyı tanıtmaktadır. Bölüm 3.2'de, olasılık tabanlı modele dayalı istatistiksel ayrıştırıcı tanıtılmaktadır. Bu kısımlarda kullanılan ayrıştırıcıların amaçları derlemin sadece sağa bağımlı ve kesişmeyen bağlılıklar içeren bir alt kümesi üzerinde etiketsiz bağlılıkları bulmaktır. Çalışmamız kapsamında geliştirilen bu ilk modellerde, ayrıştırma birimleri olarak sözcüklerden daha küçük olan biçimbilimsel birimleri kullanmanın başarımı arttırdığı bu bölüm içinde anlatılmıştır. Bölüm 3.3'de, Bölüm 3.2'de elde edilen bilgiler doğrultusunda derlemin tamamı üzerinde etiketli$^{\times}$ ve etiketsiz$^{\times}$ bağlılıkları bulmak üzere tasarlanan ayırdedici öğrenmeye dayalı sınıflandırıcı tabanlı bir

ayrıştırıcı tanıtılmıştır. Önceki kısımın sonuçlarına paralel olarak bu kısımda da biçimbilimsel birimleri ayrıştırma birimi olarak kullanmanın başarımı arttırdığı gösterilmiştir.

- *Bölüm 4 - İncelemeler ve Tartışma*

  Bu bölümde, önceki bölümlerde elde edilen bilgiler ışığında, geliştirilen modeller üzerinde iyileştirmeler ve sonuçların karşılaştırılması yapılmıştır. Buna ek olarak, biçimbilimsel ve görünüm bilgisi özellikleri kullanmanın etkileri, modeller üzerinde ayrıntılı olarak incelenmiş ve yorumlanmıştır. Ayrıca, eğitim verisinin boyutu, yetkin$^{\times}$ etiketlerin kullanılmasının etkileri ve hata incelemesi yapılmıştır. Bölümün son kısmı, geliştirilen ayrıştırıcının yayınlanan diğer bağlılık ayrıştırıcıları ile karşılaştırılmasına ayrılmıştır.

- *Bölüm 5 - Sonuçlar ve Öneriler*

  Bu bölüm, tez sonucunda ortaya çıkan bulguların kısa bir özetini ve gelecek araştırmalar için önerileri içermektedir.

2. TÜRKÇE'NİN ÖZELLİKLERİ

Bu bölümde, Türkçe'nin bağlılık ayrıştırması için gerekli olan ön bilgiler özet şeklinde verilecektir. Bunlar Türkçe'nin biçimbilimsel yapısı, bağlılık yapısı ve Türkçe derlem[1] ile ilgili bilgilerdir. Konular ile ilgili ayrıntılı bilgiler şu çalışmalardan elde edilebilir: Oflazer (1994), Cebiroğlu (2002), Oflazer (2003), Oflazer ve diğ. (2003), Atalay ve diğ. (2003), Bozşahin (2000). Bu bölümdeki bilgiler, tezin bütünlüğünü sağlamak amacıyla verilmiştir.

2.1 Türkçe'nin Biçimbilimsel Yapısı

Bitişken bir dil olan Türkçe çok zengin biçimbilimsel bir yapıya sahiptir. Sözcükler sonlarına ardarda ekler konularak yüzlerce farklı sözcüğe dönüştürülebilirler. Birçok dilde sözcükten ayrı olarak yazılan ilgeçler, Türkçe'de genelde bir sonek olarak kelimeye eklenip tek bir sözcük oluştururlar. Benzer şekilde kişi, yardımcı eylem gibi birçok ayrı yazılan sözcük, Türkçe'de yine ekler vasıtasıyla ifade edilirler. Bu nedenle, bir başka dilde uzun bir tümce ile ifade edilen deyişlerin Türkçe'de tek bir sözcük ile ifade edilmesi çok rastlanan bir durumdur. Türkçe'de bir sözcüğün ekler yardımı ile dönüştürülebileceği farklı sözcük sayısı kuramsal olarak sonsuzdur. Her ne kadar günlük dilde çok kullanılan bir yapı olmasa da "Osmanlılaştıramadıklarımızdanmışsınızcasına" türünde örnekler bir çok yazar tarafından Türkçe'nin bu özelliğine dikkat çekmek üzere gösterilmiştir.

Türkçe'de sözcükler sonlarına eklenen bu eklerle farklı türde sözcüklere de dönüşebilirler; eylemler isimlere, isimler eylemlere vb ... . Türkçe'nin bu özelliği ilgili yayınlarda (Oflazer ve diğ., 2003; Oflazer, 2003; Eryiğit ve Oflazer, 2006; Hakkani-Tür ve diğ., 2002) sözcüklerin *çekim kümelerine*[⋊] (*ÇK*) ayrılması biçiminde

[1] Burada anlatılmak istenen esas olarak ağaç yapılı derlemdir. "ağaç yapılı derlem" tez içerisinde kısaltılarak "derlem" olarak anılmıştır.

gösterilmektedir. Bu gösterimde, Türkçe bir sözcüğün bir dizi çekim kümesinden oluştuğu ve bu *ÇK*'lerin *türetim sınırlarından*$^{\wedge}$ (TS) bölündüğü varsayılmaktadır. Bu özellik aşağıdaki gibi gösterilmektedir:

$$\texttt{gövde+ÇK}_1 + \texttt{\^{}TS+ÇK}_2 + \texttt{\^{}TS+}\cdots + \texttt{\^{}TS+ÇK}_n.$$

Burada her $ÇK_i$, ilgili olduğu çekim kümesine ait biçimbilimsel özellikleri ve sözcük sınıflarını belirtmektedir. Bu çalışmada kullanılan Türkçe derlem hazırlanırken, sözcüğün biçimbilimsel çözümlemesi olarak adlandırılan bu işlem için Oflazer (1994)'in iki seviyeli biçimbilimsel çözümleyicisi kullanılmıştır. Aşağıdaki örnekte, türemiş bir niteleyici olan "sağlamlaştırdığımızdaki" sözcüğünün biçimbilimsel çözümleme sonucundaki hali derlem gösterimi kullanılarak gösterilmektedir.

```
sağlam+Adj[2]
      +^TS+Verb+Become
      +^TS+Verb+Caus+Pos
      +^TS+Noun+PastPart+A3sg+P1pl+Loc
      +^TS+Adj+Rel
```

^TS sınırları sözcük üzerinde gösterilmek istenirse şöyle görünecektir:

$$\underbrace{|\ \text{sağlam}\ |}_{ÇK_1}\texttt{\^{}TS}\underbrace{|\ \text{laş}\ |}_{ÇK_2}\texttt{\^{}TS}\underbrace{|\ \text{tır}\ |}_{ÇK_3}\texttt{\^{}TS}\underbrace{|\ \text{dığımızda}\ |}_{ÇK_4}\texttt{\^{}TS}\underbrace{|\ \text{ki}\ |}_{ÇK_5}$$

Buradaki beş çekim kümesi, ^TS türetim sınırı işaretleri ile birbirinden ayrılmış özellik dizileridir. İlk *ÇK* gövdenin tek özelliği olan sözcük sınıfını göstermektedir. "sağlam" sözcüğü bir sıfattır. İkinci *ÇK*, önceki sıfata "oluşmak" anlamı katılarak bir eylem türetmeyi göstermektedir. Üçüncü *ÇK* önceki eylemden olumlu bir ettirgen eylemin

[2]Derlem gösterimi ile belirtilmiş biçimbilimsel özellikler ve sınıflar şöyledir: +Adj: Sıfat, +Verb: Eylem, +Become: oluşmak, +Caus: Ettirgen, +Pos: olumlu, +Noun: İsim, +PastPart: geçmiş zaman ortacı, +A3sg: 3. tekil kişi kişi/sayı uyum imi, +P3sg: 3. tekil kişi iyelik imi, +Loc: -de hali, +Rel: ilişkilendirici. Bundan sonraki bölümlerde, kullanılan gösterim ile ilgili bilgi Ek B'de verilmektedir.

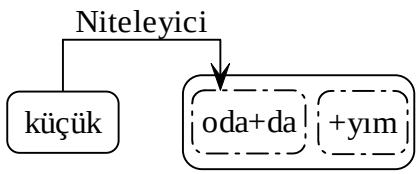
türetildiğini belirtmektedir. Dördüncü *ÇK* alt sözcük sınıfı olarak geçmiş zaman ortacı taşıyan, birinci çoğul kişi iyelik ve -de hal eki almış bir isimin türetilmesini belirtmektedir. Son olarak da, beşinci *ÇK* ilişkilendirici bir sıfat türetilmesini belirtmektedir.

2.2 Türkçe'nin Bağlılık Yapısı

Türkçe'nin türetim sistemi çok üretkendir ve bir sözcüğün uydu veya iye olarak içerisinde bulunduğu tümce yapısı ilişkileri, sözcüğün içerdiği bir veya daha fazla türemiş yapının biçimbilimsel özellikleriyle belirlenmektedir.

Bağlılıkları sadece sözcükler arasında göstermek ayrıştırma işlemi için yeterince anlamlı bilgi taşımamaktadır. Şekil 2.1 'de "*küçük odadayım*" tümceciği içerisindeki bağlılık gösterilmektedir. Şekildeki gösterimde sözcükler dikdörtgenler içerisinde ve bunların içerdikleri çekim kümeleri de noktalı dikdörtgenler içerisinde gösterilmektedirler. Şekilde "odadayım" sözcüğü iki *ÇK*'den oluşmaktadır: $\underbrace{oda\text{+Noun+A3sg+Pnon+Loc}}_{ÇK_1}$ ^*TS* $\underbrace{\text{Verb+Pres+A1sg}}_{ÇK_2}$. Birinci *ÇK* "oda" isimi ve bu isime ait biçimbilimsel özellikleri içermektedir. Bu özellikler, isimin tekil, iyelik eki almamış ve -de halinde olduğunu belirtmektedir. İkinci *ÇK* ise bu isimden türemiş "odada olmak" eylemini ve biçimbilimsel özelliklerini içermektedir. Eylem birinci tekil kişi eki almıştır ve şimdiki zamandadır. Örnekte "küçük" olan, "odadayım" sözcüğü değil "oda" dır. "odadayım" isimden eyleme dönüşmüş bir sözcüktür. İki sözcük arasında kurulan bağlantı "odadayım" sözcüğünün eyleme dönüşmeden önceki isim halinden kaynaklanmaktadır. Bu durum sıfatların genel olarak isimlere bağlanması kuralından kaynaklanmaktadır. Buradan yola çıkarak, ayrıştırıcının bulduğu bağlılıklar sadece uydu ve iye sözcüğü değil bu sözcüklerin uydu ve iye *ÇK*'lerini belirtmelidirler. Bu tümcecikte, "küçük olan nedir?" sorusunun cevabı bir sözcük değil, bir çekim kümesidir.

Şekil 2.2 "*Bu okuldaki öğrencilerin en akıllısı şurada duran küçük kızdır*" tümcesi üzerinde çekim kümelerini ve bunlar arasında oluşan bağlılıkları göstermektedir. Bağlılıkların yönü uydu *ÇK*'den iye *ÇK*'ye doğru gösterilmişdir. Bağlılık türleri

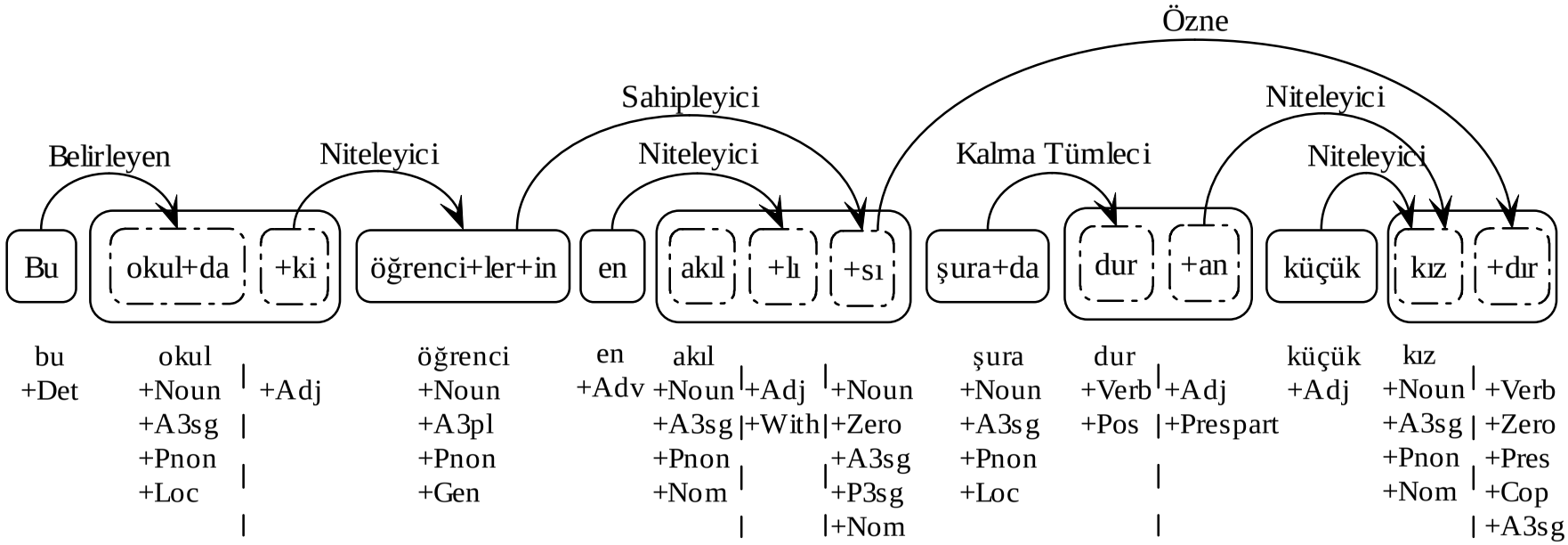


Şekil 2.1: Türkçe'de Bağlılık Yapısı

bağlılık oklarının üzerinde belirtilmektedir. Her sözcüğe ait biçimbilimsel çözümleme ilgili sözcüğün altında derlem gösterimiyle verilmiştir. Bağlılıklar uydu sözcüğün sadece son *ÇK*'sinden çıkmaktadırlar. Bu nedenle, şekilde bağlılık çıkmayan bazı *ÇK*'ler bulunmaktadır (örn., okuldaki sözcüğünün ilk *ÇK*'si). Bu tip *ÇK*'ler bağlılıklarda sadece iye olarak bulunurlar. Bu ara *ÇK*'lerin arkalarından gelen *ÇK*'ye biçimbilimsel olarak bağlı oldukları varsayılır. Ancak bu bağlılıklar özellikle belirtilmez. İye *ÇK* ise iye sözcüğün herhangi bir *ÇK*'si olabilir. Bir başka deyişle, bağlılık herhangi bir sözcüğün herhangi bir *ÇK*'sinde sonlanabilir. "n" adet sözcükten oluşan bir tümcede "n-1" adet bağlılık vardır (Şekilde, 9 adet sözcük arasında oluşan 8 adet bağlılık gösterilmektedir). Tümcenin bağlılık ağacının kökü olarak nitelenen sözcük herhangi başka bir sözcüğe bağlanmaz. Şekilde bu sözcük en sonda yer alan ana eylemdir ("kızdır").

Şekilden de görülebileceği gibi bir sözcükten sadece bir bağlılık oku çıkarken, birden fazla bağlılık oku girebilmektedir. Bir diğer deyişle, her sözcüğün sadece bir iyesi vardır ancak bir iye sözcüğün birden fazla uydusu olabilir. Birden çok *ÇK* içeren sözcüklere gelen bağlılıklar sözcüğün farklı *ÇK*'lerinde sonlanabilir. Şekildeki "akıllısı" sözcüğü bu duruma güzel bir örnek teşkil etmektedir. "en" sözcüğü "akıllısı" sözcüğünün ikinci *ÇK*'sine ("en akıllı") bağlıdır. "öğrencilerin" sözcüğü ise aynı sözcüğün üçüncü *ÇK*'sine bağlıdır ("öğrencilerin akıllısı").

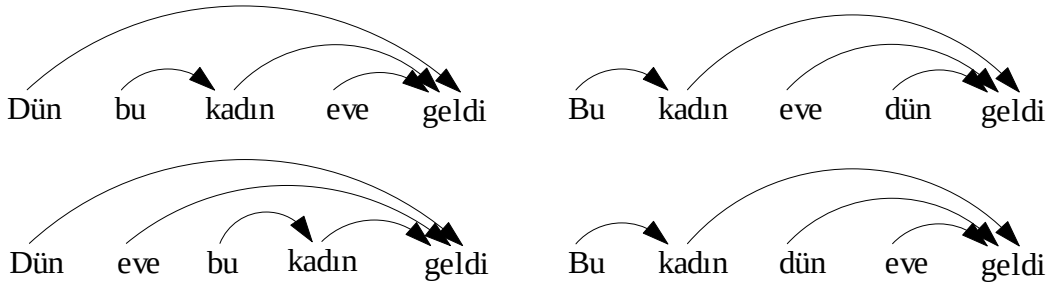
Kök sözcük dışındaki tüm sözcükler son *ÇK*'lerinden bir iye *ÇK*'ye bağlanmalıdırlar. Diğer bir anlatımla, kök sözcüğün son *ÇK*'si dışındaki tüm son *ÇK*'lerden bir bağlılık oku çıkmaktadır. Ancak bağlılık oku girmeyen (herhangi bir *ÇK*'nin iyesi olmayan) *ÇK*'ler bulunabilir. Bunlar oluşan bağlılık ağacının yaprakları veya ara düğümlerin bazı *ÇK*'leri olabilir. Örnek: "akıllısı" sözcüğünün ilk *ÇK*'si ve "bu" sözcüğü.



Şekil 2.2: Örnek Tümce

Det: Belirleyen, Noun: İsim, Adj: Sıfat, Adv: Belirteç, Verb: Eylem, A3sg: 3. tekil kişi eki, A3pl: 3. çoğul kişi eki, Pnon: iyelik eki yok, Loc: -de hali, Gen: sahip olma hali, Nom: yalın hali, With: -li isimden sıfat türetme eki, Zero: ek almadan türetme, Pos: olumlu, Prespart: Şimdiki zaman ortacı, Pres: Şimdiki zaman, Cop: Koşaç

Türkçe'de tümceler içerisindeki sözcük dizilişleri büyük çoğunlukla Özne-Nesne-Yüklem veya Nesne-Özne-Yüklem kalıplarına uymasına rağmen, öğelerin anlatılmak istenen içeriğe ve vurguya bağlı olarak tümce içerisinde serbestçe yer değiştirebildikleri görülmektedir. Bunun nedeni, Türkçe'de tümcenin öğelerinin, tümce içindeki konumlarıyla değil; aldıkları eklerle (özne ve belirtisiz nesne ek almaz) belirlenmesidir. İngilizce "*Wash me*" tümcesinde sözcüklerin yeri değiştirildiğinde (*Me wash*) tümcenin anlamının tamamen kaybolmasına rağmen, aynı tümcenin Türkçe karşılığında ("*Beni yıka*" ve "*Yıka beni*") hiçbir anlam kaybı olmamaktadır. Bağlılık yapısına bakıldığında, her zaman olmasa da ağırlıklı olarak sağa bağımlı türde oldukları görülmektedir. Bunun nedeni ise Türkçe'de kurallı tümcelerde yüklemin tümce sonunda bulunmasıdır. Şekil 2.3'de kurallı bir tümcede öğelerin tümce içerinde serbestçe yer değiştirmelerine bir örnek gösterilmektedir. Görüldüğü gibi tümce devrik hale getirilmediği sürece bağlılıklar sağa bağımlı olmaya devam etmektedirler. Ancak tümce içerisinde serbestçe yer değiştiren parçaların sözcükler değil öğeler olduğunu belirtmekte fayda vardır. Örnekte de görülebileceği gibi "bu kadın" tamlamasında "bu" işaret sıfatı her zaman "kadın" sözcüğünden bir önceki konumda bulunmalıdır.



Şekil 2.3: Öğelerin Serbestçe Yer Değiştirmesi

2.3 Derlem

Bu çalışmada sınama verisi olarak bağlılık grameri yapısına uygun olarak hazırlanmış ODTÜ-Sabancı Ağaç Yapılı Derlemi (Oflazer ve diğ., 2003; Atalay ve diğ., 2003) kullanılmıştır. Daha önce de belirtildiği gibi, bu ağaç yapılı derlem tez içerisinde kısaca "*derlem*" olarak anılacaktır. Derlem sekiz farklı türdeki yazılardan derlenmiş

5635 tümce içermektedir. Bu tümcelerdeki sözcükler, öncelikle biçimbilimsel çözümleyiciden geçirilmiş ve daha sonra farklı biçimbilimsel çözümler arasında belirsizlik giderme işlemleri insan tarafından yapılmıştır. Derlemde bağlılıklar, önceki bölümde anlatıldığı gibi *ÇK*'ler arasında kurulmuştur. Bağlılıklar, uydu sözcüğün son *ÇK*'sinden başlayarak iye sözcüğün herhangi bir *ÇK*'sinde sonlanmaktadır. Derlem, noktalama işaretleri hariç 43572 adet sözcük içermektedir[3].

Türkçe genelde ve özellikle yazım dilinde sağa bağımlı bir dil olarak nitelendirilebilir. Nitekim kullandığımız derlemdeki bağlılıkların %95'i bu tür bağlılıklardan oluşmaktadır. Türkçe'de bir sözcüğe ait biçimbilimsel özellikler büyük çoğunlukla o sözcüğün içerisinde bir çekim eki olarak yer almaktadırlar. Ancak bazı ekler ("de, mi, ki"[4]) sözcüğe ait biçimbilimsel özellik taşımalarına rağmen sözcükten sonra ve sözcükten ayrı olarak yazılırlar ve derlemde kendilerinden önce gelen iye sözcüğe bağlanarak sola bağımlı türde bağlılıklar yaratırlar. Bu bağlılıklar bir önişlemci yazılarak rahatlıkla bulunabilir. Bu işlemin sonucunda sağa bağımlı kuralına uymayan bağlılıkların oranı %5'den %3'e inmektedir. Herhangi bir iye sözcüğe bağlanmamış noktalama işaretleri gözardı edildiğinde, Türkçe derlemdeki bağlılıkların %2,5'nin başka bir bağlılığı kestiği ve bunlardan kaynaklanarak tümcelerin %7,2'sinin kesişen bağlılıklardan oluştuğu saptanmaktadır.[5]

Şekil 2.4'de, Şekil 2.2'deki örnek tümcenin Türkçe derlemde kullanılan XML biçiminde gösterimi verilmektedir. Bu gösterimde tümceler <S><\S>, sözcükler ise <W><\W> etiketleri arasında gösterilir. Her sözcük IX, LEM, MORPH, IG ve REL olmak üzere beş farklı etiket barındırır. Bu etiketlerin anlamları şöyledir:

1. IX: Sözcüğün tümce içerisindeki sıra numarası,
2. LEM: Sözcüğün Türkçe bir sözlükte nasıl geçeceği[6],
3. MORPH: Biçimbilimsel temsil[6],

---

[3]Noktalama işaretleri katıldığında bu sayı 58K mertebesindedir.

[4]"de, mi, ki" ekleri ve bu eklerin farklı görünüm şekilleri; "de/da", "mi" soru ekinin kişi ve zaman ekleri almış tüm çeşitleri ve "ki"

[5]Ancak bu cümleler incelendiğinde, oluşan kesişmelerin büyük çoğunlukla derlemdeki birden çok *ÇK* içeren eylemlere doğru yapılan bağlılık hatalarından kaynaklandığı görülmektedir.

[6]Derlemin şu anki sürümünde bu alan boştur.

4. IG: Biçimbilimsel çözümleme (gövdesi, *ÇK* yapısı ve biçimbilimsel özellikleri),
5. REL: Sahip *ÇK*'nin sıra numarası ve bağlılık türü.

"REL" etiketi, iye *ÇK*'nin birbirinden virgül ile ayrılan iki tam sayıdan oluşan sıra numarası ve bir bağlılık türü ile ifade edilir. Sayılardan birincisi sahip sözcüğün sıra numarası, ikincisi ise bağlanılacak olan sahip *ÇK*'nin sahip sözcük içerisindeki sıra numarasını belirtmektedir. Örneğin IX="4" nolu sözcüğün REL="[5,2,(MODIFIER)]" etiketi bu sözcüğün 5 numaralı sözcüğün ikinci *ÇK*'sine bağlanacağı anlamına gelmektedir. Örnekten de görüldüğü üzere, derlemde tümcelerin bağlılık ağacının kökü olarak genelde en sondaki noktalama işareti alınmıştır. Tümcenin ana eylemi bu noktalama işaretine "SENTENCE" bağlılık türü ile bağlanır.

Derlem, 23 adet[7] farklı türde bağlılık içermektedir. Tablo B.3'de listesi ve kısa açıklamaları verilen bu bağlılıklar ile ilgili ayrıntılı açıklamalar ve örnekler derlem kullanma kılavuzundan (Say, 2004) incelenebilir.

2.4 Derlem Üzerindeki İyileştirmeler

Biçimbilimsel belirsizlik giderimi ve tümce bağlılık çözümlemesi, insanlar tarafından yapılmış olan ODTÜ-Sabancı ağaç yapılı derlemi, diğer birçok derlem gibi hatalar içermektedir. Birden çok kişi tarafından hazırlanan bu derlemlerin hatasız hale getirilmesi de, en az hazırlanması kadar emek yoğun ve uzun bir iş olabilmektedir. Türkçe ağaç yapılı derlemi ilk olarak 2004 yılında kullanıma sunulmuş (Say, 2004) ve diğer araştırmacılar (Eryiğit ve Oflazer, 2006; Buchholz ve Marsi, 2006; Çakıcı, 2005) derlem üzerinde incelemelere bu tarihten sonra başlamışlardır. Bu tez çalışmasında bu derlemin iyileştirilmesi için yoğun çalışmalar yapılmıştır. Bu bölümde, derlemin ilk halinden Conll-X ortak çalışmasında kullanılan sürümüne kadar geçen evrede yapılan iyileştirme çalışmalarının kısa bir özeti verilecektir. Derlem ile ilgili düzeltmeler ve bulunan hatalar birçok araştırmacı tarafından tarafımıza

[7]Derlem üzerinde yapılan iyileştirmelerden önce bu sayı 24'tür. Derlemin ilk sürümünde bulunan "R.SENTENCE" bağlılık türü (bu türde sadece 4 adet bağlılık bulunmakta idi) değiştirilmiş ve "SENTENCE" bağlılık türüne dönüştürülmüştür.

```
<?xml version="1.0" encoding="windows-1254" ?>
<Set sentences="1">
    <S No="1">
      <W IX="1" LEM="" MORPH="" IG="[(1,"bu+Det")]"
         REL="[2,1,(DETERMINER)]">Bu</W>
      <W IX="2" LEM="" MORPH=""
         IG="[(1,"okul+Noun+A3sg+Pnon+Loc")(2,"Adj+Rel")]"
         REL="[3,1,(MODIFIER)]">okuldaki</W>
      <W IX="3" LEM="" MORPH=""
         IG="[(1,"öğrenci+Noun+A3pl+Pnon+Gen")]"
         REL="[5,3,(POSSESSOR)]">öğrencilerin</W>
      <W IX="4" LEM="" MORPH="" IG="[(1,"en+Adv")]"
         REL="[5,2,(MODIFIER)]">en</W>
      <W IX="5" LEM="" MORPH=""
         IG="[(1,"akıl+Noun+A3sg+Pnon+Nom")(2,"Adj+With")
         (3,"Noun+Zero+A3sg+P3sg+Nom")]"
         REL="[9,2,(SUBJECT)]">akıllısı</W>
      <W IX="6" LEM="" MORPH=""
         IG="[(1,"şura+Noun+A3sg+Pnon+Loc")]"
         REL="[7,1,(LOCATIVE.ADJUNCT)]">şurada</W>
      <W IX="7" LEM="" MORPH=""
         IG="[(1,"dur+Verb+Pos")(2,"Adj+PresPart")]"
         REL="[9,1,(MODIFIER)]">duran</W>
      <W IX="8" LEM="" MORPH="" IG="[(1,"küçük+Adj")]"
         REL="[9,1,(MODIFIER)]">küçük</W>
      <W IX="9" LEM="" MORPH=""
         IG="[(1,"kız+Noun+A3sg+Pnon+Nom")
         (2,"Verb+Zero+Pres+Cop+A3sg")]"
         REL="[10,1,(SENTENCE)]">kızdır</W>
      <W IX="10" LEM="" MORPH="" IG="[(1,".+Punc")]"
         REL="[,( )]">.</W>
    </S>
</Set>
```

Şekil 2.4: Türkçe Derlem Veri Biçimi (XML)

halen bildirilmektedir. Ancak düzeltmelerin tek tek kontrol edilerek ve genel bütünlüğü bozmaması açısından derlemi oluşturan araştırmacılar ile ortak kararlar doğrultusunda yapılması gerektiğinden, bu işlem oldukça zahmetli ve zaman alıcıdır. Buna ek olarak, derlem şu anda birçok araştırmacı tarafından kullanımda olduğundan, düzeltmelerin sürekli değil, toplu halde ve yeni sürümler oluşturacak şekilde yapılması gerekmektedir.

Derlemdeki hatalar basit ve karmaşık olmak üzere iki ana sınıfa ayrılabilirler. Basit hatalar genelde küçük program parçacıkları ile tespit edilebilen ve düzeltilmeleri diğerlerine oranla daha kolay olan hatalardır. Bunlara örnek olarak, biçimbilimsel etiketlerde veya bağlılık türü etiketlerinde yapılan yazım hataları, büyük küçük

harf hataları, var olmayan sıra numarasındaki bir sözcüğe bağlanma hataları, dairesel bağlılıklara yol açan hatalar[8] gösterilebilir. Karmaşık hatalar ise ancak hata incelemesi sırasında tümcelerin tek tek incelenmesi sonucunda ortaya çıkan ve düzeltilebilmeleri için tümcenin tümünün yeniden incelenip ayrıştırılmasını gerektiren hatalardır. Bunlara örnek olarak, sözcüklerin belirsizlik giderme işlemi sırasında yanlış biçimbilimsel çözümlemenin seçilmiş olması, aynı türde bağlılıklara farklı tümcelerde farklı bağlılık türleri atanması (derlemi ayrıştıran kişiler arasında standardın sağlanamaması) gösterilebilir. Bu tür hataların düzeltilebilmesi için derlemin hazırlanış mantığını çok iyi kavramış uzmanlara gereksinim vardır.

Derlemde şu ana kadar yapılmış olan düzeltmeler aşağıdaki başlıklar altında toplanabilir. Yapılan değişikliklerin bazıları ile ilgili ek açıklamalar Ek C'de verilmektedir.

- Bağlılık türü etiketlerinde yapılan imla hatalarının düzeltilmesi,
- Biçimbilimsel etiketlerde yapılan imla hatalarının düzeltilmesi,
- Var olmayan sıra numaralarına yapılan bağlılıkların düzeltilmesi,
- Dairesel bağlılıklara neden olan hataların düzeltilmesi,
- Biçimbilimsel belirsizlik giderimi yanlış yapılmış sözcüklerin ve bunlara bağlı bağlılıkların düzeltilmesi[9],
- Yanlış bağlılıkların düzeltilmesi[9],
- "bir" sözcüğünün biçimbilimsel çözümlemesine ve bağlılık türüne dair uyumluluğun sağlanması,
- "var" ve "yok" sözcüğünün biçimbilimsel belirsizlik giderimindeki uyumsuzluğun düzeltilmesi,
- Noktalama işaretleri ile ilgili hataların bir kısmının düzeltilmesi,
- R.SENTENCE bağlılık türünün kaldırılması.

[8] Bu tür hatalar bulunmaları kolay ancak düzeltilmeleri zor hatalardır. Bu nedenle karmaşık hata olarak da nitelendirilebilirler.

[9] Bu sınıftaki hataların tümü değil ancak hata incelemeleri sırasında rastlananları düzeltilebilmiştir.

Derlemde sayıca çok fazla olan noktalama işaretlerinin nasıl bağlandığı ve bu bağlılıkların ayrıştırma sırasında nasıl işleme alındığı ayrıştırma başarımını önemli ölçüde etkileyen bir konudur. Ancak, diğer birçok derlemde olduğu gibi Türkçe derlemde de noktalama işaretlerinin bağlanmasında büyük ölçüde uyumsuzluklar görülmektedir. Çalışma kapsamında bunların bir kısmı giderilmeye çalışılmıştır. Ancak bu konuda yapılan hataların tümünün giderilebilmesi için tüm derlem gözden geçirilmelidir.

Derlem, 2006 yılında Bölüm 4.8'de ayrıntıları verilen Conll-X ortak çalıştayında kullanılmak üzere ayrı bir biçime dönüştürülmüştür. Bu dönüşüm sırasında yeni bulunan hatalar düzeltilirken, bunlara ek olarak ortak çalışma sırasında kullanılan tüm derlemler arasında uyumluluğu sağlamak üzere (noktalama işaretleri ile ilgili olarak), derlem bağlılık yapısında önemli bir değişiklik yapılmıştır. Derlem içi uyumlulukta sorunlara yol açan bu değişimin ayrıntıları Bölüm 4.8'de anlatılmaktadır. Ancak ileriki dönemlerde derlemin Conll-X versiyonu üzerinde derlem içi uyumluluğu yeniden sağlamak üzere güncelleme yapılması gerekecektir.

Tüm bunlara ek olarak, Türkçe ağaç yapılı derlem, diğer diller için hazırlanmış derlemlere oranla halen küçük boyutlu derlem sayılmaktadır. Bu nedenle, derlem boyutunun büyütülmesi ve çeşitliliğin arttırılması gerekmektedir. Bu tür çalışmalar uzun yıllar gerektirmekte ve konusunda uzman olan kişilerce ortak projeler[10] kapsamında yapılmaları gerekmektedir.

Şu an için, Türkçe derlemin yukarıdaki düzeltmeler yapılmış halde iki yeni sürümü bulunmaktadır. Bunlardan birincisi derlemin ilk sürümünün yukarıdaki hatalardan ayıklanmış hali, ikincisi ise noktalama işaretleri ile ilgili dönüşüm uygulanmış Conll-X sürümüdür. Tez içerisinde, Bölüm 4.8'a kadar olan bölümde birinci sürüm kullanılmıştır.

[10]Derlem geliştirilmesi üzerinde çok çalışılan ve tartışılan konulardan biridir. Konuyla ilgili konferanslar düzenlenmektedir. Bunlardan önemli iki tanesi "International Treebanks and Linguistic Theories Conference" ve "International Conference on Language Resources and Evaluation"dir.

3. TÜRKÇE'NİN BAĞLILIK AYRIŞTIRMASI

Bu bölümde, bu tez çalışmasıyla ortaya konan yenilikler ve katkılar tanıtılacaktır. Giriş bölümünde de değinildiği gibi veri güdümlü bir ayrıştırıcı üç bileşenden oluşmaktadır: ayrıştırma algoritması, ayrıştırma modeli ve öğrenme modeli. Bu bölümde, yapılan incelemeler sonucunda, farklı ayrıştırıcıların Türkçe'ye uygun görülen bileşenleri bir araya getirilerek, iki farklı veri güdümlü ayrıştırıcı oluşturulmuştur. Bunlar, koşullu olasılık tabanlı modele dayalı istatistiksel bir ayrıştırıcı (Olasılık Tabanlı Ayrıştırıcı olarak anılacaktır) ve ayırdedici öğrenmeye dayalı sınıflandırıcı tabanlı bir ayrıştırıcıdır (Sınıflandırıcı Tabanlı Ayrıştırıcı olarak anılacaktır). Oluşturulan ayrıştırıcıların yapısı ve bileşenleri ile ilgili ayrıntılı bilgi ilgili kısımlarda verilecektir. Bu ayrıştırıcılara ek olarak, başarımlara bir alt sınır oluşturmak üzere üç farklı temel ayrıştırıcı geliştirilmiştir. Bunlar iki basit ayrıştırıcı ve bir kural tabanlı ayrıştırıcıdır. Bu bölümde, oluşturulan ayrıştırıcılar kullanılarak Türkçe'ye özgü geliştirdiğimiz farklı tasarım modellerinin veri güdümlü ayrıştırıcılardaki etkisi incelenmiştir. Yapılan çalışmalar, biçimbilimsel yapının, biçimbilimsel olarak çok zengin olan Türkçe'nin tümce içi ilişkilerini bulmada önemli etkisi olduğunu göstermektedir. Bu bölümde sağlanan bilimsel katkı, Türk dilinin bağlılık çözümlemesinde en yüksek başarımı elde etmek için gerçeklediğimiz modelleme biçimidir. Bu modelleme, ayrıştırmada ana birim olarak sözcükler yerine çekim kümelerinin kullanılmasına dayalıdır. Ayrıca, biçimbilimsel özelliklerin kullanılmasının Türkçe'nin ayrıştırılmasında vazgeçilemez bir yere sahip olduğu gösterilmiştir. Aşağıdaki bölümlerde, geliştirilen

- Temel ayrıştırıcılar,
- Olasılık tabanlı ayrıştırıcı,
- Sınıflandırıcı tabanlı ayrıştırıcı

ve farklı tasarım modelleri tanıtılmaktadır.

3.1 Temel Ayrıştırıcılar

Geliştirdiğimiz modellerin başarımlarına bir temel oluşturmak üzere üç temel ayrıştırıcı geliştirilmiştir. Geliştirilecek herhangi bir modelin başarılı sayılabilmesi için en azından bu temel ayrıştırıcıların başarımını aşması gerekmektedir.

Temel Ayrıştırıcı 1: Bu ayrıştırıcıda, her sözcük (son *ÇK*'sinden) sağındaki sözcüğün ilk *ÇK*'sine bağlanır.

Temel Ayrıştırıcı 2: Bu ayrıştırıcıda, her sözcük (son *ÇK*'sinden) sağındaki sözcüğün son *ÇK*'sine bağlanır. Tek *ÇK*'den oluşan iye sözcükler için Temel Ayrıştırıcı 1 ve 2 aynı şekilde davranırlar. Bu iki ayrıştırıcıda, tümcenin en sonunda yer alan noktalama işareti, oluşacak ayrıştırma ağacının kökü olarak kabul edilir ve hiçbir yere bağlanmaz.

Temel Ayrıştırıcı 3: Kural tabanlı gerekirci bir ayrıştırıcıdır.

Veri güdümlü yaklaşımları tanıtırken kullanılan yönteme benzer bir tanıtım yapmak istenirse, kural tabanlı ayrıştırıcı şu üç yöntemin birleşimi olarak gösterilebilir:

1. Gerekirci ayrıştırma algoritması (Oflazer, 2003; Nivre, 2003),
2. Kural tabanlı ayrıştırma modeli (Eryiğit ve diğ., 2006a),
3. İnsan tarafından oluşturulmuş kurallar.

Bu ayrıştırıcıda, ayrıştırma algoritması olarak Nivre (2006b)'nin gerekirci algoritması sadece sağa bağımlı türde bağlılıkları işlemek üzere geliştirilerek kullanılmıştır. Ayrıştırma birimi olarak sözcükler kullanılmaktadır. Bu algoritma basit *ötele indirge* algoritmasının bir çeşididir. Bu tür algoritmalar genelde tümceyi soldan sağa doğru, iki farklı veri yapısından faydalanarak ayrıştırırlar:

- İşlenmekte olan sözcüklerin tutulduğu yığın yapısı
- İşlenmek üzere bekleyen sözcüklerin tutulduğu kuyruk yapısı

Sadece sağa bağımlı türde bağlılıkları bulmak üzere geliştirilen algoritmanın işleyişi aşağıdaki gibidir (i = yığının en üstünde duran sözcüğün sıra numarası, j = kuyrukta bekleyen sıradaki sözcüğün sıra numarası):

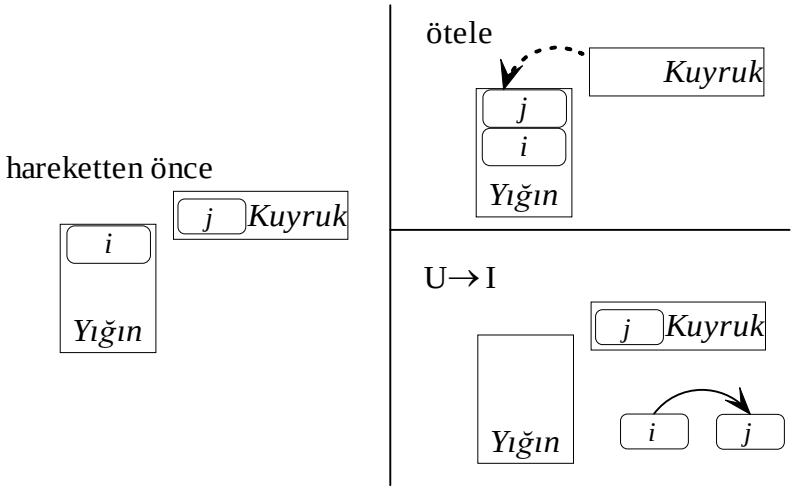
```
Kuyrukta bekleyen sözcük olduğu sürece tekrarla{
     eğer Yığın boş ise
        Ötele(Yığın)
     değil ise
       hareket = Ayrıştırma_Modeli(i,j)
       eğer hareket == Ötele ise
         Ötele(Yığın)
       eğer hareket == U → I ise
         Bağlılık_Kur(i → j)
         Çek(Yığın)
}
```

Algoritma sadece kesişmeyen bağlılıkları bulmaya yöneliktir. Ayrıştırıcı her adımında iki farklı hareketten (Ötele, $U \rightarrow I$) birini gerçekleştirir (Hareketler sonrasında yığının ve kuyruğun durumu Şekil 3.1'de gösterilmiştir). Ayrıştırıcının bir sonraki hareketinin ne olacağına, ayrıştırma modeli "i" ve "j" sıra numaralı elemanların görünüm bilgisi içermeyen özelliklerine bakarak, önceden insan tarafından hazırlanmış kuralları[1] kullanarak karar verir. "Ötele"me işleminde kuyrukta bekleyen eleman yığına itilir. Bu işlem yığının boş olduğu durumlarda veya "i" ve "j" sıra numaralı elemanlar arasında herhangi bir bağlılık kurulamadığı durumlarda gerçekleşir. "$U \rightarrow I$" işlemi "i" sıra numaralı eleman ile tümce içerisinde sağ tarafında yer alan "j" numaralı eleman arasında uydu-iye ilişkisi olduğu durumlarda gerçekleşir. Ayrıştırma sonunda yığında noktalama işareti olmayan ve bağlantısı yapılmamış bir sözcük kalırsa, bu sözcük tümcenin en son sözcüğünün son *ÇK*'sine bağlanır.

3.2 Olasılık Tabanlı Ayrıştırıcı

Bu bölümde Türkçe'nin veri güdümlü bağlılık çözümlemesi ile ilgili yapılan ilk incelemeler sunulmaktadır. İlk olarak, geliştirilen olasılık tabanlı ayrıştırıcının

[1] İleriki bölümlerde, derlemin tümü üzerinde (sola bağımlı türde bağlılıkları da kapsayacak şekilde) çalışan daha gelişmiş bir kural tabanlı ayrıştırıcı tanıtılacaktır. Bu ayrıştırıcının kullandığı kurallar burada kullanılan kuralları da kapsayacak şekilde Ek A'da verilmektedir.



Şekil 3.1: Ayrıştırma Algoritması

mimarisi ve ayrıştırma birimlerinin gösterimleri tanıtılmaktadır. Daha sonra oluşturulan modeller verilerek elde edilen sonuçlar yorumlanmaktadır.

3.2.1 Mimari

Veri güdümlü bir bağlılık ayrıştırıcısı olan olasılık tabanlı ayrıştırıcı üç farklı teknik birleştirilerek oluşturulmuştur:

1. Bağlılık grafiğini oluşturmak için kullanılan dinamik bir ayrıştırma algoritması (Uchimoto ve diğ., 1999; Sekine ve diğ., 2000)
2. Çözümlemeleri değerlendirmek üzere kullanılan *koşullu olasılık tabanlı* ayrıştırma modeli (Collins, 1996)
3. Olasılık modeli ile ilgili çıkarım yapmak üzere kullanılan *en büyük olabilirlik kestirimi*[⋈] (Collins, 1996; Chung ve Rim, 2004)

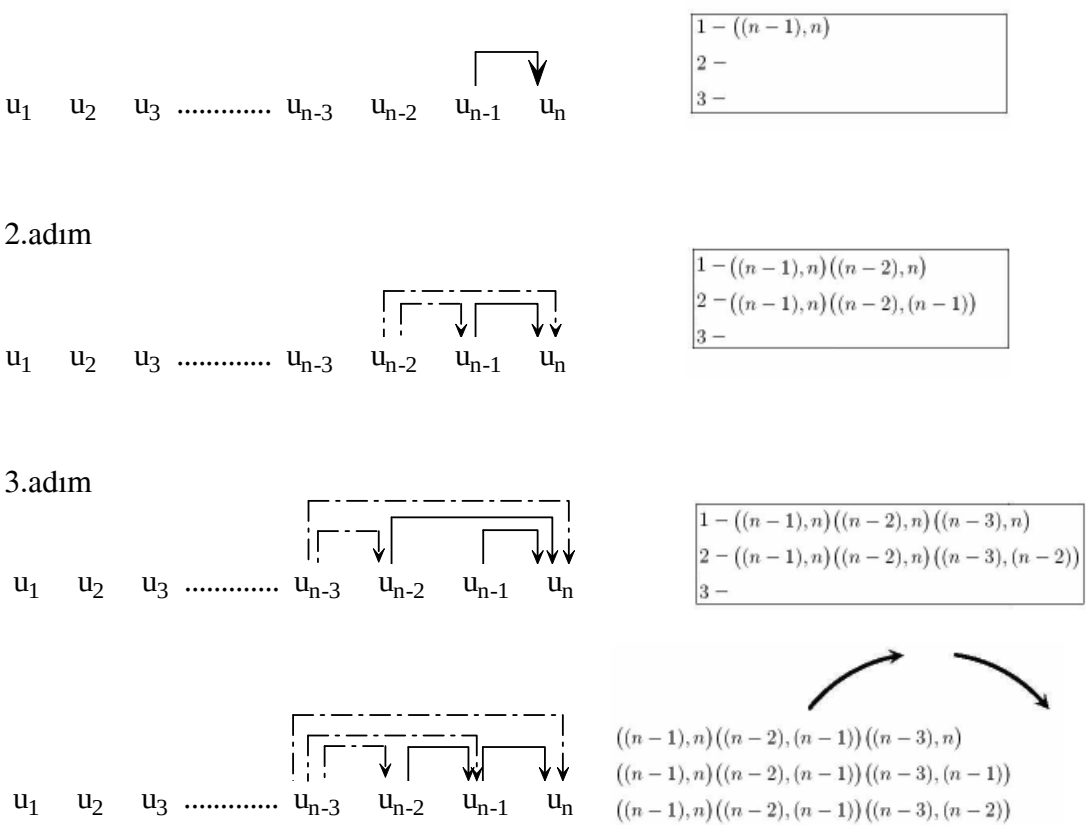
Olasılık tabanlı modelin amacı olası her bağlılığa eğitim kümesi içerisinde yeralan benzer bağlılıkların görülme sıklığından yola çıkarak bir olasılık değeri atamaktır. Bir diğer deyişle, birimler arasındaki ikili bağlılık olasılıklarını hesaplamaktır. Bundan sonra, ayrıştırma algoritmasının amacı ise bu olasılıkları kullanarak arama uzayı içerisindeki en olası bağlılık ağacını (T^*) bulmaktır. Denklem 3.1 koşullu olasılık tabanlı ayrıştırma yaklaşımının denklemini vermektedir. Bu denklemde S, n adet

birimin sıralanmasıyla oluşan tümcedir. $u_{H(i)}$, uydu u_i biriminin bağlandığı iye birimi ifade etmektedir. T ise $bag(u_i, u_{H(i)})$ uydu-iye bağlılık ilişkilerini içeren olası tüm bağlılık ağaçlarını ifade etmektedir.

$$\begin{aligned} T^* &= \underset{T}{\text{argmax}}\ P(T|S) \\ &= \underset{T}{\text{argmax}} \prod_{i=1}^{n-1} P(bag\,(u_i, u_{\mathcal{H}(i)})\,|\,S) \end{aligned} \tag{3.1}$$

Türkçe derlemdeki bağlılıkların %95'inin sağa bağımlı türde bağlılıklar olmasından yola çıkarak, bu ayrıştırıcıda ayrıştırma algoritması olarak Sekine ve diğ. (2000)'nin geriye doğru demetli arama algoritması seçilmiştir. Algoritma Türkçe'nin biçimbilimsel yapısına uygun hale getirilerek gerçeklenmiştir. İlk olarak sağa bağımlı türde bir dil olan Japonca için geliştirilen bu algoritma, tümceyi sondan başlayarak başa doğru ayrıştırır. Algoritma her adımında üzerinde bulunduğu birimi, tümce içerisinde bu birimin sağ tarafında yer alan birimlerden birine bağlamaya çalışır. Bu işlem sırasında, ayrıştırması kısmen yapılmış tümce üzerinde, en olası d adet kısmi ayrıştırmayı bir demet içerisinde tutar. Her adımda, ortaya çıkan yeni ayrıştırmalar içerisinde demettekilerden daha yüksek olasılıklı olanları, demette yer alan daha düşük olasılıklı ayrıştırmaların en az olası olanının yerine geçirir. Algoritma, kesişmeyen bağlılık ilkesini benimsemektedir. Bu nedenle, uydu birimlerin bağlanacakları iye birimler seçilirken yeni kurulan bağlılıkların daha önceden oluşmuş kısmi bağlılıklarla çakışmamasına dikkat edilir.

Şekil 3.2'de n adet birimden oluşan bir S dizisi için ayrıştırmanın ilk üç adımı gösterilmiştir. Örnekte demet boyutu 3 olarak alınmıştır (Ayrıştırıcı her adımda üç adet en yüksek olasılıklı yapının kaydını tutacaktır) ve demet içindeki kayıtlar dikdörtgen içerisinde gösterilmiştir. İlk adımda ayrıştırıcı $(n-1)$. sıradaki birimi doğrudan n. birime bağlar. Sağ tarafta yer alan başka birim olmadığından dolayı ayrıştırıcının yapabileceği tek hareket budur. Demet boş olduğundan ötürü, bu yapı doğrudan demete kaydedilir. İkinci adımda ise, ayrıştırıcı birinci adımda oluşmuş yapıdan yola çıkarak iki farklı işlem yapabilir. Bunlar $(n-2)$. birimi n. veya $(n-1)$. birime bağlamaktır (Şekilde noktalı çizgiler ile belirtilmiştir). Oluşan bu iki yeni yapı, bağlılık olasılıkları hesaplanarak demete girerler. Üçüncü adımda ayrıştırıcının elinde kısmi



Şekil 3.2: Geriye Doğru Demetli Arama Örneği

olarak oluşmuş iki farklı yapı yer almaktadır. Ayrıştırıcı her ikisi için de olası yeni bağlılıkları oluşturmalı ve en yüksek olasılıklı olanlarını demette tutmalıdır. Bir önceki adımda oluşmuş birinci kısmi yapı $\big((n-1),n\big)\big((n-2),n\big)$ ele alındığında, $(n-3)$. birimin bağlanabileceği sadece iki olası birim vardır. Bunlar n. ve $(n-2)$. birimlerdir (Şekilde, 3. adımın ilk satırında noktalı çizgiler ile belirtilmişlerdir). $(n-1)$. birim, kesişmeyen bağlılık ilkesine aykırı düşeceğinden[2] iye sözcük olma adayları arasına alınmaz. Yeni oluşan bu iki yapı, $\big((n-1),n\big)\big((n-2),n\big)\big((n-3),n\big)$ ve $\big((n-1),n\big)\big((n-2),n\big)\big((n-3),(n-2)\big)$, demete kaydedilir. Sıra ikinci adımda oluşan ikinci kısmi yapıyı, $(n-1,n)(n-2,n-1)$, işlemektedir. Bu durumda $(n-3)$. birim sağ tarafında yer alan her üç birime de bağlanabilir (Şekilde, 3. adım ikinci satır). Yeni oluşan bu üç yeni bağlılık yapısının demete girebilmesi için, eğer varsa demette daha düşük olasılıklı olan yapıların çıkarılması gerekmektedir.

[2] $\big((n-1),n\big)\big((n-2),n\big)$ bağlılığı ile çakışıklık yaratması nedeniyle.

Geriye doğru demetli arama algoritması kısmi olarak oluşturduğu yapıların olasılıklarını hesaplamak için Denklem 3.1'deki eşitliği kullanır. Bu denklemdeki $P(bag\,(u_i, u_{\mathcal{H}(i)}) \mid S)$ ile belirtilen ikili bağlılık olasılıklarının hesaplanması ise Denklem 3.2'de gösterilmektedir. Bu yaklaşım, Collins (1996)'in olasılık tabanlı modelinin değiştirilmiş bir çeşidi olan Chung ve Rim (2004)'in yaklaşımıdır.[3] Yaklaşım, birbirlerine bağlanan birimler arasındaki uzaklığın, bağlılık olasılığının hesaplanmasında önemli bir etkisi olduğu fikrine dayanmaktadır.

$$P(bag\,(u_i, u_{\mathcal{H}(i)}) \mid S) \approx P(ilk(u_i, u_{\mathcal{H}(i)}) \mid \Phi_i\,, \Phi_{\mathcal{H}(i)}) \cdot P(u_i' \text{nin } uzk(i, H(i)) \text{ uzağa bağlanması } \mid \Phi_i) \tag{3.2}$$

Denklem 3.2'de $P(bag\,(u_i, u_{\mathcal{H}(i)}) \mid S)$ uydu u_i'nin bir $u_{\mathcal{H}(i)}$ iye birimine bağlanma olasılığıdır ve iki diğer olasılığın hesaplanması ile bulunur:

- $P(ilk(u_i, u_{\mathcal{H}(i)}) \mid \Phi_i\,, \Phi_{\mathcal{H}(i)})$: Benzer bir bağlılığın benzer bir bağlam içerisinde görülme olasılığıdır. Φ_i, u_i uydu birim etrafındaki bağlam bilgisini ve $\Phi_{\mathcal{H}(i)}$, $u_{H(i)}$ iye birim etrafındaki bağlam bilgisini belirtmektedir.
- $P(u_i' \text{nin } uzk(i, H(i)) \text{ uzağa bağlanması } \mid \Phi_i)$: Uydu birimin benzer bağlamda benzer uzaklıktaki herhangi bir iye birime bağlanma olasılığıdır. Uydu ve iye birim arasındaki uzaklık bir uzaklık fonksiyonu kullanılarak hesaplanır.

Bu tür modellerin en büyük sorunlarından biri, bir derlem üzerinde eğitilmeye gereksinim duymalarıdır. Eğitim verisi ne kadar büyük olursa, sınama verisinde karşılaşılan verilerin daha önceden görülme olasılığı da o kadar artacaktır. Her koşulda, dilin tümünü örnekleyen bir eğitim verisi oluşturmak çok zordur. Bu nedenle, bu tür modellerde seyrek veri sorunuyla karşılaşılmaktadır. Seyrek veri sorunu gerekli olasılıkların hesaplanması için yeterli veriye sahip olunmaması durumudur. Özellikle Türkçe derlem gibi küçük boyutlu derlemlerde bu soruna daha da sık rastlanmaktadır. Seyrek veri sorununun aşılabilmesi için farklı *düzleştirme algoritmaları*[⋊] uygulanmaktadır.

[3] Chung ve Rim (2004)'in Korece için oluşturduğu bu yaklaşımın, Türkçe'nin çözümlemesinde de Collins (1996)'in modeline göre daha yüksek başarım verdiği gözlemlenmiştir.

Bu ayrıştırıcıda, düzleştirme algoritması olarak Collins (1996)'in çalışmasında kullanılan *düşürerek düzleştirme*$^{\times}$ algoritmasına benzer bir algoritma kullanılmıştır. Denklem 3.2, iye ve uydu birimin bağlam bilgilerinin hepsinin birden, bir seferde kaldırılması ile elde edilen olasılık değerleri ile *aradeğerlenerek*$^{\times}$ hesaplanmıştır.[4] Buna göre, asıl yürütmeler sırasında,

- $P(ilk(u_i, u_{\mathcal{H}(i)}) \mid \Phi_i, \Phi_{\mathcal{H}(i)})$ düzleştirilmiş olasılığı derlemden çıkarılmış iki düzleştirilmemiş olasılık aradeğerlenerek hesaplanmıştır: $P_1(ilk(u_i, u_{\mathcal{H}(i)}) \mid \Phi_i, \Phi_{\mathcal{H}(i)})$ ve $P_2(ilk(u_i, u_{\mathcal{H}(i)}))$.
- $P(u_i'\text{nin } uzk(i, H(i))$ uzağa bağlanması $\mid \Phi_i)$ olasılığı da benzer şekilde $P_1(u_i'\text{nin } uzk(i, H(i))$ uzağa bağlanması $\mid \Phi_i)$ ve $P_2(u_i'\text{nin } uzk(i, H(i))$ uzağa bağlanması $)$ olasılıklarının aradeğerlenmesi ile hesaplanmıştır.

Eğer bu aradeğerlendirmeden sonra bile olasılık değeri sıfır çıkıyorsa o zaman olasılık değeri olarak sıfıra yakın çok küçük bir değer atanmıştır.

Olasılık değerleri eğitim verisi üzerinde en büyük olabilirlik kestirimi yapılarak hesaplanır. Yukarıdaki düzleştirilmemiş olasılıklar, derlemde benzer bağlılıkların görülme sıklığının *düzgelenmiş*$^{\times}$ değerleri hesaplanarak bulunur. Örneğin $P(ilk(u_i, u_j))$ olasılığı şu şekilde hesaplanır:
a ve b gösterimlerine sahip iki birimin birbirlerine uydu-iye ilişkisi ile bağlanma sıklığı $F(a, b)$ ile gösterilirse, $F(a, b)$ Denklem 3.3'de gösterildiği gibi a ve b'nin birbirlerine bağlanma (R) sayısının, a ve b'nin aynı tümce içerisinde görülme sayısına[5] bölünmesi ile kestirilir.

$$F(a, b) = \frac{C(R, a, b)}{C(a, b)} \tag{3.3}$$

[4] Araştırmalar sırasında, bağlam bilgisini teker teker azaltmak veya çekim özelliklerini azaltmak gibi birçok farklı düşürerek düzleştirme modeli denenmiştir. Deneyler sonucunda, burada tanıtılan modelin en yüksek başarımı sağladığı gözlemlenmiştir.

[5] a ve b tek bir tümce içerisinde birden fazla kez görülebilirler. Örneğin S=(a b b) olması durumunda, o tümce için $C(a, b) = C(b, a) = 2$'dir.

Buradan yola çıkarak,

$$\sum_{k=1..m,\, k\neq i} P(ilk(u_i, u_k)) = 1 \tag{3.4}$$

u_i uydu biriminin olası tüm u_k (m olası iye birim sayısı) iye birimlerine bağlanma olasılıkları toplamının bire eşit olmasını (Denklem 3.4) sağlamak üzere, $P(ilk(u_i, u_j))$ aşağıdaki şekilde kestirilir.

$$P(ilk(u_i, u_j)) = \frac{F(u_i, u_j)}{\sum_{k=1..m,\, k\neq i} F(u_i, u_k)} \tag{3.5}$$

Denklem 3.2'de kullanılan uzaklık fonksiyonu uydu birim ve iye birim arasında kalan birim sınırlarının sayısı ile hesaplanır. Yine seyrek veri sorununu azaltmak üzere, belirli bir eşik değerinden (k) yüksek olan uzaklıklar aynı olasılık değerine çekilerek hesaplanmışlardır. Bu yaklaşımla uzaklık fonksiyonu Denklem 3.6'da gösterildiği gibidir.

$$uzk(i, H(i)) = \begin{cases} H(i) - i & \text{eğer } H(i) - i < k \text{ ise} \\ k & \text{eğer } H(i) - i \geq k \text{ ise} \end{cases} \tag{3.6}$$

Yukarıdaki tüm denklemlerde, u_i ayrıştırma sırasında kullanılan i sıra numaralı birimin gösterimidir. Bu bölümün devamında, aşağıdaki iki sorunun cevabını bulmaya yönelik incelemeler yapılmıştır:

- Ayrıştırma birimi nasıl seçilmelidir?
- Birimlerin gösterimi için hangi bilgiler kullanılmalıdır?

Türkçe tümcelerde ilişkileri belirleyen yapıların *ÇK*'ler olması (bknz Bölüm 2) nedeni ile kullanılacak ayrıştırma biriminden bağımsız olarak öncelikle bu yapıların gösteriminde hangi bilgilerin kullanılacağına karar verilmesi gerekmektedir. Bu nedenle öncelikle *ÇK* gösterimleri için seçilen yöntem tanıtılacak ve ayrıştırma biriminin seçimi ile ilgili örnekler bu yöntem kullanılarak anlatılacaktır. Tezin ileriki bölümlerinde, seçilen gösterim yönteminin etkileri ve olası diğer yöntemlerin başarımı nasıl etkileyeceği ayrıntılı olarak incelenecektir.

3.2.2 *ÇK*'lerin Gösterimi

Derlem oluşturulurken sözcüklerin çözümlemesi için kullanılan biçimbilimsel çözümleyici (Oflazer, 1994) oldukça zengin bir çözümleme bilgisi sunmaktadır. Bunlar sözcüğün *ana sözcük sınıfı* (isim, eylem, adıl vb...), bazı ana sınıflar için *alt sözcük sınıfı* (kişi adılı, soru adılı vb...)[6], *görünüm bilgisi*, *gövde bilgisi*, *biçimbilimsel bilgileridir*. Derlem boyunun kısıtlı oluşu ve bu durumun seyrek veri sorununu arttıracağından dolayı, incelemeler ilk olarak görünüm bilgisi eklenmemiş bir gösterim ile başlatılmıştır. Bu gösterimde, her *ÇK* ana sözcük sınıfı ve biçimbilimsel bilgileri ile ifade edilecektir. Kullanılacak bilgiler dinamik bir yöntemle seçilmektedir.

Türkçe tümceler üzerinde yapılan incelemeler sonucunda, derlem tarafından sağlanan biçimbilimsel bilgilerin tümünün bağlılık çözümlemesi için gerekli olmadığı görülmüştür. Bu bilgiler üzerinde yapılacak düzgün bir indirgeme ile hem seyrek veri sorununun azaldığı, hem de başarımda artış sağlandığı gözlemlenmiştir. Ayrıca, birimin ayrıştırma sırasında aldığı göreve göre (iye $u_{\mathcal{H}(i)}$ veya uydu u_i), farklı bilgilerin daha anlamlı olduğu belirlenmiştir. Bu nedenle, birimin görevine göre, ayrıştırma sırasında dinamik olarak belirlenecek bir şeçme yöntemi geliştirilmiştir. Bu yöntemde:

- *ÇK* bir uydu olarak kullanıldığında,
  - – Eğer isim türünden[7] bir *ÇK* ise, o zaman sadece durum imi ile belirtilir.
  - – Diğer türden *ÇK*'ler, sadece ana sözcük sınıfları ile belirtilirler.
- *ÇK* bir iye olarak kullanıldığında,
  - – Eğer isim türünden bir *ÇK* veya zaman ortacı olan bir sıfat[8] *ÇK* ise, o zaman ana sözcük sınıfı ve iyelik uyum imi ile birlikte ifade edilir.
  - – Diğer türden *ÇK*'ler, sadece ana sözcük sınıfları ile belirtilirler.

[6]Derlemde kullanılan ana sözcük sınıfları ve bunlara bağlı alt sözcük sınıfları Tablo B.2'de gösterilmektedir.

[7]Sadece isim türünden *ÇK*'ler durum imine sahiptirler ve bunların uydu olarak görevini belirleyen imler esas olarak durum imleridir.

[8]Şimdiki/Geçmiş/Gelecek zaman ortacına sahip olan sıfatlar, isim türünden *ÇK*'ler dışında iyelik uyum imine sahip tek *ÇK* türleridir.

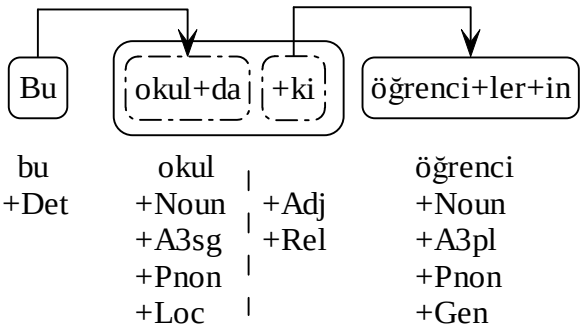
3.2.3 Birim Seçim Modelleri

Önceki bölümlerde, Türkçe bir sözcüğün birden çok çekim kümesinden oluşabileceği anlatılmıştır. Bu nedenle, ayrıştırmada kullanılacak birimler tanımlanırken, bu yapının nasıl ifade edileceği ile ilgili kararlar alınmalıdır. Aşağıda, bu doğrultuda hazırlanmış farklı birim seçim modelleri tanıtılmaktadır.

Sözcük Tabanlı Model 1

Böyle bir araştırmada, akla ilk gelen fikir, diğer dillerde yapıldığı gibi ayrıştırmada kullanılan en küçük birim olarak sözcükleri seçmektir. Bu seçim yeni sorular ortaya çıkarmaktadır:

- Her çekim kümesi kendine ait bir sözcük sınıfı ve biçimbilimsel bilgileri barındırdığına göre, birden fazla *ÇK* içeren bir sözcüğü ifade etmek için hangi *ÇK*'ye ait sözcük sınıfı ve biçimbilimsel özellikler kullanılmalıdır? İlk yöntem olarak, sözcüğün gösteriminde, içerisinde barındırdığı tüm bilgileri kullanma yaklaşımı benimsenebilir. Böyle bir yöntemde, sözcüğü oluşturan tüm *ÇK*'lerin birleşimi kullanılabilir. Bu mantıkla oluşturulan ilk model "*sözcük tabanlı model 1*" olarak adlandırılacaktır. Bu modelde, uzaklık fonksiyonu birimler arasında yer alan sözcük sınırları kullanılarak hesaplanmıştır. Şekil 3.3, bu modelde ayrıştırma birimi olarak kullanılan sözcüklerin yukarıda anlatılan dinamik seçim yöntemine göre gösterimlerini vermektedir. Örnekte, iki *ÇK*'den oluşan "okuldaki" sözcüğü ayrıştırma sırasında iye ($u_{\mathcal{H}(i)}$) olarak kullanıldığında birinci *ÇK*'sinden "+Noun+Pnon" ve ikinci *ÇK*'sinden "Adj" imlerini alarak, bunların birleşimi olan "+Noun+Pnon+Adj" ile gösterilir. Aynı sözcük uydu (u_i) olarak kullanıldığında ise birinci *ÇK*'sinden "Loc" ve ikinci *ÇK*'sinden "Adj" imlerini alır. Bu modelde ve bölümün devamında anlatılacak diğer modellerde, bağlam içerisinde yer alan komşu birimlerin gösterimleri de ilişkili oldukları birime göre olan konumlarına dayanarak belirleneceklerdir. İlişkili oldukları birimin sol tarafında yer alan komşu birimler uydu olarak, sağ tarafından yer alanlar ise iye olarak işlem göreceklerdir.



Uydu olarak kullanıldığında	<+Det>	<+Loc+Adj>	<+Gen>
İye olarak kullanıldığında	<+Det>	<+Noun+Pnon+Adj>	<+Noun+Pnon>

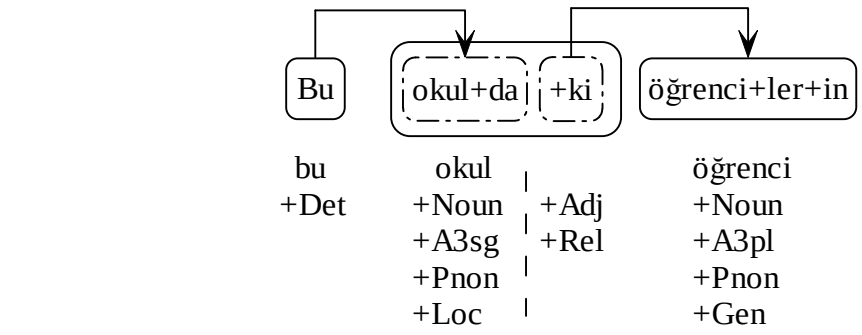
Şekil 3.3: Sözcük Tabanlı Model 1

- Ortaya çıkan ikinci soru ise şudur: Eğer ayrıştırma birimi olarak sözcükler seçilirse, esas amacımız olan *ÇK*'ler arası bağlılıklar nasıl tespit edilebilir? Bu durumda, birbirlerine bağlanacak *ÇK*'leri seçmek üzere bazı varsayımlarda bulunulması gerekmektedir. Uydu sözcüğün bağlılığın çıktığı *ÇK*'sini belirlemek kolay bir karardır. Bağlılıklar her zaman için uydu sözcüğün son *ÇK*'sinden çıktığından dolayı, uydunun son *ÇK*'si ayrıştırma sırasında uydu *ÇK* olarak seçilecektir. İye *ÇK*'nin seçim kararı daha zor bir karardır. Bu tür bir sözcük tabanlı model temel yapı hakkında gerekli bilgiyi sağlamayacağından ötürü, bağlanılacak *ÇK* ile ilgili ya bir varsayımda bulunulması veya bir ardişlemci geliştirilerek iye *ÇK*'nin hangisi olduğuna karar verilmesi gerekmektedir. Bu tür bir ardişlemci geliştirmek, bu modelin ileriki bölümlerde tanıtılacak modellerden daha yüksek *sözcükler arası başarım* vermesi halinde anlamlı olabilir. Bir diğer deyişle, eğer bu model, *ÇK*'ler arası bağlılıkları gözardı ettiğimiz takdirde, sözcükler arası bağlılıkları bulmada en iyi yöntem ise, bu tür bir ardişlemci geliştirilmesi düşünülebilir. Ancak ileriki bölümlerde deney sonuçlarından da görüleceği gibi, böyle bir durum söz konusu değildir. Bu nedenle, iye *ÇK*'nin konumu için de uydu için yapıldığı gibi varsayımda bulunulacaktır. Mevcut derlem incelendiği zaman, derlem içerisinde bağlılıkların %85.6'sının iye sözcüğün ilk *ÇK*'sine bağlandığı görülmektedir. Aynı zamanda, uydu sözcüğü iye sözcüğün ilk *ÇK*'sine bağlayan birinci temel ayrıştırıcımızın,

yaptığımız deneyler sonucunda iye sözcüğün son *ÇK*'sine bağlayan ikinci temel ayrıştırıcımızdan daha yüksek başarım verdiği görülmektedir. Bu gözlemler ışığında bu modelde, sözcükler arası bağlantıların iye sözcüğün ilk *ÇK*'sinde sonlandığı varsayımı yapılmaktadır.

Sözcük Tabanlı Model 2

Uydu *ÇK*'lerin her zaman için sözcüğün son *ÇK*'si olmasından yola çıkılarak, ikinci bir model geliştirilebilir. Bu modelde tüm sözcükleri, içerdikleri *ÇK*'lerin birleşimi ile ifade etmek yerine, uydu sözcüklerin ve iye sözcüklerin gösterimlerinde farklılaştırmaya gidilir. "*sözcük tabanlı model 2*" olarak adlandırdığımız bu modelde, sözcüğün ayrıştırma içerisinde aldığı göreve bağlı olarak yine bir dinamik yaklaşımla, uydu sözcükler sadece son *ÇK*'leri ile ifade edilmişlerdir. İye sözcükler ise önceki modeldeki gibi içerdikleri *ÇK*'lerin birleşimi olarak ifade edilirler. Şekil 3.4'de Şekil 3.3'de verilen örneğin bu yaklaşım ile işlenmesi gösterilmektedir. Örnekte, iki *ÇK*'den oluşan "okuldaki" sözcüğü ayrıştırma sırasında iye ($u_{\mathcal{H}(i)}$) olarak kullanıldığında birinci *ÇK*'sinden "+Noun+Pnon" ve ikinci *ÇK*'sinden "Adj" imlerini alarak, bunların birleşimi olan "+Noun+Pnon+Adj" ile gösterilir. Aynı sözcük uydu (u_i) olarak kullanıldığında ise sadece ikinci (son) *ÇK*'sinden "Adj" imini alır.

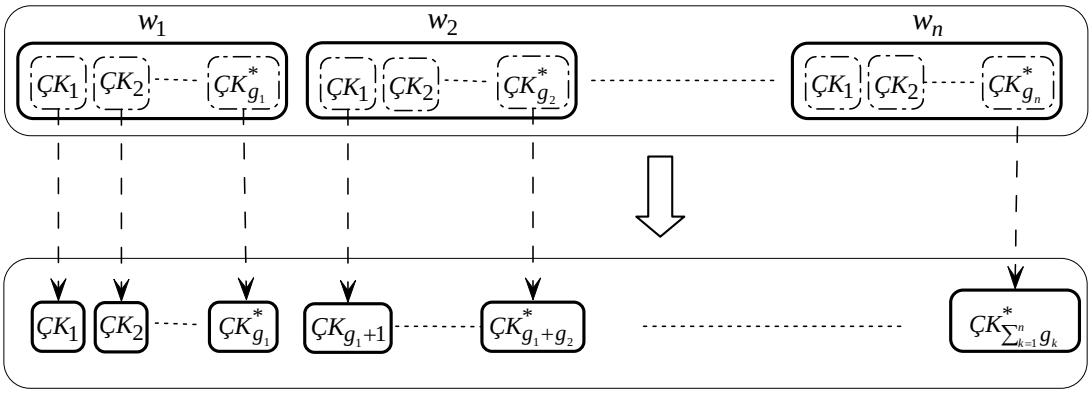


Uydu olarak kullanıldığında	<+Det>	<+Adj>	<+Gen>
İye olarak kullanıldığında	<+Det>	<+Noun+Pnon+Adj>	<+Noun+Pnon>

Şekil 3.4: Sözcük Tabanlı Model 2

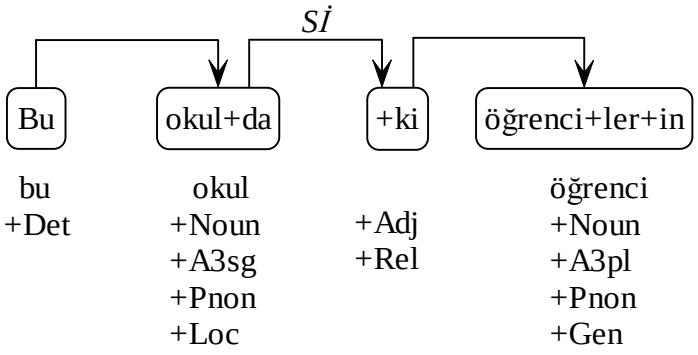
ÇK Tabanlı Model 1

Birim seçiminde geliştirilen üçüncü yaklaşım ise ayrıştırma birimi olarak sözcükleri değil doğrudan çekim kümelerini kullanmaktır. *ÇK tabanlı model 1* adını verdiğimiz bu yöntemde, sözcükler *ÇK*'lerinin birleşimi olarak gösterilmeleri yerine, *ÇK*'lerine ayrılmış ve bu *ÇK*'ler yeniden numaralandırılmıştır. Şekil 3.5 *ÇK* tabanlı modele geçişi göstermektedir. Sözcüklerin son *ÇK*'leri ile ilgili bilgiyi kaybetmemek üzere, sözcük sonunda yer alan *ÇK*'ler, farklı bir şekilde gösterilmiş ve sadece bu *ÇK*'lerden çıkan bağlılıkların bulunması amaçlanmıştır. Şekilde bu yapı $ÇK^*$ olarak gösterilmiştir. *ÇK* tabanlı model 1'de, bağlam içerisinde yer alan komşular, birimin sol tarafında yer alan *ÇK*'ler (soldaki sözcüğün son *ÇK*'sinden başlanarak) ve sağ tarafında yer alan *ÇK*'ler olarak kullanılmışlardır.



Şekil 3.5: Çekim Kümeli Gösterime Geçiş

Şekil 3.6 *ÇK* tabanlı modellerde, birimlerin gösterimini vermektedir. Her birim (*ÇK*), Bölüm 3.2.2'de anlatılan dinamik seçim yöntemi kullanılarak ifade edilmektedir. Sözcüklerin içerdikleri *ÇK*'lerin parçalara ayrılarak, birbirlerinden bağımsız olarak işlenmeleri sonucunda, aynı sözcük içindeki *ÇK*'ler arasındaki bağlılığı ifade etmek üzere sisteme yeni bir bağlılık türü eklenmiştir. Sözcük içi ("Sİ") olarak adlandırılan bu bağlılık ile, aynı sözcük içerisinde yer alan tüm *ÇK*'ler (son *ÇK* dışında) yine aynı sözcük içerisinde kendilerinden sonra gelen ilk *ÇK*'ye bağlanırlar. Örnekte, "okuldaki" sözcüğünü oluşturan iki *ÇK*'nin birbirlerine bu bağlılık türü ile bağlandığı görülebilir.



Uydu olarak kullanıldığında	<+Det>	<+Loc>	<+Adj>	<+Gen>
İye olarak kullanıldığında	<+Det>	<+Noun+Pnon>	<+Adj>	<+Noun+Pnon>

Şekil 3.6: *ÇK* Tabanlı Modeller

Bu modelde, uzaklık fonksiyonu olarak birimler arasında yer alan *ÇK* sınırları kullanılmıştır. Buna göre, Şekil 3.6'daki örnekte, "bu" ve "öğrencilerin" birimleri arasındaki uzaklık üçe eşittir.

ÇK Tabanlı Model 2

Geliştirilen dördüncü model, *ÇK* tabanlı model 1 ile benzer bir modeldir. Birimlerin gösterimleri yine Şekil 3.6'da verildiği gibidir. Bu modelde sözcüklerin tümce içerisindeki uydu olarak görevlerinin son *ÇK*'leri ile belirlenmesinden yola çıkılarak, bağlam bilgisinde kullanılan birimlerin bu *ÇK*'ler ile ifade edilmesi hedeflenmiştir. Getirilen yeni özellikler şunlardır:

- Bağlam bilgisi için sol ve sağ tarafta sözcüklerin sadece son *ÇK*'leri kullanılmış, sözcük sonunda yer almayan *ÇK*'ler göz ardı edilmiştir. (sadece iye sözcük ile komşunun üst üste geldiği durumlarda bağlam bilgisi olarak son *ÇK* yerine iye *ÇK*'nin gösterimi kullanılmıştır)
- Uzaklık fonksiyonu, *ÇK* sınırları yerine, sözcük sınırları kullanılarak hesaplanmıştır.

3.2.4 Deney Sonuçları

Bölüm 2.3'de bahsedildiği gibi, derlem verisinin sınırlı boyutta olması ve sola bağımlı ve kesişmeyen bağlılık ilkesine aykırı örneklerin derlem verisine oranla az miktarda oluşları, bu tür bağlılıkları içeren örneklerin sayısının çok az olmasına yol açmaktadır. Bu nedenle ilk incelemeler, derlem verisinin sadece sağa bağımlı ve kesişmeyen bağlılık ilkesine uygun tümceler içeren bir alt kümesi üzerinde yapılacaktır. Bu küme üzerinde farklı modellerin başarımları ölçülerek, etkileri incelendikten sonra, ileriki bölümlerde tüm derlem üzerinde ayrıntılı incelemeler yapılacaktır. Tez içerisinde, bu iki veri kümesi şu şekilde isimlendirilecektir:

- "*KsmSb Derlem* " (Ksm: Kesişmeyen, Sb: Sağa bağımlı)[9]
- "*Tüm Derlem* "

Temel ayrıştırıcılar dışında diğer tüm ayrıştırıcıların değerlendirilmesi sırasında "10 katlı çapraz doğrulama"[⋊] tekniği kullanılmıştır. Veri öncelikle rastgele 10 eşit kümeye bölünmüş[10], daha sonra her ayrıştırıcı bu veri üzerinde on defa çalıştırılarak ortalama başarımı bulunmuştur. Ayrıştırıcı, her döngüsünde verinin farklı bir kümesini sınama, geri kalan dokuz kümesini de eğitim amaçlı olarak kullanmaktadır. Sonuçlar, 10 katlı çapraz doğrulama sonucunda elde edilen değerlerin ortalaması ve standart hatası olarak verilmiştir.

Deneyler sırasında kullanılan değerlendirme ölçütleri ise şunlardır:

- *ÇKB* (Çekim kümeleri arası başarım): Ayrıştırma birimlerinin doğru iye *ÇK*'ye bağlanma oranı
- *SB* (Sözcükler arası başarım): Ayrıştırma birimlerinin doğru iye sözcüğe bağlanma oranı (bağlanılan *ÇK* doğru iye *ÇK* olmayabilir.)
- *TB* (Tümce başarımı): Bir tümce içerisindeki tüm birimlerin doğru iye *ÇK*'ye bağlanma oranı.

[9] *KsmSb Derlem* , Türkçe Ağaç Yapılı Derlemin 3398 tümcesinden oluşmaktadır.

[10] Derlem kümeleri için eşitlik her kümenin eşit sayıda tümce içermesi anlamına gelmektedir.

Tez kapsamında, yeni eklenen modellerde, ana hedef daha yüksek *ÇKB* başarımları elde etmektir. Bunun yanısıra gerekli görüldüğü yerlerde *SB* ve *TB* başarımları da verilecektir. Derlem içerisinde noktalama işaretlerinin bağlanmasında bir standart görülememektedir. İleriki bölümlerde daha ayrıntılı değinilecek olan bu durum, diğer bir çok dil için oluşturulmuş derlemlerin de ortak sorunudur. Bu nedenle ilgili çalışmalarda, noktalama işaretlerini başarım ölçümlerinin dışında bırakmak gelenek haline gelmiştir. Ölçümler sırasında, noktalama işaretlerinden çıkan bağlılıklar tüm ölçütlerde değerlendirme dışı bırakılmışlardır.[11] Yine benzer şekilde, sözcük sonunda yer almayan *ÇK*'lerin yanlarındaki ilk *ÇK*'ye bağlandıkları varsayılmış ve "Sözcük içi" olarak adlandırılan bu bağlılık türleri de değerlendirme dışı bırakılmışlardır.

Tablo 3.1 Bölüm 3.1'de anlatılan temel ayrıştırıcılar ile yapılan ayrıştırma sonucunda elde edilen *ÇKB* ve *TB* başarımlarını vermektedir. Tablodan da görülebileceği gibi uydu sözcükleri sağ taraflarındaki sözcüklerin ilk *ÇK*'lerine bağlayan birinci temel ayrıştırıcımız, uydu sözcükleri sağ taraflarındaki sözcüklerin son *ÇK*'lerine bağlayan ikinci temel ayrıştırıcımızdan %1,7 daha yüksek *ÇKB* başarımı sağlamaktadır. Tablonun en son satırında yeralan kural tabanlı ayrıştırıcı ise 70,5'lik *ÇKB* ile ilk iki temel ayrıştırıcıdan da daha yüksek başarım göstermiştir. Temel ayrıştırıcıların içerisinde benzer bir sıralamanın *TB* başarımı için de geçerli olduğu görülmektedir.

Tablo 3.1: Temel Ayrıştırıcılar ile Ayrıştırma Sonuçları

Model	*ÇKB*	*TB*
Temel ayrıştırıcı 1	63,9	24,0
Temel ayrıştırıcı 2	62,2	22,6
Temel ayrıştırıcı 3	70,5	36,6

Olasılık tabanlı ayrıştırıcı tarafından kullanılan parametreler (her dört model için aynı olmak kaydıyla) şöyledir:

- Dl ve Dr: Φ_i bağlam bilgisi içerisinde kullanılmak üzere, uydu birimin sol (Dl) ve sağ (Dr) taraflarından kaçar adet komşu birim kullanılacağının sayısı,

[11] Bölüm 3.2'de yer alan başarımların noktalama işaretleri dahil edilerek hesaplanmış hallerine Eryiğit ve Oflazer (2006)'den ulaşılabilir. Modellerin başarımlarının noktalama işaretleri dahil edilmiş veya edilmemiş halleri arasında başarım sıralamasında bir fark yoktur.

- Hl ve Hr: $\Phi_{\mathcal{H}(i)}$ bağlam bilgisi içerisinde kullanılmak üzere, iye birimin sol (Hl) ve sağ (Hr) taraflarından kaçar adet komşu birimin kullanılacağının sayısı,
- k: Uzaklık fonksiyonu içerisinde kullanılan eşik değeri,
- d: Geriye doğru demetli arama algoritmasında kullanılan demet boyu (her adımda d adet en olası ayrıştırma demette tutulmaktadır).

Deneyler sırasında, demet boyu olarak en yüksek başarımı verdiği gözlenen $d = 3$ değeri kullanılmıştır. Derlem tümceleri üzerinde yapılan bir istatistiksel çalışmayla, ayrıştırma birimi olarak sözcükler kullanıldığında, bağlılıkların %90'ının 3 veya daha yakın uzaklıkta bir sözcükte sonlandığı görülmüştür. Benzer şekilde, ayrıştırma birimi olarak *ÇK*'ler de alındığında, bağlılıkların %90'ının 4 veya daha yakın uzaklıkta bir *ÇK*'de sonlandığı görülmüştür. Bu nedenle, uzaklığın sözcük bazında ölçüldüğü sözcük tabanlı model 1 ve model 2 ve *ÇK* tabanlı model 2'de k parametresi 4 olarak alınmıştır. Uzaklığın *ÇK* bazında ölçüldüğü *ÇK* tabanlı model 1'de ise $k = 5$ olarak alınmıştır. Dl, Dr, Hl, Hr parametreleri için model bazında eniyileştirme"$^{\bowtie}$" yapılmıştır.

Tablo 3.2'de, geliştirilen modeller için en iyi sonuçları veren parametre kümeleri ve seçilmiş bazı diğer parametre kümeleri kullanılarak elde edilen başarımlar verilmektedir. Bu tabloda, "Bağlam" sütunundaki değerler uydu ve iye sözcüğün etrafındaki bağlam bilgisini belirtmektedirler. *Dl=1* ve *Dr=1* uydunun solundan ve sağından birer birimin bağlam bilgisi olarak kullanılacağını belirtmektedir. Benzer şekilde, *Hl=1* ve *Hr=1* de iye birimin solundan ve sağından birer birimin bağlam bilgisi olarak kullanılacağını belirtmektedir. Yapılan deneylerde, bu parametreler için kullanılan birden büyük değerlerin başarımda artış sağlamadığı görülmüştür. Bu tabloda, modeller için elde edilen en yüksek *ÇKB* başarımları koyu yazılarak belirtilmiştir.

Tablo 3.3, Tablo 3.2'de en yüksek *ÇKB* başarımlarını veren yapılandırmaların ve temel modellerin başarımlarını özet bir tabloda toplamaktadır. Tablonun üçüncü ve dördüncü sütunlarında, modellerin *SB* ve *TB* başarımları da verilmektedir. Bu değerlerden görülebileceği üzere, sözcükleri (hem uydu hem de iye) içerdikleri *ÇK*'lerin birleşimi olarak ifade eden salt sözcük tabanlı modelimiz (sözcük tabanlı nodel 1) *ÇKB* =

Tablo 3.2: Olasılık Tabanlı Modeller ile Ayrıştırma Sonuçları

Model	Bağlam	*ÇKB*
Sözcük tabanlı model 1 (k=4)	Yok	71,1$\pm$1,2
	Dl=1	71,1$\pm$1,2
	Dl=1 Dr=1	70,3$\pm$1,1
	Hl=1 Hr=1	71,1$\pm$1,3
	Dl=1 Dr=1 Hr=1	**71,2**$\pm$1,1
	Dl=1 Dr=1 Hl=1 Hr=1	71,1$\pm$1,2
Sözcük tabanlı model 2 (k=4)	Yok	71,0$\pm$1,3
	Dl=1	71,1$\pm$1,2
	Dl=1 Dr=1	72,5$\pm$1,2
	Hl=1 Hr=1	65,5$\pm$1,3
	Dl=1 Dr=1 Hr=1	72,0$\pm$1,1
	Dl=1 Dr=1 Hl=1 Hr=1	**72,6**$\pm$1,1
ÇK tabanlı model 1 (k=5)	Yok	71,9$\pm$1,0
	Dl=1	72,7$\pm$0,9
	Dl=1 Dr=1	73,1$\pm$0,9
	Hl=1 Hr=1	57,6$\pm$0,7
	Dl=1 Dr=1 Hr=1	**73,3**$\pm$0,9
	Dl=1 Dr=1 Hl=1 Hr=1	72,2$\pm$0,9
ÇK tabanlı model 2 (k=4)	Yok	72,6$\pm$0,9
	Dl=1	72,6$\pm$1,1
	Dl=1 Dr=1	**73,5**$\pm$1,0
	Hl=1 Hr=1	55,1$\pm$0,7
	Dl=1 Dr=1 Hr=1	72,7$\pm$0,9
	Dl=1 Dr=1 Hl=1 Hr=1	72,4$\pm$0,9

71,2$\pm$1,1 ile diğer tüm olasılık tabanlı modellerden daha kötü sonuç vermiştir. Bu değer temel ayrıştırıcılarımızın *ÇKB* başarımlarından daha yüksek olmasına karşın, Tablo 3.3'e bakıldığında sözcük tabanlı model 1'in *SB* ve *TB* başarımının, kural tabanlı ayrıştırıcının (temel ayrıştırıcı 3) gerisinde kaldığı görülmektedir. Bir diğer deyişle, geliştirilen kural tabanlı ayrıştırıcı, sözcüklerin bağlanacağı iye sözcüğü bulmada bu modele göre daha başarılı iken, bağlanılan doğru *ÇK*'yi tahmin etmede aynı başarımı gösterememektedir.

Sözcük tabanlı model 1 dışında diğer tüm istatiksel modellerimiz, tüm değerlendirme ölçütlerinde temel modellerimizden daha yüksek başarımlı sonuçlar vermişlerdir. *ÇK* tabanlı her iki modelin de başarımlarının birbirlerine çok yakın olduğu söylenebilir.

İstatistiksel olarak anlamlı olmasa bile, en yüksek *ÇKB* başarısını %73,5 ile *ÇK* tabanlı model 2'nin sağladığı görülmektedir.

Tablo 3.3: Olasılık Tabanlı Modeller ve Temel Ayrıştırıcılar Özet Tablo

Model	**Bağlam**	*ÇKB*	*SB*	*TB*
Temel ayrıştırıcı 1	-	63,9	72,1	24,0
Temel ayrıştırıcı 2	-	62,2	72,1	22,6
Temel ayrıştırıcı 3	-	70,5	80,3	36,6
Sözcük tabanlı model 1	Dl=1 Dr=1 Hr=1	71,2±1,1	79,1±0,7	34,4±0,4
Sözcük tabanlı model 2	Dl=1 Dr=1 Hl=1 Hr=1	72,6±1,1	80,8±0,9	37,2±0,6
ÇK tabanlı model 1	Dl=1 Dr=1 Hr=1	73,3±0,9	81,3±0,8	38,7±0,9
ÇK tabanlı model 2	Dl=1 Dr=1	73,5±1,0	81,2±1,0	38,7±0,9

Tablo 3.4: Daha Az Eğitim Verisi Kullanmanın Etkileri

Model	**Bağlam**	***ÇKB***
ÇK tabanlı model 2	Yok	72,2 ±1,5
(k=4, 1500 tümce)	Dl=1 Dr=1	72,6 ±1,1

Tablo 3.4 eğitim verisi olarak daha küçük boyutlu bir derlem kullanmanın *ÇK* tabanlı model 2 üzerindeki etkisini göstermektedir. Bu incelemede, her bir 10 katlı çapraz doğrulama kümesi kendi içerdiği tümceler dışında kalan 1500 tümce ile eğitilmiş olan ayrıştırıcı ile sınanmıştır.[12] Eğitim verisinin boyutunu küçültmenin, ayrıştırıcının başarımını önemli ölçüde düşürmediği gözlemlenmiştir. Bu durum, görünüm bilgisi içermeyen modelimizin derlem boyutundan çok fazla etkilenmendiği ve oldukça etkin bir model olduğu şeklinde yorumlanabilir. Ancak bu durum aynı zamanda, görünüm bilgisi kullanmayan bu tür bir modelleme ile daha büyük bir derlemin kullanılmasının bile bağlılık başarımını arttırmada çok yararlı olmayacağı anlamına da gelebilir.

Tablo 3.5: Farklı Uzunluktaki Tümceler Üzerinde Başarım

Tümce Uzunluğu l (*ÇK* bazlı)	***ÇKB***
$1 < l \leq 10$	80,2 ±0,5
$10 < l \leq 20$	70,1 ±0,4
$20 < l \leq 30$	64,6 ±1,0
$30 < l$	62,7 ±1,3

[12] Önceki deneylerde bu sayı yaklaşık olarak 3058'dir. (≈3398*9/10)

En iyi başarımı veren modelin sonuçları üzerinde daha ayrıntılı bir inceleme Tablo 3.5'de verilmektedir.[13] Buradaki incelemede ayrıştırıcı farklı uzunluktaki tümceler üzerinde sınanmıştır. Tablodan görülebileceği gibi, tümce uzunluğu arttıkça, başarım azalmaktadır. Özellikle uzun tümceler için, görünüm bilgisi de içeren daha karmaşık modellere gereksinim duyulmaktadır.

3.2.5 Kısım Sonucu

Bu kısımda tarafımızdan geliştirilmiş olan, birimlerin ikili bağlanma olasılıklarına dayanan olasılık tabanlı bir ayrıştırıcı tanıtılmıştır. Türkçe'nin nasıl modelleneceğine ilişkin yapılan bu ilk araştırmalarda (Eryiğit ve Oflazer, 2006), görünüm bilgisi içermeyen modeller sadece kesişmeyen ve sağa bağımlı türde bağlılıklar içeren tümceler üzerinde denenmiştir. Seyrek veri sorununun yaşandığı bu modellerde, birimler ifade edilirken görünüm bilgisi yerine, sözcüğü oluşturan alt parçaların sınıf bilgisi ve biçimbilimsel özellikleri dinamik bir seçim yöntemi ile kullanılmıştır. En yüksek başarım, ayrıştırma birimi olarak çekim kümelerinin kullanıldığı modeller ile elde edilmiştir. 10 katlı çapraz doğrulama sonucunda elde edilen başarım değerlerine bakıldığında, bu değerler için ortaya çıkan standart hata aralıklarının oldukça geniş olduğu ve *ÇK* tabanlı modeller arasında istatistiksel olarak anlamlı bir fark olmadığı görülmektedir.

3.3 Sınıflandırıcı Tabanlı Ayrıştırıcı

Bölüm 3.2'de, Türkçe'nin bağlılık çözümlemesi ile ilgili ilk incelemeler olasılık tabanlı bir ayrıştırma yöntemi kullanılarak yapılmıştır. Bu bölümde, benzer tasarım modellerinin sınıflandırıcı tabanlı bir ayrıştırıcı üzerindeki etkileri incelenecektir. Sınıflandırıcı tabanlı ayrıştırıcı, olasılık tabanlı ayrıştırıcıya benzer biçimde, tümcelerin çözümlemesini herhangi bir gramer kuralı kullanmadan, eğitim verisi üzerinden tümevarımsal çıkarım yaparak gerçekleştirmektedir. Her iki veri güdümlü ayrıştırma algoritması da sağlam ve verimlidir. Burada ayrıştırıcının sağlam olması

[13] Bu sonuçların hepsi, tüm tümce uzunluğu sınıfları için en iyi temel modelimizden (kural tabanlı) istatistiksel olarak anlamlı bir şekilde daha yüksektir.

herhangi bir tümce için her zaman bir çözüm üretebilmesi anlamına gelmektedir. Verimli olması ise çözümleme süresinin, tümce uzunluğu ile doğrusal veya karesel orantılı olmasıdır. Aşağıda ilk olarak, geliştirilen ayrıştırıcının mimarisi ve daha sonra bu yönteme uygun olarak hazırlanmış tasarım modelleri tanıtılacaktır.

3.3.1 Mimari

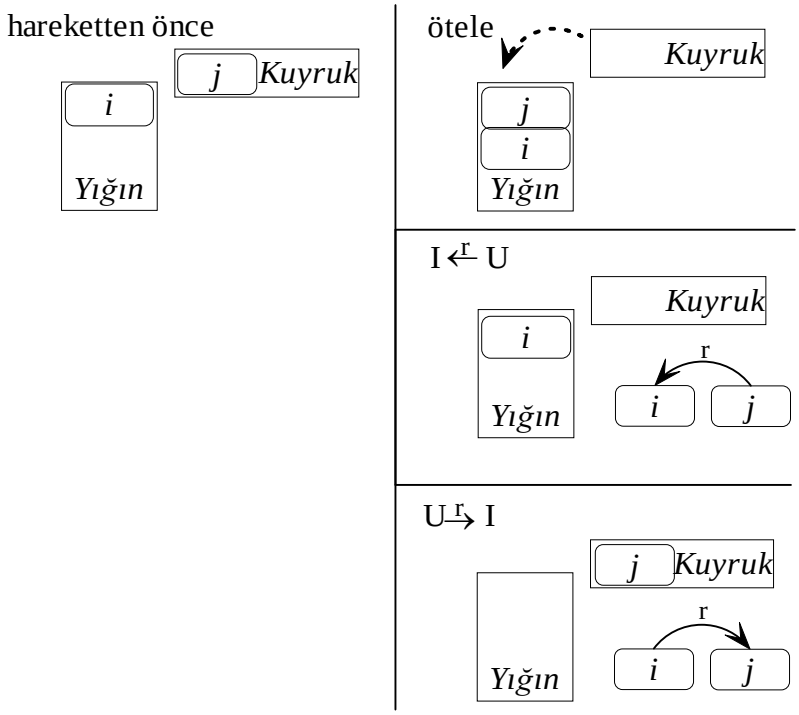
Veri güdümlü bir bağlılık ayrıştırıcısı olan sınıflandırıcı tabanlı ayrıştırıcı üç farklı tekniğin birleşiminden oluşmaktadır:

1. Bağlılık grafiğini oluşturmak için kullanılan gerekirci bir ayrıştırma algoritması (Kudo ve Matsumoto, 2002; Yamada ve Matsumoto, 2003; Nivre, 2003),
2. Ayrıştırıcının bir sonraki hareketini belirlemek üzere kullanılan geçmişe dayalı ayrıştırma modeli (Black ve diğ., 1992; Magerman, 1995; Ratnaparkhi, 1997; Collins, 1999),
3. Geçmişte olan olayları ayrıştırıcının hareketleri ile ilişkilendirmek üzere kullanılan ayırdedici sınıflandırıcı (Veenstra ve Daelemans, 2000; Kudo ve Matsumoto, 2002; Yamada ve Matsumoto, 2003; Nivre ve diğ., 2004).

Bu ayrıştırıcıda, giriş tümcesi üzerinden soldan sağa doğru tek geçişte bağlılık etiketli bir bağlılık grafiği oluşturan Nivre (2003; 2006a)'nin doğrusal zamanlı gerekirci algoritması kullanılmıştır. Diğer birçok bağlılık ayrıştırıcısında olduğu gibi bu algoritma da kesişmeyen bağlılıklardan oluşan tümceleri ayrıştırmakla sınırlıdır.[14] Bu algoritmanın iki farklı çeşidi vardır. Bunlardan birincisi *olağan yay*$^{\prec}$ ikincisi ise *hevesli yay*$^{\prec}$ olarak adlandırılır. İki yöntemde de, ayrıştırıcı kısmi olarak işlenmiş birimlerin tutulduğu bir yığın σ ve sırada işlenmek üzere bekleyen birimlerin tutulduğu bir kuyruk τ olmak üzere iki ana veri kümesinden faydalanır. Algoritma, işleme, giriş tümcesinin tüm birimlerinin tutulduğu bir kuyruk ve boş bir yığın ile başlar. Giriş listesinin (kuyruk) boşalması ile de son bulur. σ ve τ listelerindeki elemanlar

[14]Türkçe derlem içerisinde kesişen bağlılık örneklerinin miktarının bu tür bağlılıkları öğrenebilmek için yeterli olmamasından dolayı, diğer birçok dil için başarılı olduğu raporlanan Nivre ve Nilsson (2005)'nin kesişen bağlılıklara özel yaklaşımı kullanıldığında, başarımda herhangi bir artış gözlemlenememiştir. Buna ek olarak, derlem içerisindeki kesişen bağlılıklar incelendiğinde bunların genelde hatalı bağlılıklardan kaynaklandığı görülmektedir.

0'dan başlayarak numaralandırılırlar. Bu numaralamada, σ_0 yığının en üstünde duran elemanı (*üst birim*), τ_0 ise sırada bekleyen kuyruktaki ilk elemanı (*sıradaki birim*) belirtir; σ_0 ve τ_0 ayrıştırma algoritması tarafından bir bağlılık ilişkisine aday birimler olarak görüldüğü için ikisi birlikte *hedef birimler* olarak adlandırılırlar.



Şekil 3.7: Ayrıştırıcı Hareketleri

Olağan yay yöntemi, üç hareket içermektedir:

1. $U \xrightarrow{r} I$:

   $(\sigma|i, j|\tau, h, d) \rightarrow (\sigma, j|\tau, h[i \mapsto j], d[i \mapsto r])$

   eğer $h(i) = 0$ ise

2. $I \xleftarrow{r} U$:

   $(\sigma|i, j|\tau, h, d) \rightarrow (\sigma|i, \tau, h[j \mapsto i], d[j \mapsto r])$

   eğer $h(j) = 0$ ise

3. ÖTELE:

   $(\sigma, j|\tau, h, d) \rightarrow (\sigma|j, \tau, h, d)$

Yukarıdaki tanımlamada, i: yığının en üstünde duran birimin sıra numarasını, j: kuyrukta bekleyen, sıradaki birimin sıra numarasını, h: birimler arası bağımlılıkları

tutan fonksiyonu ve d: bir birimden çıkan bağımlılığın türünü tutan fonksiyonu (başlangıçta tüm $i \in \{1..n-1\}$ için h(i)=0 ve d(i)=0) belirtmektedir. Hareketlerden sonra kuyruğun ve yığının aldığı yeni durumlar, Şekil 3.7'de gösterilmektedir.

"Ötele"me işleminde kuyrukta bekleyen eleman yığına itilir. Bu işlem yığının boş olduğu durumlarda veya "i" ve "j" sıra numaralı elemanlar arasında herhangi bir bağlılık kurulamadığı durumlarda gerçekleşir.

"$U \xrightarrow{r} I$" işlemi "i" sıra numaralı eleman ile tümce içerisinde onun sağ tarafında yer alan "j" numaralı eleman arasında "r" etiketli bir uydu-iye ilişkisi olduğu durumlarda gerçekleşir. Bu işlemin yürütülebilmesi için üst birimin daha önceden başka bir yere bağlanmamış olması ($h(i) = 0$) gerekir. Hedef birimler arasında bağlılık oluşturulduktan sonra, yığının en üstündeki eleman çekilir.

"$I \xleftarrow{r} U$" işlemi ise iye sözcüğün uydunun sol tarafında yer alması durumlarında kurulan "r" türünden bağlılıklar için geçerlidir. Sıradaki birimin daha önceden başka bir yere bağlanmamış olması ($h(j) = 0$) gerekir. Hedef birimler arasında ("j" uydu, "i" iye olacak şekilde) bağlılık oluşturulduktan sonra, kuyruktaki ilk eleman kuyruktan çıkarılır ve kuyrukta bekleyen sıradaki birime doğru ilerlenir.

Hevesli yay yöntemi, aşağıdan yukarıya ayrıştırma yapan birinci yönteme ek özellikler getirerek aynı zamanda yukarıdan aşağıya da işlem yapmaktadır. Başka bir deyişle, soldaki uyduların aşağıdan yukarıya, sağdaki uyduların ise yukarıdan aşağıya işlenmelerini sağlar. Birinci yöntemin yapısı itibari ile $I \xleftarrow{r} U$ hareketi gerçekleştiğinde, j sıra numaralı eleman kuyruktan çekilir ve kuyrukta bekleyen sıradaki eleman doğru ilerlenir. Bu nedenle, j numaralı elemana onun sağ tarafında yer alan herhangi bir elemanın uydu olarak bağlanması mümkün değildir. Olağan yay yöntemiyle, $a \xleftarrow{I \leftarrow U} b \xleftarrow{I \leftarrow U} c$ şeklinde arka arkaya gelen, iki sola bağımlı türde bağlılığın ayrıştırılabilmesi mümkün değildir.

Hevesli yay yöntemi, dört hareket içermektedir. Ötele ve $U \xrightarrow{r} I$ hareketleri birinci yöntemle aynıdır. Bunlara ek olarak, *indirgeme* hareketi eklenmiş ve $I \xleftarrow{r} U$ hareketinde değişiklik yapılmıştır:

1. $U \xrightarrow{r} I$:

   $(\sigma|i, j|\tau, h, d) \rightarrow (\sigma, j|\tau, h[i \mapsto j], d[i \mapsto r])$

   eğer $h(i) = 0$ ise

2. $I \xleftarrow{r} U$:

   $(\sigma|i, j|\tau, h, d) \rightarrow (\sigma|i|j, \tau, h[j \mapsto i], d[j \mapsto r])$

   eğer $h(j) = 0$ ise

3. ÖTELE:

   $(\sigma, i|\tau, h, d) \rightarrow (\sigma|i, \tau, h, d)$

4. İNDIRGE:

   $(\sigma|i, \tau, h, d) \rightarrow (\sigma, \tau, h, d)$

   eğer $h(i) \neq 0$ ise

$I \xleftarrow{r} U$ hareketindeki fark, bağlılık kurma işleminden sonra "j" sıra numaralı elemanın yığına atılmasıdır. İndirgeme işlemi ise önceden bir iye birime bağlanmış olması koşulu ile üst birimi yığından çeker.

Türkçe derlem içerisinde, $a \xleftarrow{I \leftarrow U} b \xleftarrow{I \leftarrow U} c$ türünde bağlılıkların görülme sıklığı %0,1'den daha azdır. Bu nedenle, diğer diller için başarımı arttıran ancak sınıflandırıcının ayırt etmesi gereken sınıf sayısının artması açısından sistemi karmaşıklaştıran hevesli yay yönteminin, Türkçe için başarımı arttırmadığı, aksine çok küçük bir oranda azalmaya[15] neden olduğu görülmüştür. Bu nedenle, burada olağan yay yöntemi kullanılmıştır.

Ayrıştırma algoritması, Bölüm 3.1'de tanıtılan algoritmaya benzer şekilde çalışmaktadır. Burada farklı olarak, ayrıştırıcı modeli olağan yay yönteminde tanımlı üç farklı hareketten birini seçerek, ayrıştırma algoritmasına iletir. Bu aşamadan sonra algoritma gerekli hareketi yürütür.

[15] Bölüm 4'de bahsedilecek olan Conll-X ortak çalışmasında, diğer diller ile uyumluluk sağlaması için Türkçe de hevesli yay algoritması kullanılarak ayrıştırılmıştır. Bu seçim hem *ÇKB* hem de $ÇKB_E$ başarımlarında %0,2 düşüşe neden olmuştur.

Birimler arası bağlılıkların doğru olarak saptanmasının yanı sıra, bu bağlılıkların türlerinin neler olduğunun bulunması da gerekli bir işlemdir. Örneğin "Ali eve gitti." tümcesinde, "Ali" sözcüğünün "gitti" sözcüğüne bağlılığının bulunmasına ek olarak bu bağlılığın türünün "Özne" olduğunun da bulunması gerekmektedir. Burada kullanılan ayrıştırıcının, ilgili yayınlardaki diğer benzer ayrıştırıcılardan önemli bir farkı bağlılık türünün bağlılığın bulunması işlemi ile aynı anda tek bir işlem olarak yapılmasıdır. Bir diğer deyişle, geçmişte olan olayları, ayrıştırıcının hareketleri ile ilişkilendirmek üzere kullanılan ayırdedici sınıflandırıcılar, örnekleri $1 + r_1 + r_2$ adet (1 adet "Ötele" sınıfı, r_1 adet farklı türde $U \rightarrow I$ sınıfı, r_2 adet farklı türde $I \leftarrow U$ sınıfı) farklı sınıfa ayırmaya çalışırlar. Bu yaklaşımın, bağlılık yapısını iki aşamalı olarak bulma yaklaşımlarına[16] (önce hareketi bulup sonra etiketi bulmak) göre daha başarılı olduğu öngörülmektedir. (Nivre ve diğ., 2006b)

Geçmişe dayalı ayrıştırma modeli, birimlerin özelliklerine bakarak bir sonraki hareketin ne olacağına karar verir. Kullanılan özellikler, hedef birimlerin özelliklerine ek olarak, yığındaki ve kuyruktaki komşuların özellikleri de olabilir. Bu modelin en önemli niteliği, birimlerin o ana kadar oluşan kısmi bağlılık ağacındaki iyelerinin veya uydularının özelliklerini veya parçası oldukları bağlılıkların türlerini de kullanmasıdır.

Belirli bir birim için kullanılabilir olan özellikler şöyledir:

- Görünüm bilgisi (tümü veya gövdesi)
- Sözcük sınıfı (ana sınıf veya alt sınıf)
- Biçimbilimsel özellikler
- Bağlılık türü (Eğer bağlanmışsa)

Özellikler, dinamik ve statik olmak üzere iki kümeye ayrılabilirler. Statik özellikler ayrıştırma boyunca aynı kalan özelliklerdir. Bunlar görünüm bilgileri, sözcük sınıfı ve biçimbilimsel özelliklerdir. Dinamik özellikler ise ayrıştırma sırasında değişen ve belirli bir anda kısmi olarak oluşmuş ayrıştırma ağacı kullanılarak erişilen özelliklerdir. Bağlılık türü veya hedef birimlerin iyelerinin özellikleri bu kümeye girer. Ayrıştırma

[16] Bu yaklaşımların detayları Bölüm 4'de verilecektir.

işleminin en başında, bu tür özellikler hiç atanmamış (boş) olacaklardır. Ayrıştırma işlemi ilerledikçe, doldurulmaya ve böylece kullanılmaya başlarlar.

Eğitim sırasında ilk olarak, eğitim verisi üzerindeki bağlılıkları oluşturmak üzere gerçekleştirilmesi gereken ayrıştırıcı hareketlerine ilişkin özellik vektörleri oluşturulur. Bu vektörleri eğitim verisi olarak kullanan sınıflandırıcılar, daha sonra sınama verisi üzerinde ayrıştırma işlemi yaparken, her gelen yeni durum için benzer bir özellik vektörü oluşturarak bu vektörün hangi sınıfa ait olduğuna karar verir. Bir diğer deyişle, ortaya yeni çıkan durumları geçmişte olan olayları kullanarak ilgili harekete atarlar. Araştırmalarımıza sınıflandırıcı olarak *bellek tabanlı öğrenme* yönteminin kullanılmasıyla başlanmıştır. Bu yöntem kullanılarak elde edilen ilk sonuçlar Nivre ve diğ. (2006a)'de bulunabilir. Bellek tabanlı yaklaşımlar, öğrenmeyi geçmiş deneyimlerin basit bir şekilde bellekte tutulması, yeni bir sorunu çözmeyi ise, bellekte tutulan geçmiş deneyimler içerisinden yeni soruna en benzer olanın bulunması olarak görürler. Aykırı durumların da her zaman için bellekte tutulması sayesinde, kirlilik ve aykırı durumları birbirinden ayıramayan diğer istatistiksel yöntemlere göre DDİ konusunda başarılı oldukları öne sürülmektedir (Daelemans ve Bosch, 2005).

İlerleyen dönemlerde, KDM sınıflandırıcılarının bellek tabanlı öğrenmeye göre daha yüksek başarım verdiği görülmüştür. Karar destek makineleri, ilk kez Vapnik (1995) tarafından ortaya atılan iki sınıf arasındaki sınırı büyükleme ilkesini, asıl özellik uzayını daha yüksek boyutlu bir uzaya çekmek üzere çekirdek fonksiyonları[⋊] ile birleştirirler. Bu sınıflandırıcı, Kudo ve Matsumoto (2002), Yamada ve Matsumoto (2003) ve Sagae ve Lavie (2005) gibi birçok çalışmada, gerekirci bir ayrıştırma yöntemi ile birlikte başarılı bir şekilde kullanılmıştır. Bu tezde kullanılan sınıflandırıcı tabanlı ayrıştırıcıda, sınıflandırıcı olarak KDM'ler kullanılmış ve bu amaçla LibSVM (Chang ve Lin, 2001) kütüphanesinden yararlanılmıştır. KDM iki sınıfı ayırmaya yönelik bir ayrıştırıcı olduğundan, elimizdeki çok sınıflı sınıflandırma işlemi için *bire karşı bir* yöntemi kullanılmıştır. Bu yöntemde, $1 + r_1 + r_2$ adet sınıf için $(1 + r_1 + r_2)(r_1 + r_2)/2$ adet sınıflandırıcı oluşturulur; Her bir sınıflandırıcı sadece iki sınıfı ayırt etmek üzere eğitilir. Deney sırasında ise ilgili örnek hangi sınıfa daha çok kez atanıyorsa, o sınıf seçilir. Böylece, hem çok sınıflı sınıflandırma problemi iki sınıflı sınıflandırma problemine dönüştürülmüş, hem de toplam eğitim zamanı

bire karşı hepsi yöntemine kıyasla azaltılmış olur. Bire karşı hepsi yönteminde, her bir sınıf diğer tüm sınıflardan ayırt edilmek üzere $1 + r_1 + r_2$ adet sınıfılandırıcı oluşturulur. Sınıflandırıcı sayısı daha azdır, fakat eğitim süresi veri boyutuyla karesel orantılı olduğu için çok uzundur; bu durum, eğitim verisinin çok ve çeşitli olduğu doğal dil sistemlerinde sorun yaratmaktadır (Kudo ve Matsumoto, 2002).

3.3.2 Özellik Kalıpları

Bu kısımda, sınıflandırıcıya verilecek özellik vektörlerinin oluşturulmasında kullanılan özellik kalıpları tanıtılacaktır. Bu kalıplarda kullanılan bilgi sınıflarının Conll-X derlem gösterimindeki bilgi sınıflarıyla örtüşmesi nedeniyle, öncelikle bu gösterim biçimi tanıtılacak, daha sonra kalıp yapısı anlatılacaktır.

ODTÜ-Sabancı Türkçe ağaç yapılı derlemi, Conll-X (Buchholz ve Marsi, 2006) ortak çalışmasında kullanılmak üzere konferans düzenleyicileri tarafından *her satırda bir ÇK* olacak şekilde yeni bir biçime dönüştürülmüştür. Şekil 3.8'de, Şekil 2.2'deki örnek tümcenin ortak çalışmadaki tüm diller için aynı olan bu yapı ile gösterimi verilmektedir.

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 3.8: Conll-X Veri Biçimi

Şekildeki sütunlar şu bilgileri taşırlar:

1. sütun: sıra numarası
2. sütun: görünüm bilgisi tümü (LEX)
3. sütun: görünüm bilgisi gövde (LEMMA)
4. sütun: ana sınıf (CPOS)
5. sütun: alt sınıf (POS)
6. sütun: biçimbilimsel bilgi (INF)
7. sütun: bağlanılan iye birimin sıra numarası
8. sütun: bağlılık türü (DEP)

Bu gösterimde, sözcük içi bağlılıklar "DERIV" (türeme) bağlılık türü ile belirtilmişlerdir. Örneğin 2 numaralı *ÇK* hemen sonrasından gelen 3 numaralı *ÇK*'ye bu bağlılık türü ile bağlanır. Bu tür birden fazla *ÇK* içeren sözcüklerin sadece son *ÇK*'leri LEX bilgisini taşır. LEMMA bilgisi ise sadece ilk *ÇK*'de vardır, diğer *ÇK*'lerde yoktur.

Sınıflandırıcının eğitimi ve sınaması sırasında kullanılan örneklerin özelliklerini ifade etmek üzere aşağıdaki gösterim kullanılacaktır:

1. σ_i : yığının üzerindeki i sıra numaralı birim (saymaya 0'dan başlanacak)
2. τ_i: kuyrukta sırada bekleyen i sıra numaralı birim (saymaya 0'dan başlanacak)
3. $\ell(i)$: o ana kadar kısmi olarak oluşmuş bağlılık grafiğinde i sıra numaralı birimin en soldaki uydusu
4. $r(i)$: o ana kadar kısmi olarak oluşmuş bağlılık grafiğinde i sıra numaralı birimin en sağdaki uydusu

Şekil 3.9'da bu gösterim kullanılarak hazırlanmış örnek bir özellik kalıbı verilmektedir. İleriki bölümlerde, kullanılacak özellik kalıpları Şekil 3.9'a benzer bir biçimde belirtileceklerdir. Örnekteki özellik kalıbı,

- hedef birimlerin (σ_0, τ_0),
- yığında üst birimin hemen altında yer alan birimin (σ_1),

- kuyrukta bekleyen sıradaki birimin hemen arkasındaki birimin (τ_1) ve
- gerçek tümcede üst birimin sağ tarafında yer alan birimin ($\sigma_0 + 1$) ana sınıf bilgileri ile
- üst birimin en soldaki $\ell(\sigma_0)$ ve en sağdaki $r(\sigma_0)$ uydularının ve
- sıradaki birimin en soldaki $\ell(\tau_0)$ uydusunun bağlılık türünden oluşmaktadır.

	σ_0	τ_0	σ_1	σ_0+1	τ_1	$\ell(\sigma_0)$	r(σ_0)	$\ell(\tau_0)$
CPOS	+	+	+	+	+			
DEP						+	+	+
INF								
LEMMA								

Şekil 3.9: Özellik Kalıbı 1

3.3.3 Birim Seçim Modelleri

Bu kısımda, sınıflandırıcı tabanlı ayrıştırıcı için kullanılan birim seçim modelleri sunulmaktadır. Olasılık tabanlı ayrıştırıcıdan elde ettiğimiz sonuçların ışığında bu bölümde sözcük tabanlı ve *ÇK* tabanlı olmak üzere iki farklı yaklaşım geliştirilmiştir. Sınıflandırıcı tabanlı ayrıştırıcıda, ayrıştırma algoritması, bağlılıkları her iki yönde (sağa bağımlı ve sola bağımlı) bulmak üzere çalıştığından, birimler üzerinde ilerlerken belirli bir anda birimlerin olası görevleri ile ilgili bilgiye sahip değildir. Bu nedenle, olasılık tabanlı ayrıştırıcıdan farklı olarak burada, uydu ve iye birimler, aynı özellikler ile ifade edilerek, göreve bağlı özelliklerin seçimi KDM'lere bırakılmıştır[17]. İlk olarak, önceki modellerimize benzer üç farklı birim seçim modeli geliştirilmiştir:

- Ayrıştırmada kullanılan en küçük birimin *ÇK*'lerinin bileşkesi ile gösterilen sözcükler olduğu "*Sözcük tabanlı model*",
- Ayrıştırmada kullanılan en küçük birimin *ÇK*'ler olduğu ve sözcük içi bağlılıkların gerçek bağlılıklar gibi KDM'ler tarafından bulunduğu "*ÇK tabanlı model*",

[17]Olasılık tabanlı ayrıştırıcıda kullanılan özellik indirgemesinin, sınıflandırıcı tabanlı ayrıştırıcıda başarımı arttırmadığı görülmüştür. Bunun nedeni KDM'lerin gerekli özellik seçimi konusundaki yetenekleridir.

- Ayrıştırmada kullanılan en küçük birimin *ÇK*'ler olduğu ve sözcük içi bağlılıkların KDM sınıflandırıcısına başvurulmadan belirlenimci olarak işlendiği "*ÇK tabanlı belirlenimci model*".

Olasılık tabanlı ayrıştırıcı modelleri ile karşılaştırma yapılabilmesi amacıyla, bu bölümdeki ilk incelemelerde[18], biçimbilimsel bilgiler kullanılırken Bölüm 3.2'dekine benzer, ancak dinamik olmayan[19] bir indirgeme işlemi yürütülecektir. Buna göre bir *ÇK* ,

- Eğer isim türünden bir *ÇK* veya zaman ortacı olan bir sıfat *ÇK* ise, o zaman ana sözcük sınıfı, durum imi ve iyelik uyum imi ile birlikte ifade edilir.
- Diğer türden bir *ÇK* ise, sadece ana sözcük sınıfı ile belirtilir.

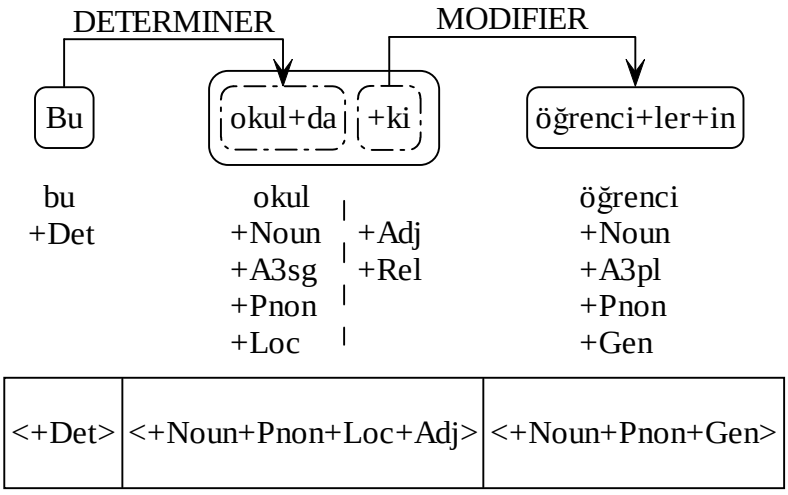
Bu akıl yürütmeyle, sınıflandırıcıya verilmek üzere bir örnek oluşturulurken CPOS bilgisi yukarıdaki şekilde hazırlanacaktır. Şekil 3.10 ve Şekil 3.11'de sözcük tabanlı ve *ÇK* tabanlı modeller gösterilmektedir. Bu şekillerde, birimler için kullanılacak CPOS bilgileri bağlılık grafiğinin en altında yer alan dikdörtgenler içerisinde yazılmıştır. Uydu birimler ile iye birimler arasındaki bağlılık türleri ise ilgili bağlılığı gösteren okların üzerinde yazılmıştır. Bağlılık etiketleri Tablo B.3'de verilen derlem gösteriminde kullanıldığı biçimde yazılmıştır. Bu örneklerde görülebileceği gibi, sözcük tabanlı modelde, iki *ÇK*'den oluşan "okuldaki" sözcüğünün CPOS özelliği "+Noun+Pnon+Loc+Adj"dir. *ÇK* tabanlı modellerde ise, iki ayrı birim olarak gösterilen bu sözcüğün ilk biriminin CPOS'u "+Noun+Pnon+Loc", ikincisininki de "Adj"dir.

Şekil 3.11'de "okuldaki" sözcüğünün iki *ÇK*'si arasındaki bağlılık türü "DERIV" ile belirtilmiştir. *ÇK tabanlı model* ile *ÇK tabanlı belirlenimci model* arasındaki fark bu türden bağlılıkların işleniş şekli ile ilgilidir. Sınıflandırıcı için eğitim ve sınama verisi hazırlanırken, bu türden bağlılıkların işlenmesi için iki farklı yol seçilebilir;

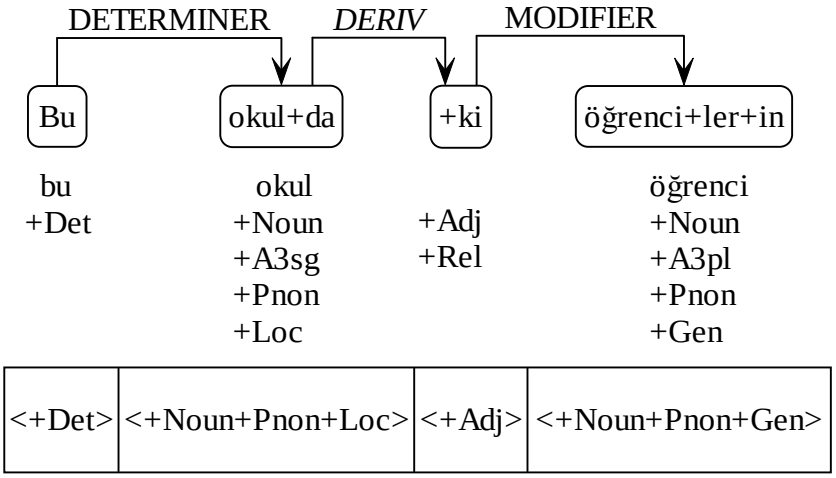
---

[18] Biçimbilimsel bilgilerin kullanımı ile ilgili ayrıntılı incelemeler Bölüm 4'de verilecektir.

[19] Bu modellerde, *ÇK*'lerin gösteriminde uzman bilgisi gerektiren dinamik seçim yöntemi bırakılarak, uydu ve iye birimler aynı bilgiler ile ifade edilmişlerdir. Bunun nedeni, daha önce de değinildiği gibi dinamik seçimin KDM'lere bırakılmasıdır.



Şekil 3.10: Sözcük Tabanlı Model



Şekil 3.11: *ÇK* Tabanlı Modeller

- Bunlardan birincisi, bu tür bağlılıkları da diğer türden bağlılıklar ile aynı şekilde işlemek ve her bağlılık için bir eğitim/sınama örneği hazırlamaktır.
- İkincisi ise bu tür bağlılıkların belirlenimci bir şekilde (doğrudan) bir sonraki *ÇK*'ye bağlanmasıdır. Bu durumda, bu tür bağlılıklar için eğitim sırasında eğitim örneği hazırlanmaz ve ayrıştırma sırasında KDM'ye başvurulmadan doğrudan bağlanırlar. Biçimbilimsel çözümleyicinin çıktısında sözcük içi bu bağlılıkların otomatik olarak gelmesi, böyle bir yaklaşımı mümkün kılmaktadır.

Yukarıda verilen bilgiler ışığında bir eğitim örneğinin nasıl olacağı aşağıda gösterilmektedir. Burada, *ÇK* tabanlı model (Şekil 3.11) kullanılırken eğitim kümesindeki "okuldaki" ve "öğrencilerin" sözcükleri arasındaki bağlığın bir eğitim örneği olarak nasıl hazırlandığı gösterilmektedir. Aşağıdaki gösterimde, anlaşılabilir olması açısından özelliklerin taşıdığı bilgiler doğrudan yazılmıştır. Esas olarak bu bilgiler KDM'ye ikilileştirilerek$^{\bowtie}$ (0/1) verilirler. *Denetimli öğrenme*$^{\bowtie}$ için hazırlanan örnek, Şekil 3.9'daki özellik kalıbında yer alan sekiz özellikten ve örneğin ait olduğu sınıf (ayrıştırıcı hareketi) bilgisinden oluşmaktadır. Hedef birimlerin (σ_0,τ_0), "okuldaki" sözcüğünün ikinci *ÇK*'si (σ_0) ve "öğrencilerin" (τ_0) olduğu durumda özellik kalıbına uygun olarak hazırlanan özellik vektörü şöyledir:

1. CPOS σ_0:	Adj
2. CPOS τ_0:	Noun+Pnon+Gen
3. CPOS σ_1:	-
4. CPOS $\sigma_0 + 1$:	Noun+Pnon+Gen
5. CPOS τ_1:	Adv
6. DEP $\ell(\sigma_0)$:	DERIV
7. DEP $r(\sigma_0)$:	-
8. DEP $\ell(\tau_0)$:	-

Ait olduğu sınıf: $U \xrightarrow{\text{MODIFIER}} I$

Burada yığının en üstünde Şekil 3.8'deki tümcenin 3 nolu *ÇK*'si, sırada ise 4 nolu *ÇK* bulunmaktadır. Ayrıştırmanın bu noktasına gelindiğinde 1 ve 2 nolu *ÇK*'lerin bağlılıkları kurulmuş olacağından yığında sadece bir eleman (σ_0) bulunmaktadır. Bu

nedenle özellik vektöründe CPOS σ_1, özelliği boştur. CPOS τ_1 sıradaki birimden sonra gelen birimdir (5 nolu *ÇK* "en"). Bu birimin CPOS'u ise Adv'dir. Bu aşamada oluşan ağaçta, üst birimin sağ uydusu yoktur. Sol uydusu ise 2 nolu *ÇK*'dir ve bu bağlılığın bağlılık türü (DEP $\ell(\sigma_0)$) DERIV'dir.

3.3.4 Biçimbilimsel Özelliklerin Kullanımı ile ilgili Modeller

Bu bölümde, biçimbilimsel özelliklerin kullanımı ile ilgili geliştirdiğimiz iki farklı model tanıtılacaktır. Bu modellerde, biçimbilimsel özellikler üzerinde indirgeme yaparak bunları ana sözcük sınıfı ile bir arada kullanmak yerine, sınıflandırıcıya verilecek vektör üzerinde ayrı bir özellik olarak kullanmak amaçlanmıştır. Böylece bu modellerde her *ÇK*'nin CPOS özelliği sadece ana sözcük sınıfını barındıracaktır. Biçimbilimsel özellikler (bknz. Şekil 3.8 sütun 6) ise INF olarak adlandırılan ayrı bir özellik türünde tutulacaklardır. Böylece ana sözcük sınıfları ve biçimbilimsel özellikler birbirlerinden bağımsız iki özellik olarak kullanılabileceklerdir. Bunlara ek olarak, ana sözcük sınıfından bağımsız şekilde, tüm biçimbilimsel özellikler herhangi bir indirgeme yapılmadan INF özelliğinde toplanacaklardır.

Şekil 3.8'de görüldüğü üzere, biçimbilimsel özellikler bir çok küçük özelliğin birleşiminden oluşabilmektedirler. Örneğin "öğrencilerin" sözcüğü bu gösterimde birbirlerinden birer dik çizgi ile ayrılmış ("A3pl|Pnon|Gen") üç biçimbilimsel özelliğe sahiptir. Bu yeni modeller kurulurken, iki farklı yol izlenmiştir. Bunlar:

1. Bu özellikleri bir arada tek bir özellik olarak kullanmak,
   "*ÇK tabanlı model (INF birleşik)*".
2. Bu özellikleri parçalara bölerek, her bir parçacığı ayrı bir özellik olarak kullanmak,
   "*ÇK tabanlı model (INF parçalı)*".

Önceki örnekte ele alınan bağlılığın bu modeller kullanıldığında oluşacak özellik vektörleri aşağıda gösterilmektedir. Kullanılan özellik kalıbına ek olarak, bu modellerde hedef birimler için INF özelliği eklenmiştir. Bu şartlar altında, *ÇK tabanlı model (INF birleşik)*'de bir eğitim örneği aşağıda gösterildiği gibidir.

1. CPOS σ_0: Adj
2. CPOS τ_0: Noun
3. INF σ_0: Rel
4. INF τ_0: A3pl|Pnon|Gen
5. CPOS σ_1: -
6. CPOS $\sigma_0 + 1$: Noun
7. CPOS τ_1: Adv
8. DEP $\ell(\sigma_0)$: DERIV
9. DEP $r(\sigma_0)$: -
10. DEP $\ell(\tau_0)$: -

Ait olduğu sınıf: $U \xrightarrow{\text{MODIFIER}} I$

Aynı örnek *ÇK tabanlı model (INF parçalı)*'da ise şöyledir:

1. CPOS σ_0: Adj
2. CPOS τ_0: Noun
3. INF σ_0: Rel
4. INF τ_0: A3pl
5. INF τ_0: Pnon
6. INF τ_0: Gen
7. CPOS σ_1: -
8. CPOS $\sigma_0 + 1$: Noun
9. CPOS τ_1: Adv
10. DEP $\ell(\sigma_0)$: DERIV
11. DEP $r(\sigma_0)$: -
12. DEP $\ell(\tau_0)$: -

Ait olduğu sınıf: $U \xrightarrow{\text{MODIFIER}} I$

3.3.5 Deney Sonuçları

Bu bölümde, sınıflandırıcı tabanlı ayrıştırıcı kullanılarak *Tüm Derlem* üzerinde elde edilen deney sonuçları verilmektedir. Bölüm 3.3.3 ve 3.3.4'de tanıtılan beş

ayrı model için deneyler hem görünüm bilgisi eklenmeden hem de eklenerek gerçekleştirilmişlerdir. Önceki bölümlerde ayrıntıları sunulan bu modeller şunlardır:

- Sözcük tabanlı model
- *ÇK* tabanlı model
- *ÇK* tabanlı belirlenimci model
- *ÇK* tabanlı (INF birleşik) model
- *ÇK* tabanlı (INF parçalı) model

Ayrıştırıcının sadece kesişmeyen bağlılıklara yönelik olmasından dolayı, eğitim verisi olarak derlemin kesişmeyen bağlılıklardan oluşan tümceleri kullanılmış ve sınama, *Tüm Derlem* üzerinde yapılmıştır. Daha ayrıntılı ifade etmek gerekirse, 10 katlı çapraz doğrulama sırasında derlem verisi rastgele 10 parçaya bölünmüş ve ayrıştırıcının her adımında bir küme sınama için kullanılırken, geri kalan dokuz kümenin sadece kesişmeyen bağlılıklardan oluşan tümceleri eğitim verisi olarak kullanılmıştır. Bu ayrıştırıcının amacı doğru bağlılıklar ile birlikte doğru bağlılık türlerini de bulmak olduğundan, *ÇKB* ve $ÇKB_E$ başarımları bir arada verilmiştir.

- $ÇKB_E$ (Çekim kümeleri arası etiketli başarım): Ayrıştırma birimlerinin doğru iye *ÇK*'ye doğru bağlılık türü ile bağlanma oranıdır.

Tablo 3.6 deney sonuçlarını vermektedir. Tablonun ilk üç satırı birim seçim modelleri ile ilgili sonuçları göstermektedir. Bu modellerde, görünüm bilgisi içermeyen sonuçlar eğitim ve sınama verileri için Şekil 3.9'daki özellik kalıbı kullanılarak, görünüm bilgisi içeren sonuçlar ise bu kalıba hedef birimler için görünüm bilgisinin eklenmesi ile oluşan özellik kalıbı (Şekil 3.12) kullanılarak elde edilmiştir. Sonuçlar farklı bir ayrıştırma yöntemi kullanılmasına rağmen, olasılık tabanlı ayrıştırıcının sonuçlarıyla benzer bulgular göstermektedir; *ÇK* tabanlı modeller genel olarak, sözcük tabanlı modelden daha yüksek başarım (yaklaşık %3 artış ile) vermektedirler. Ancak, *ÇK* tabanlı ayrıştırmanın tüm faydalarından yararlanabilmek için, görünüm bilgisi içermeyen modelde sözcük içi bağlılıkların belirlenimci bir şekilde işlenmesinin gerekliliği görülmektedir. (Görünüm bilgisi içermeyen *ÇK* tabanlı model ile görünüm bilgisi içermeyen *ÇK* tabanlı belirlenimci model arasında yaklaşık %2,5 fark vardır.)

Bunun nedeni, sınıflandırıcıların bu tür bağlılıkları görünüm bilgisi olmadan doğru olarak tahmin edememeleridir. Görünüm bilgisi içeren modelde ise, belirlenimci bir sözcük içi bağlılık işlemi eklemenin ayrıştırma başarımına herhangi bir etkisi olmadığı görülmektedir. Sınıflandırıcılar görünüm bilgisini kullanarak bu tür bağlılıkları çok kolay bir biçimde bulabilmektedirler.[20] Başarımın artmamasına karşın, belirlenimci yaklaşım eğitim örneği sayısını azaltarak KDM sınıflandırıcılarının eğitim ve sınama sürelerini kısaltmaktadır.

Tablo 3.6: Sınıflandırıcı Tabanlı Ayrıştırıcı Deney Sonuçları

Model	Görünüm Bilgisi Eklenmemiş		Görünüm Bilgisi Eklenmiş	
	ÇKB	$ÇKB_E$	*ÇKB*	$ÇKB_E$
Sözcük tabanlı	67,2±0,3	57,9±0,3	70,7±0,3	62,0±0,3
ÇK tabanlı	68,3±0,2	58,2±0,2	73,8±0,2	64,9±0,3
ÇK tabanlı belirlenimci	70,6±0,3	60,9±0,3	73,8±0,2	64,9±0,3
ÇK tabanlı (INF birleşik)	71,6±0,2	62,0±0,3	74,4±0,2	65,6±0,3
ÇK tabanlı (INF parçalı)	71,9±0,2	62,6±0,3	74,8±0,2	66,0±0,3
Eniyileştirilmiş			76,0±0,2	67,0±0,3

	σ_0	τ_0	σ_1	σ_0+1	τ_1	$\ell(\sigma_0)$	r(σ_0)	$\ell(\tau_0)$
CPOS	+	+	+	+	+			
DEP						+	+	+
INF								
LEMMA	+	+						

Şekil 3.12: Görünüm Bilgisi İçeren Özellik Kalıbı 1

	σ_0	τ_0	σ_1	σ_0+1	τ_1	$\ell(\sigma_0)$	r(σ_0)	$\ell(\tau_0)$
CPOS	+	+	+	+	+			
DEP						+	+	+
INF	+	+						
LEMMA								

Şekil 3.13: Özellik Kalıbı 2

[20]Sadece ilk *ÇK*'ler dolu bir LEMMA bilgisi içermekte, diğer *ÇK*'lerin LEMMA özelliği ise "_" olmaktadır.

	σ_0	τ_0	σ_1	σ_0+1	τ_1	$\ell(\sigma_0)$	r(σ_0)	$\ell(\tau_0)$
CPOS	+	+	+	+	+			
DEP						+	+	+
INF	+	+						
LEMMA	+	+						

Şekil 3.14: Görünüm Bilgisi İçeren Özellik Kalıbı 2

Tablo 3.6'nın dördüncü ve beşinci satırları *ÇK* tabanlı (INF birleşik) ve *ÇK* tabanlı (INF parçalı) modellerinin sonuçlarını vermektedir. Biçimbilimsel özelliklerin, özellik vektöründe ayrı bir özellik olarak gösterildikleri bu modeller sınanırken, Şekil 3.13 ve Şekil 3.14'deki görünüm bilgisi içermeyen ve içeren özellik kalıpları kullanılmıştır. Bu şekillerden görülebileceği gibi, özellik kalıplarında sadece hedef birimler (σ_0 ve τ_0) için INF özelliği eklenmiştir. Tablo 3.6'da görülebileceği gibi, her iki model de ayrıştırma başarımını tüm ölçütlerde %1'den fazla olmak üzere arttırmaktadır. Ancak, biçimbilimsel özellikleri parçalara bölerek her birini ayrı bir özellik olarak işlemenin, bu özellikleri birleşik halde kullanmaya göre hafif bir üstünlük sağladığı görülmektedir. (Aradaki fark, değer olarak küçük ancak tutarlıdır; görünüm bilgisi içeren modelin etiketli başarımında ortalama fark $0{,}4^{\pm 0{,}1}$'dir.)

Bu bilgilerin ışığında elde edilen en yüksek başarımlı model üzerinde yapılan özellik eniyileştirmesi sonucunda elde edilen başarım Tablo 3.6'nın en son satırında (Eniyileştirilmiş model) verilmektedir. Bu eniyileştirme işlemi sonucunda *ÇK*'lerin ana sözcük sınıfları yerine (CPOS), alt sözcük sınıflarını (POS) kullanmanın ve kuyrukta sıradaki birimden sonra gelen birim (τ_1) için LEMMA özelliği eklemenin en yüksek başarımı sağladığı görülmüştür. Bu modelde kullanılan özellik kalıbı Şekil 3.15'de verilmektedir. Bu eniyi model ile *Tüm Derlem* üzerinde *ÇKB* $= 76{,}0^{\pm 0{,}2}$ $\mathit{ÇKB}_E = 67{,}0^{\pm 0{,}3}$, *SB* $= 82{,}7^{\pm 0{,}5}$ ve *TB* $= 37{,}4^{\pm 0{,}6}$ elde edilmiştir.

Ayrıştırıcı Bölüm 3.2'de kullanılan *KsmSb Derlem* üzerinde eğitilip *KsmSb Derlem* üzerinde sınandığında *ÇKB* $= 78{,}3^{\pm 0{,}3}$, $\mathit{ÇKB}_E = 68{,}9^{\pm 0{,}2}$, *SB* $= 85{,}5^{\pm 1{,}0}$ ve *TB* $= 45{,}2^{\pm 1{,}1}$ elde edilmiştir. Hatırlanacağı gibi, olasılık tabanlı ayrıştırıcının bu veri üzerindeki başarımı *ÇKB* $= 73{,}5^{\pm 1{,}0}$, *SB* $= 81{,}2^{\pm 1{,}0}$ ve *TB* $= 38{,}7^{\pm 0{,}9}$'dir. Sınıflandırıcı tabanlı ayrıştırıcıda, aynı eniyileştirilmiş özellik modelini sadece LEMMA özelliklerini

	σ_0	τ_0	σ_1	σ_0+1	τ_1	$\ell(\sigma_0)$	r(σ_0)	$\ell(\tau_0)$
POS	+	+	+	+	+			
DEP						+	+	+
INF	+	+						
LEMMA	+	+			+			

Şekil 3.15: Özellik Kalıbı 3

çıkararak kullanırsak (Şekil 3.16), *ÇKB* = 76,1$\pm$0,3, $ÇKB_E$ = 65,9$\pm$0,4, *SB* = 82,8$\pm$1,2 ve *TB* = 42,1$\pm$0,9 elde edilmekte ve başarım bu durumda da olasılık tabanlı ayrıştırıcının başarımından yüksek olmaktadır. Bu durum, başarımdaki artışın sadece görünüm bilgilerinin kullanılmasına atfedilemeyeceğini göstermektedir.

	σ_0	τ_0	σ_1	σ_0+1	τ_1	$\ell(\sigma_0)$	r(σ_0)	$\ell(\tau_0)$
POS	+	+	+	+	+			
DEP						+	+	+
INF	+	+						
LEMMA								

Şekil 3.16: Görünüm Bilgisi İçermeyen Özellik Kalıbı 3

3.3.6 Kısım Sonucu

Bu bölümde, ayrıştırma işleminin sınıflandırıcılar kullanılarak yapıldığı veri güdümlü bir ayrıştırıcı tanıtılmıştır (Eryiğit ve diğ., 2006b). Geliştirilen ayrıştırıcı tüm derlem üzerinde sınanmış ve başarımı ortaya konmuştur. Önceki bölümde elde edilen çıkarımlara koşut olarak, bu ayrıştırıcıda da *ÇK* tabanlı modellerin daha yüksek başarım verdiği görülmüştür. Seyrek veri sorununun olasılık tabanlı ayrıştırıcıya göre daha az yaşandığı bu ayrıştırıcıda görünüm bilgileri de kullanılmış ve aynı zamanda bağlılık türleri de belirlenmeye çalışılmıştır. Bu bölümde, sınıflandırıcı tabanlı ayrıştırıcı genel olarak tanıtılmış ve eniyileştirme ile elde edilen başarımı en yüksek modeller sunulmuştur. Modeller geliştirilirken yapılan bazı seçimler ile ilgili ayrıntılı incelemeler Bölüm 4'de yapılacaktır. Ayrıştırıcı aynı zamanda *KsmSb Derlem*

üzerinde de sınanmış ve başarımın olasılık tabanlı ayrıştırıcıdan daha yüksek olduğu görülmüştür.

3.4 Bölüm Sonucu

Bu bölümde, Türkçe'ye özgü farklı tasarım modelleri geliştirilmiş ve veri güdümlü ayrıştırıcılar üzerinde etkilerinin incelenmesi hedeflenmiştir. Bu amaçla, farklı ayrıştırma yöntemlerinin Türkçe'ye uygun görülen bileşenleri bir araya getirilerek olasılık ve sınıflandırıcı tabanlı öğrenmeye dayalı iki farklı ayrıştırıcı düzenlenmiştir. Ayrıştırıcıların başarımları üzerinde ayrıntılı incelemeler ve karşılaştırmalar Bölüm 4'de verilmektedir. Türkçe'nin güçlü sağa bağımlı yapısına yönelik tasarlanmış olasılık tabanlı ayrıştırıcı, uzman bilgisi ile oluşturulmuş dinamik bir seçim yöntemi kullanmaktadır. Derlem boyutunun küçük olması nedeni ile seyrek veri sorunu yaşayan bu ayrıştırıcı üzerinde farklı seçim yöntemleri ve görünüm bilgisi kullanmanın etkileri Bölüm 4'de gösterilmektedir. Olasılık değerlerini hesaplamadan, farklı sınıfları ayırt etmeye dayanan sınıflandırıcı tabanlı ayrıştırıcı ise olasılık tabanlı ayrıştırıcıya oranla daha az seyrek veri problemi yaşamaktadır. Bu nedenle ve seçilen sınıflandırıcının yeteneklerinin de etkisiyle, görünüm bilgilerinin ve biçimbilimsel özelliklerin kullanımında olasılık tabanlı ayrıştırıcıya oranla rahatlık sağlamaktadır. Ayrıştırıcıların bu özellikleri ileriki bölümlerde ayrıntılı olarak incelenecektir. Bu bölümde yapılan incelemeler sonucunda, her iki ayrıştırıcı üzerinde de, ayrıştırma birimi olarak sözcüklerden daha küçük olan çekim kümelerinin kullanılmasının başarımı olumlu yönde etkilediği gösterilmiştir.

4. DEĞERLENDİRMELER VE TARTIŞMA

Bu bölüm, geliştirdiğimiz ayrıştırıcıların karşılaştırılmasını ve ürettikleri sonuçlar üzerinde yapılan ayrıntılı değerlendirmeleri kapsamaktadır. Bölümün giriş kısımında, sola bağımlı türde bağlılıkları ayrıştırmak üzere güncellenmiş kural tabanlı ayrıştırıcı tanıtılmakta ve olasılık tabanlı ayrıştırıcı üzerinde yapılan iyileştirmeler anlatılmaktadır. Bunlara ek olarak, ileri sürdüğümüz dinamik seçim yönteminin etkinliğini göstermek üzere yapılan deneylerin sonuçları verilmektedir. Sonraki kısımlarda sırasıyla, ayrıştırıcıların eniyileştirilmiş başarımları sunularak karşılaştırılmakta, biçimbilimsel özellikleri ve görünüm bilgilerini kullanmanın ayrıştırma başarımı üzerindeki etkileri ayrıntılı olarak değerlendirilmekte, eğitim kümesi boyutunun etkileri irdelenmekte, farklı bağlılık türlerine, yönlerine ve tümce uzunluklarına göre hata incelemeleri yapılmakta, yetkin etiket kullanımının başarıma etkileri değerlendirilmektedir. Bölümün sonunda, aynı derlem üzerinde sınanmış diğer çalışmalar ile bu çalışmada geliştirilen en iyi ayrıştırıcının başarımı karşılaştırılmıştır.

4.1 İyileştirmeler ve Dinamik Seçim Yönteminin Etkinliği

Bu kısımda sırasıyla, kural tabanlı ayrıştırıcı ve olasılık tabanlı ayrıştırıcı üzerinde yapılan iyileştirmeler ve olasılık tabanlı ayrıştırıcıda kullanılan dinamik seçim yönteminin etkinliği gösterilecektir.

Sınıflandırıcı tabanlı ayrıştırıcının geliştirilmesi sırasında kullanılan eğitim ve sınama verisinde herhangi bir iye birime bağlı olmayan noktalama işaretlerini, hemen sağ taraflarında yer alan birime bağlamanın başarımı arttırdığı görülmüştür. Başarım ölçümünde bu türde birimlerden çıkan bağlılıkların[1] gözardı edilmesi nedeni ile, bu artışın doğrudan bir etki sonucunda oluşmadığı anlaşılmaktadır. Bu işlemin başarıma

[1] Bu tür bağlılıklar özel bir bağlılık türü ile gösterilmişlerdir.

dolaylı etkisi şu şekildedir: Ayrıştırma algoritmasının işleyişi sırasında yığında üst birimin altında kalan birimlerin, üst birimin noktalama işareti olup hiçbir yere bağlanamaması sonucunda yığında yığılı kalmaları engellenmektedir. Bu bölümde, yapılan deneylerde, sınıflandırıcı tabanlı ayrıştırıcının başarımına olumlu etkisi olan bu değişiklik, kurallar ve ayrıştırma algoritması üzerinde değişiklikler yapılarak kural tabanlı ve olasılık tabanlı ayrıştırıcılara uygulanmıştır. Ayrıca kural tabanlı ayrıştırıcı sola bağımlı bağlılıkları da işleyecek şekilde iyileştirilmiş ve kurallar *ÇK* temelli hazırlanmıştır. Buna ek olarak, önceki bölümde tanıtılan *ÇK* tabanlı modeller üzerinde yapılan incelemeler sonucunda, olasılık tabanlı ayrıştırıcı için yeni bir *ÇK* tabanlı model oluşturulmuş ve başarımı sunulmuştur.

4.1.1 Kural Tabanlı Ayrıştırıcı

Bölüm 3.2'deki temel modellerimizden biri olan kural tabanlı ayrıştırıcı, *Tüm Derlem* üzerinde çalışabilmek, bir diğer deyişle sola bağımlı türde bağlılıkları da işleyebilmek üzere güncellenmiştir. Böylelikle, bu yeni ayrıştırıcı hem olasılık tabanlı model (*KsmSb Derlem* üzerinde) hem de sınıflandırıcı tabanlı ayrıştırıcı (*KsmSb Derlem* ve *Tüm Derlem* üzerinde) için temel model olarak kullanılabilecektir. Yeni kural tabanlı ayrıştırıcı (Eryiğit ve diğ., 2006a), öncekinden farklı olarak, ayrıştırma birimi olarak *ÇK*'leri kullanacak ve kurallarını sadece *ÇK*'lere bakarak belirleyecektir. Kullanılan kurallar (*Tüm Derlem* üzerindeki başarımları ve toplam uygulanış sayıları ile beraber) Ek A'da ayrıntılı olarak verilmektedir. Noktalama işaretlerinin ayrıştırma algoritmasına getirdiği sorunu aşmak üzere "Sözcükler hemen sağ taraflarındaki noktalama işaretine bağlanırlar" kuralı eklenmiştir. Bu kuralın başarımı çok düşük olmasına karşın (bknz Tablo A.1 devam), yukarıda anlatılan ayrıştırma algoritmasının işleyişine getirdiği çözüm ile genel başarımda artışa neden olmaktadır.

Ayrıştırma algoritması olarak Bölüm 3.1'de tanıtılan kural tabanlı ayrıştırıcıda kullanılan algoritma sola bağımlı türde bağlılıkları işleyecek şekilde değiştirilmiştir. Bu işlem gerçekleştirilirken, algoritmaya Türkçe'ye özel değişiklikler getirilmiştir. Bu açıdan, yine sola bağımlı türde bağlılıkları işleyebilen Nivre (2003)'nin asıl algoritmasından farklılık göstermektedir. Algoritmanın işleyişi şöyledir:

```
Kuyrukta bekleyen ÇK olduğu sürece tekrarla{
        eğer Yığın boş ise
           Ötele(Yığın)
        değil ise
          hareket=Ayrıştırma_Modeli(i,j)
          eğer hareket==Ötele ise
            Ötele(Yığın)
          eğer hareket==U → I ise
            Bağlılık_Kur(i → j)
            Çek(Yığın)
          eğer hareket==S ← B ise
            Bağlılık_Kur(j → i)
            Kuyrukta bekleyen sıradaki ÇK'ye ilerle
    }
```

Bu ayrıştırıcıda, Bölüm 3.3'de tanıtılan olağan yay yönteminden farklı olarak, ayrıştırma modeli, j. elemanı yığının en üstünde olmayan bir h. elemanına bağlama kararı da verebilir. Bu durumda ayrıştırıcıya basit "$S \leftarrow B$" hareketini bildirmeden önce kendi içerisinde "$S \leftarrow^* B$" işlemi yürütür. Bu işlem yığında h. elemanın üzerinde yer alan tüm elemanları h. elemanın uydusu haline getirir ve bu elemanları yığından çeker. Böylece işlem sonunda h. eleman yığındaki en üst eleman haline gelir. Bundan sonra basit "$S \leftarrow B$" hareketi bildirilir. Bir başka deyişle, bu hareket $h \overset{S \leftarrow B}{\longleftarrow} j$ bağlılığına karar verildiğinde (h<x<j) h. ve j. eleman arasında yer alan tüm x sıra numaralı elemanları h. elemana bağlamak anlamındadır.

Model ilk olarak, sözcükten ayrı yazılan ve çekim eklerinden dolayı oluşan sola bağımlı türde bağlılıkları bulmaya çalışır. Bu durumda eğer j. eleman bir vurgulayıcı (de,da), soru eki (mi,mu,mısın vb...), ilişkilendirici (ki) veya olumsuzluk (değil) belirten bir ek (Oflazer ve diğ., 2003) ise "$S \leftarrow B$" hareketi bildirilir.

Ayrı yazılan çekim eklerinin denetimi yapıldıktan sonra, ayrıştırıcı insan tarafından yazılmış 26 adet kuralı (Tablo A.1) kullanarak i ve j sıra numaralı birimler arasında "$U \rightarrow I$" ilişkisi kurmaya çalışır. Bu kurallar, "sıfat yanındaki isime bağlanır",

"zarf yanındaki eyleme bağlanır" türünde kurallardır. Birden fazla *ÇK*'ye sahip olan sözcükler içerisinde yer alan *ÇK*'lerin, aynı sözcük içerisinde sağ taraflarında yer alan ilk *ÇK*'ye bağlandıkları varsayılır. Bu durumda ayrıştırma modeli "$B \rightarrow S$" hareketini bildirir. Kurulan bu tür bağlılıklar başarım ölçümünde değerlendirilmezler.

i. ve j. sözcüklerinin "$U \rightarrow I$" kurallarından hiçbirine uymaması durumunda eğer j. eleman isim soylu ise ve bu elemandan sonra bir noktalama işareti geliyorsa, ayrıştırma modeli sola bağımlı türde bir bağlılık bulmaya çalışır. Derlemde yer alan sağa bağımlı türdeki bağlılıkların %83'ü bu türde bağlılıklardır. Bu durumda, yığının en üstündeki elemandan başlanarak eylem soylu bir eleman bulunmaya çalışılır. j. elemanın yığın içerisinde yer alan ilk eylem soylu h. elemana bağlanmasına karar verilir. Eğer h. eleman yığının en üstteki elemanı değilse bu durumda bu elemanın üzerindeki birimlerin yığından çekilerek bir yere bağlanmaları gerekir. Bu durum kesişmeyen bağlılık ilkesinin gereğidir. Ayrıştırma algoritması, h. eleman yığının en üstüne gelene dek, bu elemanları h. iye elemanına bağlar. Ve daha sonra ayrıştırma modeli "$S \leftarrow B$" hareketine karar verir.

Yukarıda belirtilen hareketlerden hiçbirinin bulunamaması durumunda model "ötele" hareketine karar verir. Ayrıştırmanın tamamlanabilmesi için tümce içerisindeki kök sözcük (derlemde genelde en sonda yer alan noktalama işareti) hariç tüm sözcüklerin bir iye sözcüğe bağlanmaları gerekmektedir. Ayrıştırma sonucunda kuyrukta bekleyen hiçbir eleman kalmamasına rağmen yığında birden fazla eleman bulunuyorsa, bu elemanlar yığının en üstündeki elemana bağlanırlar.

Yukarıda anlatılan bu değişiklikler ile *KsmSb Derlem* üzerindeki *ÇKB* başarımı %70,5 olan kural tabanlı ayrıştırıcının başarımı aynı veri kümesinde %73,4'e çıkmıştır. Bu sonuç yaptığımız değişikliklerin olumlu yönde katkı yaptığını göstermektedir.

4.1.2 Olasılık Tabanlı Ayrıştırıcı

Hatırlanacağı gibi Bölüm 3.2'de tanıtılan olasılık tabanlı ayrıştırıcının ayrıştırma algoritması birimlerin hepsini aynı şekilde, bağlanma olasılıklarından faydalanarak işlemektedir. Noktalama işaretleri ile ilgili bağlılıkların incelenmesi sonucunda elde edilen bulgular ışığında, algoritma noktalama işareti görüldüğünde bu birimi doğrudan

sağındaki birime bağlayacak şekilde değiştirilmiştir. Bu yaklaşım başarımda önemli bir artışa neden olmamasına karşın, ilk ayrıştırıcıda karşılaşılan çok geniş standart hata aralıklarını daraltmaktadır. Bu işlem sonucunda sınıflandırıcı tabanlı ayrıştırıcı ile olasılık tabanlı ayrıştırıcının hata aralıklarının aynı mertebelerde olduğu gözlenmiştir.

Ayrıca daha ayrıntılı yapılan incelemelerde *ÇK* tabanlı model 2'nin *ÇK* tabanlı model 1'e oranla sağladığı üstünlüğün (Tablo 3.2) kullandığı uzaklık fonksiyonundan kaynaklandığı belirlenmiş, bağlam içerisindeki komşuların işlenişine getirdiği değişikliğin başarımı, uzaklık fonksiyonu kadar etkilemediği görülmüştür. Bu bölümde kullanılacak *ÇK* tabanlı model Bölüm 3.2'deki *ÇK* tabanlı model 1 ve *ÇK* tabanlı model 2'nin karışımıdır. Bu modelde tüm birimler (iye, uydu, komşu) çekim kümeleri olarak kullanılacak ancak uzaklık fonksiyonu hesaplanırken iki birim arasında kalan sözcük sınırları sayılacaktır.

ÇK'lerin gösteriminde önceki ayrıştırıcı ile aynı dinamik seçim yöntemi (Bknz. Bölüm 3.2.2) kullanılmıştır. Ancak sınıflandırıcı tabanlı ayrıştırıcının eniyileştirilmesi sırasında gözlemlenen alt sınıf bilgilerinin kullanımının ana sınıf bilgilerinin kullanılmasından daha yüksek başarım sağlamasından yola çıkarak, alt sınıf bilgileri kullanılmıştır.[2]

Tablo 4.1'de her model için ayrı ayrı yapılan eniyileme sonucunda en iyi başarım sağlayan parametreler ve başarımlar verilmiştir. Bu yeni ayrıştırıcı ile eşik değeri[3] (k) ve demet boyutu (d) için yapılan eniyilemede $k = 6$ ve $d = 3$ olarak elde edilmiştir. Tablodan görülebileceği gibi bu ayrıştırıcı ile elde edilen en yüksek *ÇKB* başarımı 74,9$\pm$0,3'dur. Hatırlanacağı gibi, aynı veri üzerinde (*KsmSb Derlem*) elde edilen önceki değer 73,5$\pm$1,0'dir. Yaptığımız değişikliklerin getirdiği iyileşme gözlemlenmiştir.

[2]Bu değişiklik *ÇK* tabanlı modelin başarımını arttırırken, sözcük tabanlı modellerin başarımını düşürmektedir. Bunun nedeni *ÇK* tabanlı modellere alt sınıf bilgilerini kullanarak ayrıştırma için daha küçük sınıflara ayrılmış olasılıklar hazırlarken, *ÇK*'lerin birleşimini kullanan sözcük tabanlı modellerde seyrek veri sorununu daha da arttırmasıdır. Tüm ayrıştırıcılar arasında uyumluluk olması açısından buradaki sözcük tabanlı modellerde de alt sınıf bilgisi kullanılmıştır. Ana sözcük sınıfı kullanılması halinde Tablo 4.1'deki değerler sözcük tabanlı model 1 için *ÇKB* = 72,0$\pm$0,4 ve sözcük tabanlı model 2 için *ÇKB* = 72,4$\pm$0,4 olacaktır.

[3]Daha önce de belirtildiği gibi, uzaklık fonksiyonu iye ve uydu birimler arasındaki sözcük sınırlarına bağlı olarak hesaplanmaktadır. Derlemde, bağlılıkların %95'i 6'dan daha yakın uzaklıkta bir sözcüğe bağlanmaktadırlar.

Tablo 4.1: Olasılık Tabanlı Ayrıştırıcının Başarımları

Model *(parametreler)*	ÇKB (*KsmSb Derlem*)
Sözcük tabanlı model 1 *(D_l=1, D_r=1, H_l=1, H_r=1)*	71,5$\pm$0,5
Sözcük tabanlı model 2 *(D_l=1, D_r=1, H_l=1, H_r=1)*	72,0$\pm$0,4
ÇK tabanlı model *(D_l=1, D_r=1, H_l=0, H_r=1)*	74,9$\pm$0,3

4.1.3 Dinamik Seçim Yönteminin Etkinliği

Olasılık tabanlı ayrıştırıcıda kullanılan dinamik seçim yöntemi şöyledir:

- *ÇK* bir uydu olarak kullanıldığında,
  - – Eğer isim türünden bir *ÇK* ise, o zaman sadece durum imi ile belirtilir.
  - – Diğer türden *ÇK*'ler, sadece alt sözcük sınıfları ile belirtilirler.
- *ÇK* bir iye olarak kullanıldığında,
  - – Eğer isim türünden bir *ÇK* veya zaman ortacı olan bir sıfat *ÇK* ise, o zaman alt sözcük sınıfı ve iyelik uyum imi ile birlikte ifade edilir.
  - – Diğer türden *ÇK*'ler, sadece alt sözcük sınıfları ile belirtilirler.

Bu dinamik seçim yöntemi yerine, farklı seçimler yapmanın etkileri aşağıda incelenmektedir. Burada, altı farklı seçim yöntemi denenmiştir. Bunlar şöyledir:

- *Seçim 1* = Alt sözcük sınıfı yerine ana sözcük sınıf bilgisini kullanmak
- *Seçim 2* = Hiç biçimbilimsel özellik kullanmamak (sadece alt sözcük sınıfı kullanmak)
- *Seçim 3* = Biçimbilimsel özelliklerin kullanımında dinamik seçim yapmamak (tüm türler için alt sözcük sınıfını ve buna ek olarak isim türünden olanlar için durum ve iyelik imlerini, zaman ortacı bulunan sıfatlar için de iyelik imlerini kullanmak)
- *Seçim 4* = Alt sözcük sınıfı ile beraber tüm biçimbilimsel özellikleri kullanmak
- *Seçim 5* = Her türden *ÇK* için LEMMA özelliğini de kullanmak
- *Seçim 6* = Her türden *ÇK* için LEX özelliğini de kullanmak

Tablo 4.2 asıl seçim yöntemi yerine bu seçimler kullanıldığında elde edilen başarımları vermektedir. Tabloya bakıldığında, asıl dinamik seçim dışında kalan diğer seçimlerin hepsinin başarımı düşürdüğü görülmektedir. Ana sözcük sınıfları yerine (*Seçim 1*) alt sözcük sınıflarını (*asıl*) kullanmanın istatistiksel olarak belirgin olmasa bile başarımı arttırdığı görülmektedir. Sözcüğün tümünü kullanmanın (*Seçim 6*) gövdeyi kullanmaya (*Seçim 5*) göre daha başarılı olduğu görülse de her iki görünüm bilgisi ekleme yönteminde de asıl modele göre başarımlarında çok önemli bir düşüş gözlemlenmektedir. *Seçim 2* ve *Seçim 3*'ün sonuçlarına bakıldığında, bu tür bir dinamik seçim yöntemi kullanmanın faydaları görülmektedir. *Seçim 4*'ün sonuçları ise tüm biçimbilimsel özellikleri (herhangi bir indirgeme yapmadan) kullanmanın ne kadar belirgin bir düşüşe neden olduğunu göstermektedir.

Tablo 4.2: *ÇK*'lerin Gösteriminde Farklı Seçimlerin Sonuçları

ÇK tabanlı model *(D_l=1, D_r=1, H_l=0, H_r=1)*	*ÇKB* (*KsmSb Derlem*)
asıl	74,9±0,3
Seçim 1	74,6±0,3
Seçim 2	72,2±0,4
Seçim 3	73,0±0,4
Seçim 4	50,4±0,5
Seçim 5	57,8±0,4
Seçim 6	62,3±0,4

4.2 Ayrıştırıcıların Başarımları

Bölüm 3.3'de sınıflandırıcı tabanlı ayrıştırıcının *Tüm Derlem* üzerindeki başarımları verilmektedir. Olasılık tabanlı ayrıştırıcının başarımı ile doğrudan bir karşılaştırma yapabilmek üzere Tablo 4.3 ve Tablo 4.4'de sınıflandırıcı tabanlı ayrıştırıcı için geliştirdiğimiz bazı ana modellerin hem *KsmSb Derlem* hem de *Tüm Derlem* üzerindeki başarımları gösterilmektedir.

Tablo 4.3[4]'de Tablo 3.6'da sonuçları verilen *sözcük tabanlı* ve ÇK *tabanlı (belirlenimci)* modellerin *KsmSb Derlem* üzerindeki başarımları verilmiştir. Bu tablodaki *KsmSb Derlem* üzerindeki *ÇKB* başarımları olasılık tabanlı ayrıştırıcının (Tablo 4.1) başarımları ile karşılaştırıldığında, görünüm bilgisi eklenmemiş ve olasılık tabanlı ayrıştırıcıya benzer *ÇK* gösterimleri kullanan[5] sınıflandırıcı tabanlı ayrıştırıcının, olasılık tabanlı ayrıştırıcıya oranla daha başarılı olmadığı görülmektedir. Aksine sözcük tabanlı modelde istatistiksel olarak belirgin bir düşüş kaydedilmektedir ($71{,}5^{\pm 0{,}5} \rightarrow 70{,}5^{\pm 0{,}5}$). Bu düşüşün nedeni, sınıflandırıcı tabanlı ayrıştırıcıda olasılık tabanlı ayrıştırıcıda olduğu gibi biçimbilimsel özelliklerin seçiminde dinamik bir seçim yöntemi kullanılmaması olarak düşünülebilir.

Tablo 4.3: Sınıflandırıcı Tabanlı Ayrıştırıcı Görünüm Bilgisi Eklenmemiş Modeller

Model	*KsmSb Derlem*		*Tüm Derlem*	
	ÇKB	ÇKB_E	ÇKB	ÇKB_E
Sözcük tabanlı	$70{,}5^{\pm 0{,}5}$	$60{,}7^{\pm 0{,}5}$	$67{,}2^{\pm 0{,}3}$	$57{,}9^{\pm 0{,}3}$
ÇK tabanlı (belirlenimci)	$74{,}6^{\pm 0{,}3}$	$64{,}2^{\pm 0{,}4}$	$70{,}6^{\pm 0{,}2}$	$60{,}9^{\pm 0{,}3}$

Tablo 4.4'de ise sınıflandırıcı tabanlı ayrıştırıcının en gelişmiş modeli olan *ÇK* tabanlı (INF parçalı) modelinin görünüm bilgisi eklenmiş ve eklenmemiş olarak *Tüm Derlem* ve *KsmSb Derlem* üzerinde başarımları verilmektedir. Tablonun ilk satırındaki *KsmSb Derlem ÇKB* başarımı Tablo 4.1'deki *ÇK* tabanlı modelin *ÇKB* başarımından istatistiksel olarak belirgin halde daha yüksektir. Bu durum modelimizin görünüm bilgileri kullanılmasa bile yüksek başarım elde ettiğini göstermektedir. Hatırlanacağı gibi, bu modelde biçimbilimsel özelliklerin tümü ayrı ve parçalı olarak kullanılmaktadırlar. İlerideki bölümlerde, biçimbilimsel özelliklerin kademeli olarak kullanılmasının etkileri incelenecektir. Daha sonra, yapılacak bu incelemeden de görüleceği gibi, bu modelin tüm biçimbilimsel özellikler yerine sadece olasılık tabanlı ayrıştırıcının kullandığı biçimbilimsel özellikleri kullanması durumunda da başarımı daha yüksek olmaktadır.

[4]Son iki sütunda yer alan sonuçlar Tablo 3.6'den alınmıştır.

[5]Burada bahsedilen, Bölüm 3.3.3'de ayrıntılı olarak anlatılan, biçimbilimsel özelliklerin sözcük sınıf bilgisi ile beraber tek özellik olarak kullanıldığı modeldir.

Tablonun son iki satırında, görünüm bilgisi ekleme işlemi sırasında sözcüğün tümce içerisinde geçen halini (LEX) kullanmak ile sözcüğün gövdesini (LEMMA) kullanmak arasındaki fark görülmektedir. LEMMA bilgisini kullanmak istatistiksel olarak belirgin olmasa da daha yüksek sonuç vermektedir. Bunun nedeni sözcüğün gövdesinden sonra gelen eklerin zaten biçimbilimsel özellikler ile ifade ediliyor olmasıdır. Sözcüğün bütününü bir özellik olarak kullanmak aynı bilgiyi iki kez kullanmaya çalışmak olarak görülebilir. Aynı zamanda da, ardına birçok ek alarak farklı şekillerde görünen aynı gövdeye sahip sözcükler gerekli olmadığı halde seyrek veri sorununu arttırmaktadırlar.

Tablo 4.4: Sınıflandırıcı Tabanlı Ayrıştırıcı *ÇK* Tabanlı Modeller

ÇK *tabanlı model (INFparçalı)*	KsmSb Derlem		*Tüm Derlem*	
	ÇKB	$ÇKB_E$	ÇKB	$ÇKB_E$
Görünüm bilgisi eklenmemiş	76,1±0,3	65,9±0,4	72,4±0,2	63,1±0,3
Görünüm bilgisi eklenmiş (LEX ile)	78,0±0,4	68,3±0,3	75,7±0,2	66,6±0,3
Görünüm bilgisi eklenmiş (LEMMA ile)	78,3±0,3	68,9±0,2	76,0±0,2	67,0±0,3

Tablo 4.5, Tablo 4.6 ve Tablo 4.7 temel, olasılık tabanlı ve sınıflandırıcı tabanlı ayrıştırıcıların *ÇKB*, *SB* ve *TB* başarımlarını özetlemektedir. Görülebileceği gibi, her iki veri güdümlü ayrıştırıcı da dayanak modellerinin *ÇKB* başarımlarını geçmektedirler.[6]

Tablo 4.5: Ayrıştırıcıların *ÇKB* Başarımları

Ayrıştırıcı	*ÇKB*	
	KsmSb Derlem	*Tüm Derlem*
Yandakine bağlan (ilk *ÇK*)	63,9	56,0
Yandakine bağlan (son *ÇK*)	62,2	54,1
Kural tabanlı	73,4	70,5
Olasılık tabanlı	74,9±0,3	-
Sınıflandırıcı tabanlı	78,3±0,3	76,0±0,2

[6] *SB* başarımında olasılık tabanlı ayrıştırıcının başarımı istatistiksel olarak belirgin olmasa da kural tabanlı ayrıştırıcının hafifçe gerisinde kalmıştır. *TB* başarımında ise bu fark daha fazladır.

Tablo 4.6: Ayrıştırıcıların *SB* Başarımları

Ayrıştırıcı	*SB*	
	KsmSb Derlem	*Tüm Derlem*
Yandakine bağlan (ilk *ÇK*)	72,1	63,3
Yandakine bağlan (son *ÇK*)	72,1	63,3
Kural tabanlı	82,7	79,3
Olasılık tabanlı	82,2$\pm$0,8	-
Sınıflandırıcı tabanlı	85,1$\pm$1,0	82,7$\pm$0,5

Tablo 4.7: Ayrıştırıcıların *TB* Başarımları

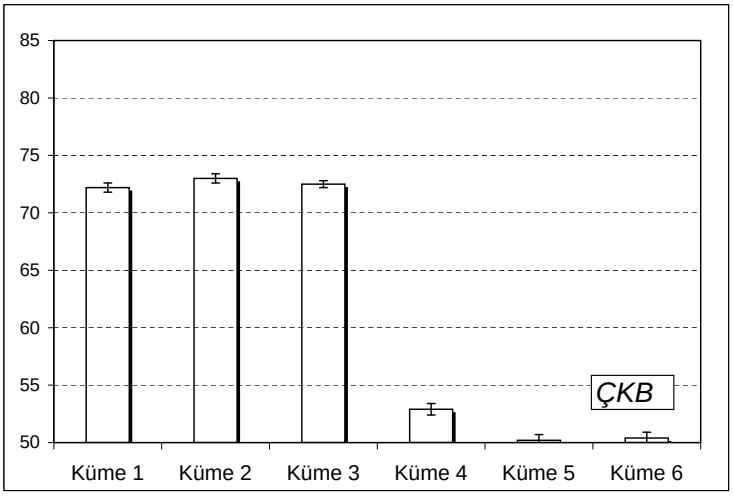
Ayrıştırıcı	*TB*	
	KsmSb Derlem	*Tüm Derlem*
Yandakine bağlan (ilk *ÇK*)	24,0	14,3
Yandakine bağlan (son *ÇK*)	22,6	13,4
Kural tabanlı	40,4	31,4
Olasılık tabanlı	38,4$\pm$1,2	-
Sınıflandırıcı tabanlı	45,2$\pm$1,1	37,4$\pm$0,6

4.3 Biçimbilimsel Özellikleri Kullanmanın Etkisi

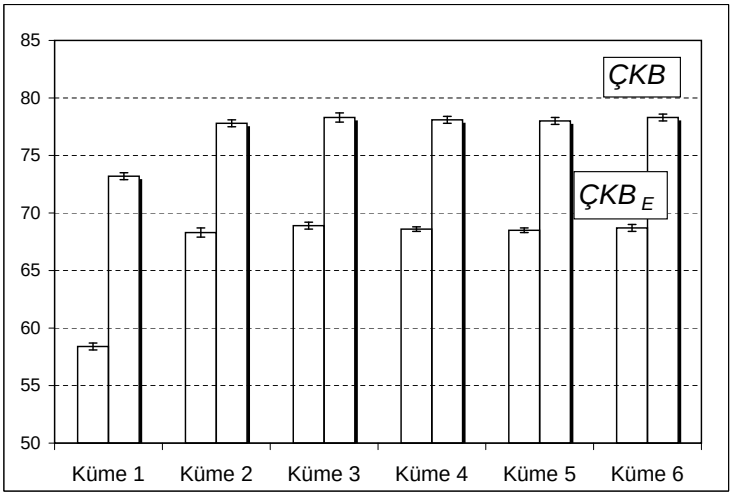
Önceki bölümlerde, ayrıştırıcılar, eniyileştirilmiş parametreleri ve özellik gösterimleri ile birlikte tanıtılmışlardır. Derlem tarafından sağlanan tüm biçimbilimsel özelliklerin özellik olarak kullanılmasının sınıflandırıcı tabanlı ayrıştırıcının başarımını çok belirgin bir şekilde arttırdığı, ancak olasılık tabanlı ayrıştırıcının başarımı üzerinde azaltıcı bir etkisi olduğu görülmüştür. Bu bölümde, biçimbilimsel özellikler kullanmanın etkisini daha ayrıntılı görebilmek üzere, değişik türde biçimbilimsel özellikler kullanan altı farklı küme hazırlanmış ve her iki ayrıştırıcının da en yüksek başarım sağlayan modelleri üzerinde sınanmıştır.

Bu bölümde ve Bölüm 4.4'de KDM eğitimi sırasında eğitim kümesi daha küçük kümelere bölündüğünden sınıflandırıcı tabanlı ayrıştırıcı ile yapılan deneylerin sonuçları daha önceki sonuçlarla birebir karşılaştırılabilir değillerdir. Bu yaklaşım KDM eğitim sürelerini azaltırken başarımda da çok belirgin bir düşüşe neden

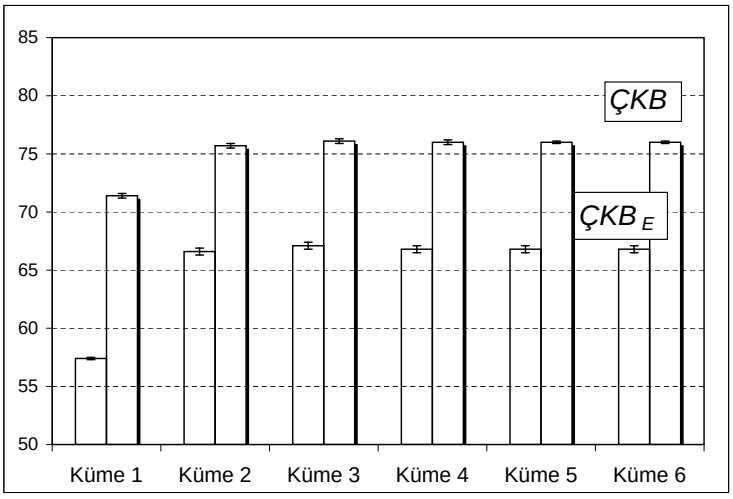
a) *KsmSb Derlem* üzerinde Olasılık tabanlı Ayrıştırıcı *ÇKB*



b) *KsmSb Derlem* üzerinde Sınıflandırıcı Tabanlı Ayrıştırıcı *ÇKB* ve $\textit{ÇKB}_E$



c) *Tüm Derlem* üzerinde Sınıflandırıcı Tabanlı Ayrıştırıcı *ÇKB* ve $\textit{ÇKB}_E$



Şekil 4.1: Biçimbilimsel Özellik Kümeleri 1–6

olmamaktadır. Olasılık tabanlı model için ise, bu bölümde farklı biçimbilimsel özelliklerin kullanımı incelenirken dinamik seçim yöntemi kullanılmamaktadır.

Aşağıda listelenen kümelerden her biri bir önceki kümeyi ve buna eklenen bazı biçimbilimsel özellikleri barındırır. Küme listesi şöyledir:

Küme 1 = Hiç biçimbilimsel özellik yok

Küme 2 = İsim türünden veya zaman ortaçlı sıfat olanlar için durum ve iyelik imi

Küme 3 = Küme 2 + İsim türünden veya eylem olanlar için kişi/sayı uyum imi

Küme 4 = Küme 3 + İsim türünden olanlar için tüm biçimbilimsel özellikler

Küme 5 = Küme 4 + Eylem olanlar için tüm biçimbilimsel özellikler

Küme 6 = Küme 5 + Tüm biçimbilimsel özellikler

Şekil 4.1 hem olasılık tabanlı (Şekil 4.1-a) hem de sınıflandırıcı tabanlı (Şekil 4.1-b, Şekil 4.1-c) ayrıştırıcıların sonuçlarını göstermektedir;

- Şekil 4.1-b ve Şekil 4.1-c, Bölüm 3.2.2'de durum ve iyelik iminin (Küme 2) seçimi için kullanılan uzman bilgisini doğrular niteliktedir.
- Buna ek olarak, isim türünden olanlarda ve eylemlerde var olan kişi/sayı uyumluluk imlerinin de (Küme 3), başarımda istatistiksel olarak belirgin bir artış sağlamasalar da (Şekil 4.1-c'deki *ÇKB* skoru hariç), önemli biçimbilimsel özellikler olduğu gözlemlenmektedir.
- Dikkat çeken bir diğer nokta, $ÇKB_E$ başarımlarının, biçimbilimsel özelliklerin kullanımından, *ÇKB* başarımlarına göre daha çok etkilenmeleridir. Küme 1 ve Küme 2 (Şekil 4.1-b ve Şekil 4.1-c) arasındaki fark *ÇKB* için yaklaşık %4 iken $ÇKB_E$ için yaklaşık %10'dur. Bu durum biçimbilimsel özelliklerin özellikle uydu ve iye birimler arasındaki bağlılığın türünü belirlemek açısından önemli olduğunu göstermektedir. Türkçe'de öğelerin görevlerini sözcük dizilişlerinin değil biçimbilimsel özelliklerin (özellikle durum imlerinin) belirlemesi bu sonucu anlamlı kılmaktadır.
- Yine bu şekillerden, sınıflandırıcı tabanlı ayrıştırıcının, derlem tarafından sağlanan tüm biçimbilimsel özellikler kullanılsa (Küme 6) bile seyrek veri

sorunundan çok fazla etkilenmediği görülmektedir. Öte yandan, olasılık tabanlı ayrıştırıcının Küme 3 ile bile bu sorunu yaşadığı ve başarımın düşmeye başladığı görülmektedir (bknz Şekil 4.1-a). Bu düşüş, tüm biçimbilimsel özellikler kullanıldığında daha da artmaktadır.

4.4 Görünüm Bilgilerini Kullanmanın Etkisi

Bu bölümde ilk olarak, görünüm bilgisi eklemenin etkisini daha iyi inceleyebilmek üzere, farklı ana sözcük sınıflarından olan *ÇK*'lere kısmi olarak görünüm bilgisi ekleme işlemi yapılmıştır. Bu inceleme, gerekli olduğu yerlerde alt sözcük sınıfları için de genişletilmiştir.

Tablo 4.8: Kısmi Görünüm Bilgisi Eklemenin Etkisi
(n=görülme sıklığı, d=farklı gövdelerde görülme sıklığı)

			Olasılık Tab.	Sınıflandırıcı Tab.			
			KsmSb Derlem	*KsmSb Derlem*		*Tüm Derlem*	
	n	*d*	ÇKB	ÇKB	$ÇKB_E$	ÇKB	$ÇKB_E$
Yok	-	-	*74,9*±0,3	*76,5*±0,3	*66,1*±0,3	*72,8*±0,2	*63,2*±0,3
Sıfat	6446	735	72,2±0,3	76,5±0,3	66,1±0,3	72,9±0,2	63,2±0,3
Belirteç	3033	221	74,8±0,3	76,6±0,3	66,3±0,3	73,1±0,2	63,4±0,3
Bağlaç	2200	44	70,4±0,4	76,6±0,3	66,2±0,3	**74,1**±0,2	**64,2**±0,3
Belirleyen	1998	13	74,8±0,3	76,6±0,3	66,1±0,3	72,8±0,2	63,3±0,3
Tekrar	11	9	74,9±0,3	76,5±0,3	66,1±0,3	72,8±0,2	63,2±0,3
Ünlem	100	34	74,9±0,3	76,5±0,3	66,1±0,3	72,8±0,2	63,2±0,3
İsim	21860	3935	60,8±0,5	**77,2**±0,2	**67,1**±0,2	**73,9**±0,2	**64,6**±0,3
Sayı	850	226	70,4±0,4	76,5±0,3	66,1±0,3	72,9±0,2	63,3±0,3
İlgeç	1250	46	73,6±0,4	76,6±0,3	66,2±0,3	72,9±0,2	63,2±0,3
Adıl	2145	28	75,0±0,3	76,5±0,3	66,1±0,3	72,8±0,2	63,2±0,3
Nokt.İş.	10420	16	75,1±0,3	77,1±0,4	66,6±0,3	73,4±0,2	63,7±0,3
Soru	228	6	74,9±0,3	76,5±0,3	66,1±0,3	72,8±0,2	63,2±0,3
Eylem	14641	1256	67,8±0,5	76,5±0,3	**66,6**±0,2	72,9±0,2	**63,8**±0,3

Tablo 4.8'de ilk sütun görünüm bilgisi eklenen birimin ana sözcük sınıfını, ikinci ve üçüncü sütunlar ilgili sözcük sınıfının toplam görülme sıklığı ve farklı gövdelerde görülme sıklığı bilgisini vermektedir. Tabloda, başarımda görülen artışlar koyu yazı karakteri ile belirtilmişlerdir.

- Burada olasılık tabanlı ayrıştırıcının, farklı gövdeler ile yüksek görülme sıklığına sahip olan sınıflarda yine seyrek veri sorunu yaşadığı görülmektedir (isim, eylem vb ...).
- Olasılık tabanlı ayrıştırıcıda, hiçbir sınıf için görünüm bilgisi eklemek ile daha yüksek bir başarım elde edilememiştir.
- Sınıflandırıcı tabanlı ayrıştırıcıda durum daha farklıdır. Hiçbir sözcük sınıfının görünüm bilgisinin eklenmesi başarımda düşüşe neden olmamaktadır.
- *KsmSb Derlem* üzerinde, isimlerin görünüm bilgisinin kullanılmasının başarımda istatistiksel olarak belirgin bir artışa neden olduğu görülmektedir.
- İsimlerin alt sözcük sınıfları[7] üzerinde yapılan daha ayrıntılı bir inceleme, sadece alt sözcük sınıfı genel isim olanların başarımda artışa neden oldukları, özel isimlerin görünüm bilgisinin kullanılmasının ise başarımda artışa neden olmadığını belirtmektedir.
- Eylemlerin görünüm bilgisinin kullanılması, etiketli başarımda ($\acute{C}KB_E$) görülür bir artışa neden olmakla beraber, bu artış istatistiksel olarak belirgin değildir.
- *Tüm Derlem* üzerinde, bağlaçların görünüm bilgisinin kullanılmasının da başarımı belirgin olarak arttırdığı görülmektedir. *KsmSb Derlem* üzerinde saptanmayan bu artış, "de, mi, ki" gibi sözcüğe ait biçimbilimsel özellik taşımalarına rağmen sözcükten sonra ve sözcükten ayrı olarak yazılan ve *KsmSb Derlem*'de bulunmayan sola bağımlı türde bağlılıklara yol açan eklerle ilişkilendirilebilir. Derlemde bağlaç olarak işaretlenen bu ekler, diğer bağlaçlardan görünüm bilgisi ekleme yolu ile ayırt edilebilmekte ve bu durum bu tür eklerin hemen sol taraflarında yer alan iye sözcüğe bağlanmalarını çok kolay hale getirmektedir.

Olasılık tabanlı ayrıştırıcıda görünüm bilgisi eklenmesi sonucunda herhangi bir artış gözlemlenmediği için bu bölümdeki daha ayrıntılı değerlendirmelere sınıflandırıcı tabanlı ayrıştırıcı üzerinde devam edilecektir. Bu değerlendirmelerde, bazı sözcük

[7]Genel isimlerden farklı olan isim sınıfları isimin özel isim veya gelecek zaman ortacı, geçmiş zaman ortacı, mastar eki ile veya ek almadan türemiş formlarını belirtmek üzere farklı alt sözcük sınıfları ile belirtilirler. Bu son dört çeşit LEMMA bilgisi içermezler.

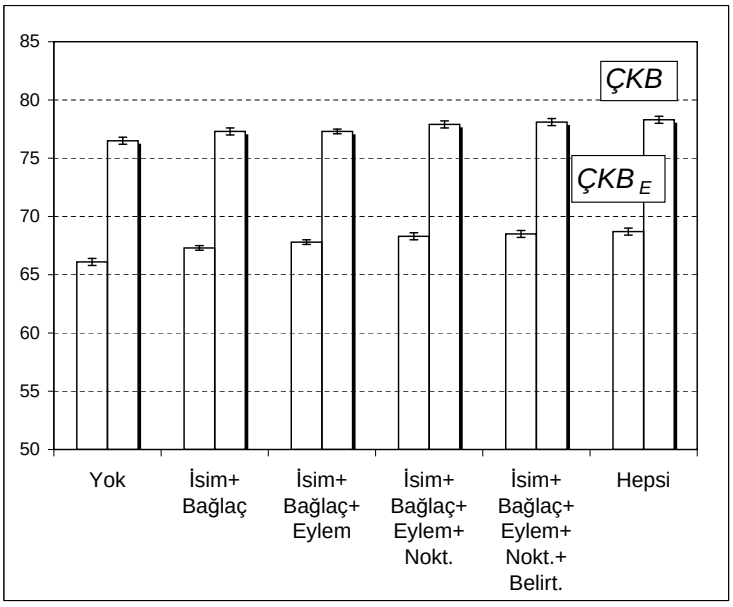
sınıflarının bileşimleri kullanılarak kısmi olarak görünüm bilgisi eklenmiş modeller sınanmıştır (bknz Şekil 4.2). Sonuçlar, Türkçe'de görünüm bilgisi kullanmanın başarımı kesinlikle arttırdığını göstermenin yanısıra, aslında sadece isimlerin ve bağlaçların görünüm bilgilerinin birlikte kullanılmasının başarımda önemli bir artışa neden olduğunu göstermektedir. Biçimbilimsel özelliklerin kullanılması ile ilgili incelemelere koşut şekilde, sınıflandırıcı tabanlı ayrıştırıcının görünüm bilgisi kullanımı tüm sözcükler için yapılsa bile seyrek veri sorunu yaşamadığı görülmektedir.

Yakın geçmişte, görünüm bilgisi kullanımının etkileri birçok çalışmada (Klein ve Manning, 2003; Dubey ve Keller, 2003; Arun ve Keller, 2005) incelenmesine rağmen birkaç çalışma haricinde (Bikel, 2004; Gildea, 2001) bu etkiler genelde ya hep ya hiç yaklaşımı ile ele alınmıştır. Bir diğer deyişle, incelemeler sırasında bir model üzerinde ya tüm sözcükler için görünüm bilgisi kullanmanın ya da hiç kullanmamanın etkileri irdelenmiştir. Türkçe için yapılan incelemeler (Eryiğit ve diğ., 2006b), görünüm bilgisi kullanmanın etkisinin sözcük sınıfları üzerinde düzgün bir dağılım göstermediğini ve de görünüm bilgisi kullanılmasının olumlu veya olumsuz etkisini anlayabilmek için daha ayrıntılandırılmış bir incelemeye gereksinim duyulduğunu açıkça göstermektedir. Bu çalışmadaki deneyler tam olarak bir örneğini teşkil etmese de, tüm birimlerin görünüm bilgisini kullanmak yerine bunların kısmi olarak kullanılmasının bazı modeller için (özellikle seyrek veri sorunundan etkilenenler için), daha iyi bir seçim olabileceği öngörülmektedir. Önceki bölümdeki sonuçlar, aynı durumun biçimbilimsel özelliklerin kullanılmasında da varolduğunu göstermektedir. Ancak bu sefer, biçimbilimsel özelliklerin kısmi olarak eklenmesinin (bütünüyle eklenmesinin aksine) olasılık tabanlı ayrıştırıcının başarımını arttırmaya yardımcı olduğu gösterilmiştir.

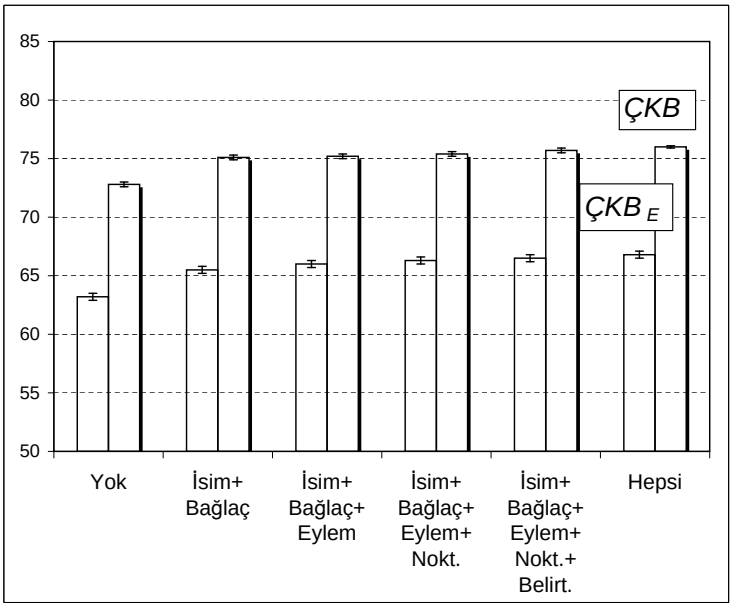
4.5 Daha Küçük Bir Eğitim Kümesi Kullanmanın Etkisi

Eğitim verisinin boyutunun ayrıştırıcıların başarımları üzerindeki etkisini görmek üzere, Şekil 4.3'de sonuçları verilen deneyler hazırlanmıştır. Aynı eğitim ve sınama kümelerini kullanmak üzere deneyler *KsmSb Derlem* üzerinde ve *ÇKB* başarımları ölçülerek yapılmıştır. Şekil, olasılık tabanlı ayrıştırıcının (görünüm bilgisi

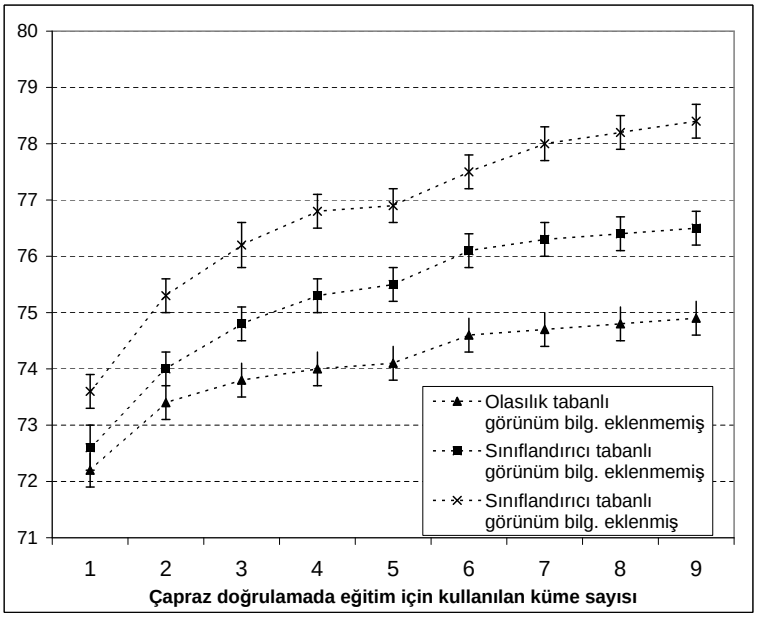
a) *KsmSb Derlem* üzerinde



b) *Tüm Derlem* üzerinde



Şekil 4.2: Sınıflandırıcı Tabanlı Ayrıştırıcı Kademeli Görünüm Bilgisi Eklenmesi



Şekil 4.3: Farklı Eğitim Verisi Boyutları ile *ÇKB* Başarımları (*KsmSb Derlem* üzerinde)

eklenmemiş) ve sınıflandırıcı tabanlı ayrıştırıcının (görünüm bilgisi eklenmiş ve görünüm bilgisi eklenmemiş) başarımlarını vermektedir. Şekilde x ekseni her adımda eğitim sırasında kullanılan çapraz doğrulama küme sayısını göstermektedir.

Görünüm bilgisi eklenmiş sınıflandırıcı tabanlı ayrıştırıcının, eğitim sırasında, çapraz doğrulama için oluşturulmuş 10 eğitim kümesinden 9'unu kullanmak ile 1'ini kullanmak arasındaki başarım düşüşünün diğer ayrıştırıcılara oranla $4{,}8_{\pm 0{,}1}$ ile en büyük düşüş olduğu görülmektedir. Bu sayı görünüm bilgisi eklenmemiş sınıflandırıcı tabanlı ayrıştırıcı için $3{,}9_{\pm 0{,}2}$, olasılık tabanlı ayrıştırıcı için $2{,}7_{\pm 0{,}1}$'dir.

Olasılık tabanlı ayrıştırıcının, düşük başarımına karşın, eğitim verisinin boyutundan en az etkilenen ayrıştırıcı olduğu görülmektedir. Ancak küme sayısının 1 olduğu durum hariç, tüm boyutlar için, modellerin göreceli sıralamaları aynı kalmaktadır. Küme sayısının 1 olduğu durumda olasılık tabanlı ayrıştırıcı ile görünüm bilgisi eklenmemiş sınıflandırıcı tabanlı ayrıştırıcıların başarımları arasında istatistiksel olarak belirgin bir fark yoktur. Bir diğer sonuç, sınıflandırıcı tabanlı modellerin artan eğitim verisi boyutu ile bilgi çıkarmada daha başarılı olduğu ancak olasılık tabanlı ayrıştırıcının eğitim verisinin artması ile çok fazla gelişemediğidir. Bu durum özellikle küme sayısının 6 olduğu durumla 9 olduğu durum arasında başarımı belirgin olarak artan görünüm

bilgisi eklenmiş sınıflandırıcı tabanlı ayrıştırıcıda saptanmaktadır. Bu aralıkta görünüm bilgisi eklenmemiş modellerde belirgin bir artış tespit edilememektedir.

4.6 Hata İncelemeleri

Bu bölümde, *Tüm Derlem*'in ayrıştırılması sonucunda elde edilen en iyi sonuçlar üzerinde ayrıntılı hata incelemeleri yapılmıştır. Öncelikle farklı bağlılık türleri için başarım değerlendirmesi yapılmış, daha sonra ayrıştırıcı tarafından atanan iye ile gerçek iye arasındaki uzaklığa bağlı olarak hata dağılımları incelenmiştir. Son olarak, tümce uzunluğuna bağlı hata dağılımları incelenmiştir. İncelemeler, 10 katlı çağraz doğrulama sonucunda elde edilen sonuçların bir araya getirilmesi ile elde edilen sonuçlar üzerinde yapılmıştır.

4.6.1 Bağlılık Türüne Göre Başarım Değerlendirmesi

Bu bölümde, sınıflandırıcı tabanlı ayrıştırıcı ile elde edilen en iyi ayrıştırma sonuçları kullanılarak farklı bağlılık türleri üzerinde değerlendirmeler yapılmıştır. Tablo 4.9 eniyileştirilmiş model ile *Tüm Derlem* üzerinde bağlılık türü temelinde elde edilen *ÇKB*, kesinlik (P), gerigetirim (R) ve F ölçütü[8] değerlerini vermektedir. Bunlara ek olarak, her bağlılık türü için görülme sıklığı (n) ve uydu-iye arasındaki ortalama uzaklık bilgileri verilmiştir. Tablo *ÇKB* değerlerine göre büyükten küçüğe doğru sıralanmıştır.

Tablodan, derlem içerisinde 100'den daha az sayıda görülen bağlılık türleri için ayrıştırıcının etiketli bağlılıkları bulamadığı gözlemlenmektedir.[9] Bu duruma tek aykırı örnek ilişkilendirici (RELATIVIZER) bağlılık türü içindir. Bu bağlılık türü için "n" 100'den küçük olmasına karşın etiketli başarımları 0'dan yüksek çıkmıştır. Bu türde bağlılıklar genellikle bağlandığı iye sözcüğün sağ tarafında yer alan ve

[8] Bu ölçütler için ayrıntılı bilgi Ek D'de bulunabilir

[9] ROOT türü tümce içerisinde hiçbir yere bağlanmadan duran sözcüklere verilen bağlılık türüdür. Derlemde genelde tümcenin en sonundaki noktalama işareti bağlılık ağacının kökü kabul edildiğinden ve noktalama işaretleri tabloda değerlendirilmeye alınmadığından, burada ROOT bağlılık türlerinin sayısı çok azdır.

Tablo 4.9: Bağlılık Türlerine Göre Başarım Değerlendirmesi
(P = kesinlik, R = gerigetirim, F = F ölçütü, n = görülme sıklığı, dist = bağlılık uzunluğu)

Tür	n	*uzaklık*	*ÇKB*	P	R	F
SENTENCE	7252	1,5	90,5	87,4	89,2	88,3
DETERMINER	1952	1,3	90,0	84,6	85,3	85,0
QUESTION.PARTICLE	288	1,3	86,1	80,0	76,4	78,2
INTENSIFIER	903	1,2	85,9	80,7	80,3	80,5
RELATIVIZER	85	1,2	84,7	56,6	50,6	53,4
CLASSIFIER	2048	1,2	83,7	74,6	71,7	73,1
POSSESSOR	1516	1,9	79,4	81,6	73,6	77,4
NEGATIVE.PARTICLE	160	1,4	79,4	76,4	68,8	72,4
OBJECT	7956	1,8	75,9	63,3	62,5	62,9
MODIFIER	11685	2,6	71,9	66,5	64,8	65,7
DATIVE.ADJUNCT	1360	2,4	70,8	46,4	50,2	48,2
FOCUS.PARTICLE	23	1,1	69,6	0,0	0,0	0,0
SUBJECT	4479	4,6	68,6	50,9	56,2	53,4
ABLATIVE.ADJUNCT	523	2,5	68,1	44,0	54,5	48,7
INSTRUMENTAL.ADJUNCT	271	3,0	62,7	29,8	21,8	25,2
ETOL	10	4,2	60,0	0,0	0,0	0,0
LOCATIVE.ADJUNCT	1142	4,2	56,9	43,3	48,4	45,7
COORDINATION	814	3,4	54,1	53,1	49,8	51,4
S.MODIFIER	594	9,6	50,8	42,2	45,8	43,9
EQU.ADJUNCT	16	3,7	50,0	0,0	0,0	0,0
APPOSITION	187	6,4	49,2	49,2	16,6	24,8
VOCATIVE	241	3,4	42,3	27,2	18,3	21,8
COLLOCATION	51	3,3	41,2	0,0	0,0	0,0
ROOT	16	-	0,0	0,0	0,0	0,0
Toplam	43572	2,5	76,0	67,0	67,0	67,0

ayrı yazılan “ki” (bağlaç) parçacıkları ile oluşmaktadırlar. Bu bağlılık türü hep aynı parçacık ile görüldüğünden, seyrek veri sorunu diğerlerindeki kadar çok ortaya çıkmamaktadır.

Eğer bu az sıklıkta görülen bağlılıklar listeden çıkarılırsa, sonuçları üç kümeye ayırmak mümkündür. İlk küme %79'un üzerinde *ÇKB* başarımı gösteren ve yakın uzaklıklarda bulunan belirleyenler, parçacıklar ve isim tamlamalarından oluşmaktadır. İkinci küme genel olarak öznelerden, nesnelerden ve farklı türdeki tümleçlerden oluşmaktadır. Bu kümedeki bağlılıklar iyelere 1,8 - 4,6 *ÇK* uzaklıkta bulunmakta ve %55 - %79 arası *ÇKB* başarımı göstermektedirler. Bu küme, biçimbilimsel özelliklerin doğru bağlılığı bulmakta en çok önem taşıdığı küme olarak gösterilebilir. Üçüncü küme

diğerlerine göre çok daha düşük başarımları olan ve iye ve uydunun birbirlerine daha uzak yerlerde bulunduğu bağlılık türlerini içermektedir. Bunlar genelde, isim, nesne ve niteleyiciler gibi isim türlü sözcüklerden kolayca ayırt edilemeyen ve bu nedenle ayrıştırıcı tarafından bulunmaları zor olan söylemsel bağlılıklar, ünleme ve ilave açıklama türündeki ilişkilerdir. Bulunması zor olan bir diğer ilişki türü ise bağlaçlardır.

4.6.2 Hata Uzaklığına Göre Hata İncelemesi

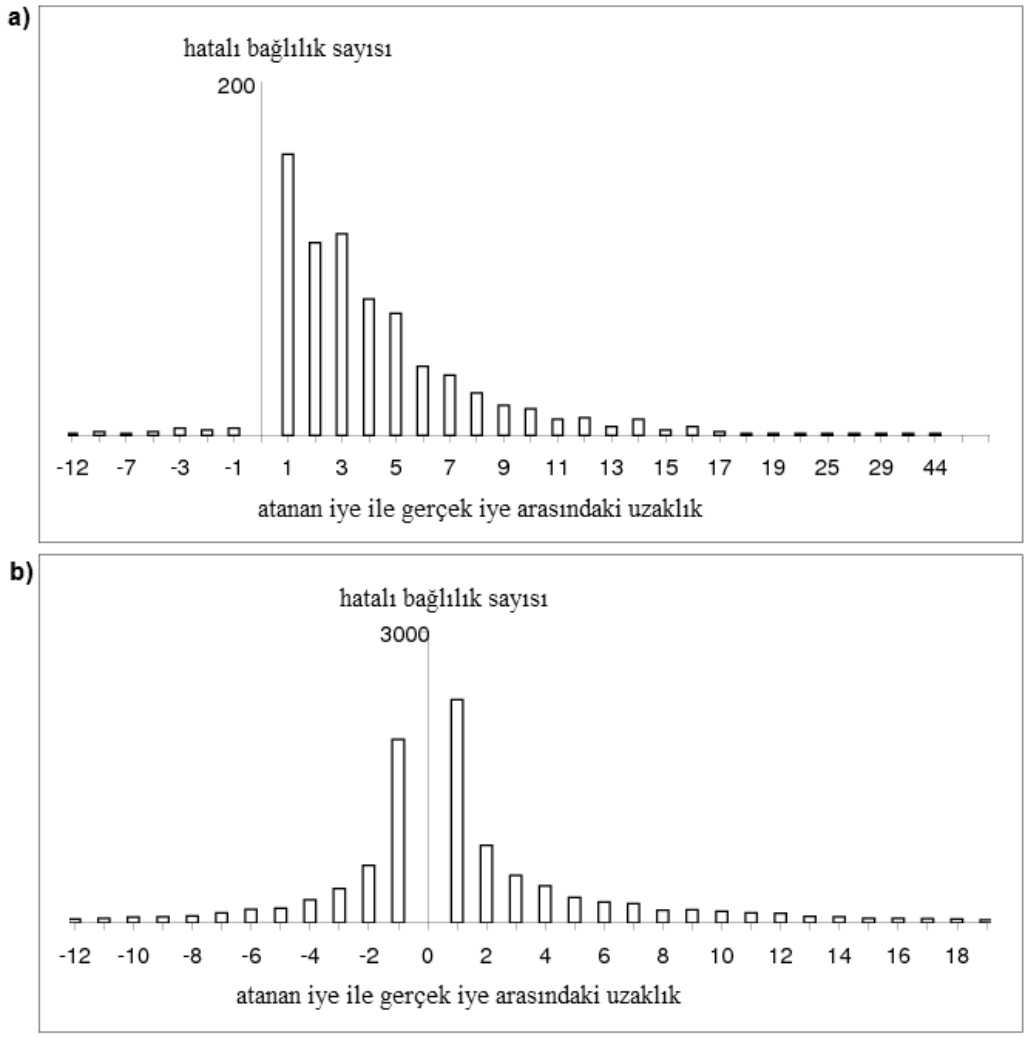
Sonuçlar bağlılık yönüne göre değerlendirildiğinde, sola bağımlı bağlılıklar için 72,2'lik bir *ÇKB* değeri, sağa bağımlı bağlılıklar için ise 76,2'lik bir *ÇKB* değeri elde ettiğimiz görülmektedir. Şekil 4.4-a ve Şekil 4.4-b, sola bağımlı ve sağa bağımlı bağlılıkların etiketsiz başarıma dayalı hata dağılımlarını vermektedir. Şekillerdeki x ekseni, ayrıştırıcı tarafından atanan iye *ÇK* ile gerçek *ÇK* arasındaki uzaklığı vermektedir. Bu uzaklık atanan iye *ÇK*'nin sıra numarasından, gerçek iye *ÇK*'nin sıra numarasının çıkarılması ile bulunmuştur.

Önceden de bahsedildiği gibi, sola bağımlı bağlılıkların sayısı derlemin %5'ini oluşturmaktadır. Şekil 4.4-a, ayrıştırıcının sola bağımlı bağlılıkları gerçek iyeden daha yakın veya yanlış yönde bir iyeye bağlama eğiliminde olduğunu göstermektedir. Bu bağlılıklar incelendiğinde, bunların %70,4'nün uyduya komşu (sınırdaş) bir iyeye bağlandıkları ve ayrıştırıcının da bu tip bağlılıkların %90,1'ini doğru olarak belirlediği görülmektedir. Buradan yola çıkarak, ayrıştırıcının sınırdaş sola bağımlı bağlılıkları bulmada sorun yaşamadığı yorumu yapılabilir. Buna ek olarak, hata uzaklığının 1 olduğu (Şekil 4.4-a)[10] hataların %86,8'nin, bağlılığın doğru iye sözcüğün yanlış *ÇK*'sine bağlanması sebebiyle oluştuğu görülmektedir. Bu tip bağlılıklarda bağlılığın yönünü bulmada yapılan hata %19,8'dir.

Ayrıştırıcı sağa bağımlı bağlılıkları bulmada %100 başarılıdır. Buna ek olarak, Şekil 4.4-b[11]'den görüldüğü üzere, ayrıştırıcının doğru iyeyi belirlerken yaptığı

[10]Gerçek iye ile atanan iyenin sınırdaş olması anlamına gelmektedir.

[11]40 görülme sıklığından az olan hata uzaklıkları şekile eklenmemiştir.



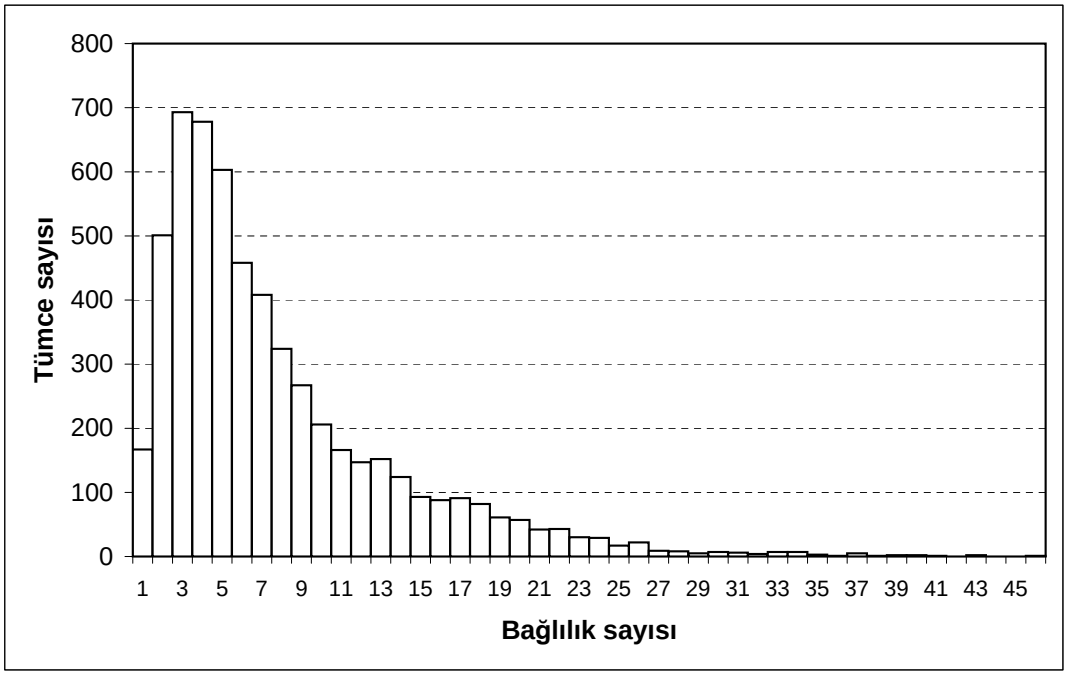
Şekil 4.4: Hatalı Bağlılık Dağılımları
a) sola bağımlı bağlılıklar b) sağa bağımlı bağlılıklar

hatalar yaklaşık olarak normal bir dağılım göstermektedir. Aynı şekilden, hataların, %57,3'nün gerçek iyeden ± 2 *ÇK* uzaklık aralığında bulunduğu görülmektedir.

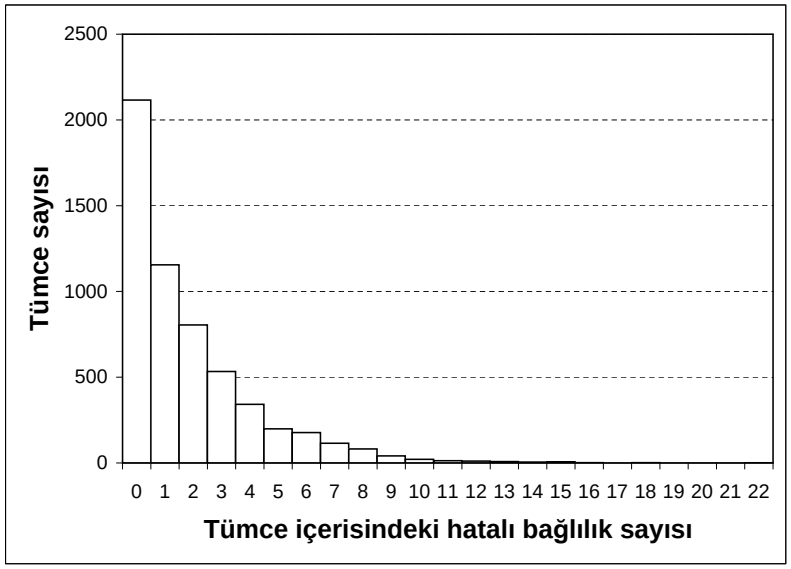
4.6.3 Tümce Uzunluğuna Göre Hata İncelemesi

Ayrıştırıcımızın, bir tümce içerisindeki bağlılıkların tümünü etiketsiz olarak doğru bulma başarımı (*Tüm Derlem*) *TB* =%37,5'dir[12]. Şekil 4.5 ve Şekil 4.6 Türkçe derlemdeki sözcüklerin kaç tümceden oluştuğu ve ayrıştırma sonucunda tümcelerin kaç hatalı bağlılıktan dolayı hatalı kabul edildiklerine ilişkin dağılımları vermektedir.

[12]Bir tümce içerisindeki bağlılıkların tümünü doğru sözcüğe (doğru *ÇK* dikkate alınmadan) bağlama başarımı %46,5'dir.

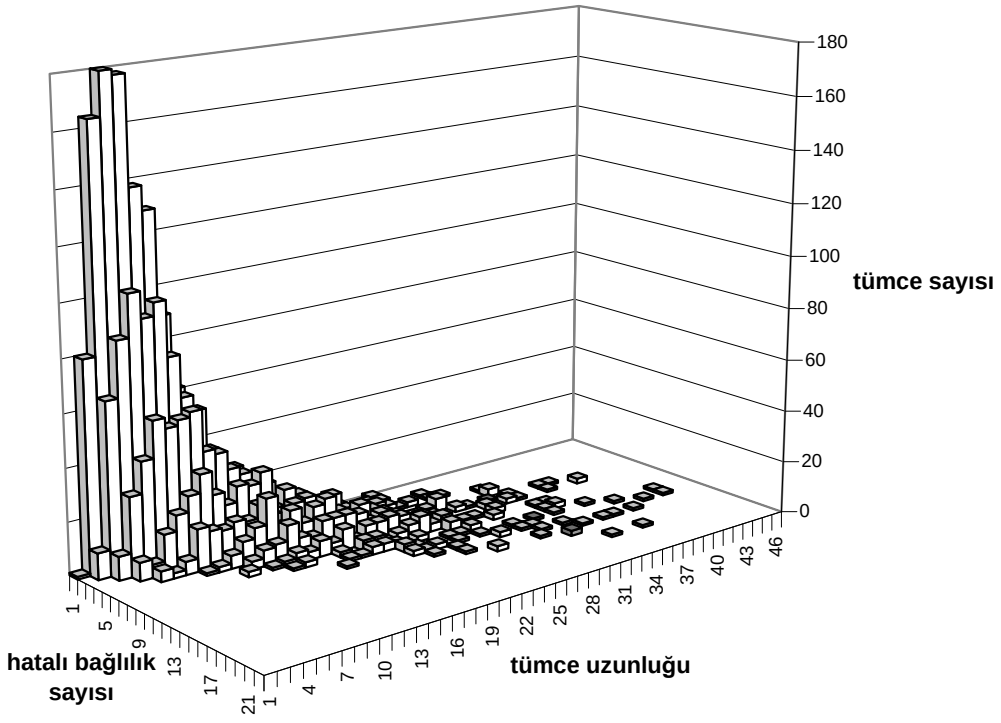


Şekil 4.5: Tümcelerin Bağlılık Dağılımları



Şekil 4.6: Tümcelerin Hatalı Bağlılık Dağılımları

Derlemdeki tümcelerin %90'ı 15 veya daha az bağlılıktan oluşmaktadır. Tümcelerin içerdiği ortalama bağlılık sayısı 8'dir[13]. Şekil 4.6'da x ekseninde hata sayısının 0 olduğu çubuk, ayrıştırma sonucunda tüm bağlılıkları doğru bulunan tümcelerin sayısını belirtmektedir. Bu değer dışarıda bırakılarak, sadece hatalı tümceler incelendiğinde, hatalı tümcelerdeki ortalama hata sayısının 3,0 olduğu görülmektedir. *KsmSb Derlem* üzerindeki başarımlara bakıldığında ise, *TB* başarımının %45,2 ve hatalı tümcelerdeki ortalama hata sayısının 2,6 olduğu görülmektedir. İki küme arasındaki başarımlar karşılaştırıldığında, *Tüm Derlem* içerisindeki kesişen ve sola bağımlı türde bağlılıkların hata oranını arttırdığı görülmektedir. Bu durum derlem içerisinde bu tür örneklerin, ayrıştırıcının bu örnekleri öğrenememesine neden olacak derecede az olmasının[14] (bknz Bölüm 2) doğal bir sonucu olarak görülmektedir.



Şekil 4.7: Tümce Uzunluğuna Bağlı Hata Dağılımları

Şekil 4.7 farklı uzunluklardaki tümceler üzerinde hata dağılımlarını göstermektedir. Şekilde, x ekseni tümce uzunluğunu (bağlılık sayısı ile hesaplanmış), y ekseni

[13]Bu sayı hesaplanırken noktalama işaretlerinden çıkan bağlılıkların sayılmadığı hatırlanmalıdır.

[14]Daha önce de bahsedildiği gibi, kesişen bağlılık içeren tümceleri ayrıştırmaya yönelik incelemeler bu tür örneklerin azlığı nedeni ile Türkçe üzerinde başarımı arttırmamışlardır.

hatalı bağlılık sayısını, z ekseni ise belirli bir tümce uzunluğu ve hata sayısının görülme sıklığını belirtmektedir. Beklenildiği gibi, hataların çoğunluğu, az sayıda hata barındıran kısa tümcelerde (özellikle 7 ve daha az bağlılık ve tek hata içeren tümceler) görülmektedir. Ortalama hata sayısının tümce uzunluğu ile doğrusal orantılı olduğu ve buradan da sözcük başına düşen hata olasılığının tümce uzunluğu ile artmadığı yorumu yapılabilir.

4.7 Yetkin Etiketler Kullanmanın Etkisi

Bu bölüme kadar yapılan bütün incelemelerde, derlem tarafından sağlanan yetkin[15] etiketler kullanılmıştır. İncelenmesi gerekli görülen bir diğer nokta ise bir *sözcük etiketleyici*$^{\times}$ tarafından atanan etiketlerin kullanılmasının etkisidir. Bu amaçla, derlemde yer alan sözcükler öncelikle Oflazer (1994)'in iki düzeyli biçimbilimsel çözümleyicisinden geçirilerek, her biri için olası biçimbilimsel çözümlemeler çıkartılmıştır.[16] Bu işlemden sonra, %96'lık etiketleme başarımı ile Türkçe için en başarılı etiketleyici olduğu öne sürülen Yüret ve Türe (2006)'in sözcük etiketleyicisi kullanılarak birden çok olası çözümlemesi olan sözcüklerdeki belirsizlikler giderilmeye çalışılmıştır. Türkçe gibi bitişken bir dilin biçimbilimsel belirsizliğinin giderilmesindeki karmaşıklık bir sözcüğe atanabilecek olası etiketlerin sayısının çokluğundan kaynaklanmaktadır (Yüret ve Türe, 2006). Türkçe'de olası biçimbilimsel etiket sayısı kuramsal olarak sonsuzdur.[17] Türkçe'de, etiketleyici doğru sözcük sınıfının yanı sıra doğru biçimbilimsel özellikleri de belirlemek durumundadır. Aşağıda "kalemi" sözcüğü için biçimbilimsel çözümleyicinin oluşturduğu üç farklı biçimbilimsel çözümleme örnek olarak gösterilmektedir. Bu örnekte, birinci çözümleme "kalemi" sözcüğünün, "kale" sözcüğünün 1. tekil kişi iyelik ve -i hal eki

[15] Yetkin etiket ile biçimbilimsel çözümlemenin sonucunda ortaya çıkan belirsizliklerin insanlar tarafından giderilmesi sonucunda bulunan etiketler kastedilmektedir. Derlem hazırlanırken veri bu şekilde hazırlanmıştır.

[16] Biçimbilimsel çözümlemenin sonunda, sözcüklerin %39'unun belirsizlik içerdiği ve bunların %44'ünün ikiden daha fazla olası biçimbilimsel çözümlemesi olduğu görülmüştür.

[17] Sınıflandırıcı tabanlı ayrıştırıcı (Tablo 4.3) sözcük tabanlı modelde farklı etiket sayısı 718'dir. Sadece farklı *ÇK* etiketlerinin sayısı sayıldığında bu sayı 108'dir.

almış şekli olduğunu belirtmektedir. İkinci ve üçüncü çözümlemelerde ise bu sözcüğün "kalem" sözcüğünün farklı ekler almış halleri belirtilmektedir.

kalemi

- kale +Noun+A3sg+P1sg+Acc (*kale* + 1. tekil kişi iyelik eki + -i hali eki)
- kalem +Noun+A3sg+P3sg+Nom (*kalem* + 3. tekil kişi iyelik eki)
- kalem +Noun+A3sg+Pnon+Acc (*kalem* + -i hali eki)

Örnekte olası biçimbilimsel çözümler arasından seçim yaparken sözcük etiketleyicinin sadece doğru sözcük sınıfını seçmesi yeterli değildir. Görüldüğü gibi her üç çözümlemede de sözcük sınıfı isimdir. Sözcük etiketleyici bu sınıf ile birlikte doğru biçimbilimsel özellikleri de ve eğer varsa *ÇK* sınırlarını da belirlemelidir.

Yüret ve Türe (2006)'nin sözcük etiketleyicisinin başarımını derlem verimiz üzerinde ölçtüğümüzde, derlem ile birebir aynı etiketleri atama başarımının, etiketleyici kullanmadan doğrudan etiketlediğimiz noktalama işaretleri ve biçimbilimsel çözümlemesi yapılamamış ve bu nedenle hazırlanan bir listeden[18] çekilen sözcükler de dahil olmak üzere, %88,4 olduğu görülmektedir. Sözcük etiketleyicinin derlem üzerindeki başarımının raporlanan başarımından (Yüret ve Türe, 2006) daha düşük çıkmasının nedeni etiketleyicinin (derlem verisinden tamamen farklı olan) eğitim kümesinin etiketlenmesi sırasında farklı seçimler yapılmış olmasına bağlanabilir.

Bu bölümde sınıflandırıcı tabanlı ayrıştırıcı, yukarıda bahsedilen sözcük etiketleyici kullanılarak etiketlenen derlem verisi üzerinde sınanmıştır. Sözcük etiketleyicinin sözcük tabanlı ve *ÇK* tabanlı modeller üzerindeki etkisini incelemek üzere öncelikle görünüm bilgisi eklenmiş bir sözcük tabanlı model yetkin etiketler kullanılmış ve sözcük etiketleyici tarafından etikenlenmiş bir veri üzerinde değerlendirilmiştir. Bu modelde özellik kalıbı olarak Şekil 3.9'daki özelliklere LEMMA σ_0, τ_0, τ_1 özelliklerinin eklenmesi ile oluşan bir kalıp kullanılmıştır.

[18]Derlem içerisindeki bazı sözcükler biçimbilimsel çözümleyici tarafından tanınmamaktadırlar. Bunlar genel olarak özel isimler, sayılar ve derlemin geliştirilmesi sırasında birden çok sözcüğün biraraya getirilmesi ile oluşturulan birleşik isimlerdir ve de derlemin %6,2'sini oluşturmaktadırlar. Bu sözcüklerin çözümlemelerini doğrudan bir arama tablosundan çekilmişlerdir. Eğer bu sözcükler de etiketleyicinin başarımı ölçülürken değerlendirme dışı bırakılırsa başarım %84,6 olmaktadır.

Değerlendirmede oluşacak bir sorun, sözcük etiketleyicinin bazı sözcükler için yetkin etiketlerden tamamen farklı *ÇK* yapısında bir biçimbilimsel çözümleme seçmesi sonucunda, ayrıştırıcı tarafından atanacak iye *ÇK*'nin gerçek yapı ile ilgisiz olacak olmasıdır. Bu sorunu çözmenin tek ve basit bir yolu yoktur. Bu nedenle, sözcük etiketlemenin etkisini ayrıntılı olarak anlayabilmek üzere oluşturduğumuz farklı değerlendirme yöntemleri aşağıda listelenmiştir. Tüm durumlarda, *SB* ölçütü hesaplanırken, uydunun doğru iye sözcüğe bağlanıp bağlanmadığına bakılmış, sözcük etiketleyiciden dolayı oluşan hatalar dikkate alınmamıştır. Benzer şekilde, sözcük tabanlı modelde *ÇKB* ve $ÇKB_E$ başarımları hesaplanırken, bağlılıkların iye sözcüğün ilk *ÇK*'sinde sonlandığı varsayımı yapılmış ve sözcük etiketleyiciden dolayı oluşan hatalar dikkate alınmamıştır. *ÇK* tabanlı modelde ise, uydu ve iye sözcüğün yetkin veri ile tamamen aynı etiketlere sahip oldukları durumlarda, *ÇKB* ve $ÇKB_E$ başarımları önceden olduğu biçimde hesaplanmışlardır. Ancak, uydu sözcükte veya iye sözcükte (veya her ikisinde birden) etiketleme hataları oluştuğunda, bağlılıklar aşağıdaki dört farklı yönteme göre değerlendirilmiştir:

Olağan Eğer bağlılık doğru iye sözcüğün ilk *ÇK*'sine bağlanıyorsa doğru kabul edilir (sözcük tabanlı model sonuçları ile karşılaştırma yapabilmek üzere hazırlanmış olağan varsayım).

İye ÇK Eğer bağlılık doğru iye sözcüğe bağlanmış ve bağlanılan iye *ÇK* yetkin verideki ile aynı sözcük sınıfına sahip ise bağlılık doğru kabul edilir.

Her iki ÇK Eğer bağlılık doğru iye sözcüğe bağlanmış ve hem uydu *ÇK* hem de iye *ÇK* yetkin verideki ile aynı sözcük sınıfına sahip ise bağlılık doğru kabul edilir.

Her iki sözcük Eğer bağlılık doğru iye sözcüğe bağlanmış ve hem uydu sözcük hem de iye sözcük yetkin verideki ile birebir aynı etiketi taşıyorsa bağlılık doğru kabul edilir.

Tablo 4.10 sözcük tabanlı model ile *ÇK* tabanlı modelin biçimbilimsel belirsizlik giderimi hatalarından eşit derecede etkilendiğini ve başarımlarındaki düşüşün aynı derecelerde olduğunu göstermektedir. (Aynı zamanda, sözcük etiketleyici tarafından etiketlenmiş verinin kullanılması durumunda da *ÇK* tabanlı modelin sözcük tabanlı modele göre daha yüksek başarım verdiği görülmektedir.) En katı değerlendirme

Tablo 4.10: Sözcük Etiketleyicinin Etkisi Özet Tablo

		ÇKB	ÇKB$_E$	SB
Sözcük tabanlı	*Yetkin veri*	*71,2*±0,3	*62,3*±0,3	*82,1*±0,9
	etiketli	69,5±0,3	59,3±0,3	80,2±0,9
ÇK tabanlı	*Yetkin veri*	*76,0*±0,2	*67,0*±0,3	*82,7*±0,5
	etiketli *Olağan*	73,1±0,3	63,0±0,3	80,6±0,7
	etiketli *İye ÇK*	73,3±0,3	63,2±0,3	80,6±0,7
	etiketli *Her iki ÇK*	70,1±0,3	61,6±0,3	80,6±0,7
	etiketli *Her iki sözcük*	62,8±0,3	55,8±0,3	80,6±0,7

yöntemimiz, "*her iki sözcük*" yöntemidir. Bu yöntem sözcük etiketleyici tarafından hatalı olarak çözümlenen sözcüklerden (tüm sözcüklerin %11.6'sı) çıkan ve bu sözcüklere giren tüm bağlılıkları hatalı kabul etmektedir. Biçimbilimsel özelliklerde yapılan bazı etiketleme hatalarının bağlılık türünü her zaman için etkilemediği göz önünde bulundurulusa, bu değerlendirmenin çok katı olduğu söylenebilir. Örneğin, yukarıda etiketlenmesi yapılmış olan "kalemi" sözcüğünün önüne "küçük" sıfatı geldiğinde, gösterilen belirsiz çözümlemelerin hiçbiri "küçük" sıfatının "kalemi" ismine niteleyici (MODIFIER) bağlılık türü ile bağlanmasını etkilemez. Buna ek olarak, ana sözcük sınıflarında ortaya çıkan etiketleme hataları ayrıştırıcının doğru iye sözcüğü bulmasını doğrudan etkileyecek ve bu da *SB* başarımında düşüş (82,7'den 80,6'ya) olarak gözlemlenecektir. Öte yandan, "*iye ÇK*" olarak adlandırdığımız değerlendirme yöntemimiz, bağlılıkların her zaman için uydu sözcüğün son *ÇK*'sinden çıktıklarını göz önüne alarak, uydu sözcükteki ve iye sözcüğün iye *ÇK* dışındaki diğer *ÇK*'lerinde oluşan etiketleme hatalarını dikkate almamaktadır. Bu yöntemi *ÇK* tabanlı model ölçütümüz olarak aldığımızda, biçimbilimsel çözümleyici ve sözcük etiketleyici kullanmanın (hem sözcük tabanlı hem de *ÇK* tabanlı model için), *ÇKB* başarımında yaklaşık %3'lük, *ÇKB*$_E$ başarımında ise yaklaşık %4'lük bir düşüşe neden olduğu söylenebilir.

4.8 Conll-X Ortak Çalışması

ACL[19]'nin doğal dil öğrenmesi× konusunda çalışan özel ilgi grubu SIGNLL[20]'ın düzenlediği Bilişimsel Doğal Dil Öğrenmesi Konferansı CoNLL (*Conference on Natural Language Learning*) bu konuda geliştirilen sistemleri karşılaştırabilmek üzere her yıl katılımcıların aynı veri üzerinde sistemlerini eğitip sınadıkları ortak çalışmalar düzenlemektedir. 2006 yılında onuncusu düzenlenen bu çalışmanın konusu çok dilli bağlılık ayrıştırması olarak belirlenmiştir (Buchholz ve Marsi, 2006). Ortak çalışma kapsamında 13 farklı dil için varolan derlemler aynı biçime dönüştürülmüş ve eğitim ve sınama kümesi olmak üzere ikiye bölünmüşlerdir. Katılımcılardan öncelikle eğitim kümesi üzerinde ayrıştırıcılarını eğitmeleri istenmiş ve üç aylık süre[21] sonunda sınama kümesi yayınlanarak başarımlar ölçülmüştür.

Aşağıdaki kısımlarda öncelikle, bu ortak çalışma için konferans düzenleyicileri tarafından, Türkçe derlem üzerinde yapılan değişiklikler ve etkileri ve daha sonra bu veri üzerinde çalıştırılmış farklı ayrıştırıcıların başarımları ile yapılan karşılaştırmalar verilecektir.

4.8.1 Derlem Dönüşümleri ve Etkileri

Ortak çalışmada, değerlendirme işlemi sırasında noktalama işaretleri işlem dışı bırakılmıştır. Derlemler, Bölüm 3.3.2'de ayrıntısı verilen Conll-X gösterim biçimine dönüştürülürken, noktalama işaretlerinin hiçbir uydusu olmayacak şekilde değiştirilmişlerdir.

Diğer birçok derlemde olduğu gibi, Türkçe derlem içerisinde de noktalama işaretlerinin bağlanmasında sorunlar ile karşılaşılmaktadır. Derlemde çoğu noktalama işareti hiçbir yere bağlanmamış şekilde durmaktadır. Ancak, bazı noktalama işaretleri bir iyeye bağlı veya kendine bağlı bir uydu bulundurmaktadır. Bu tür durumlar

[19] ACL (Association for Computational Linguistics) Bilişimsel Dilbilim konusunda çalışan kişileri ortak bir çatı altında toplayan en büyük dernektir.

[20] http://ilps.science.uva.nl/ erikt/signll/about/

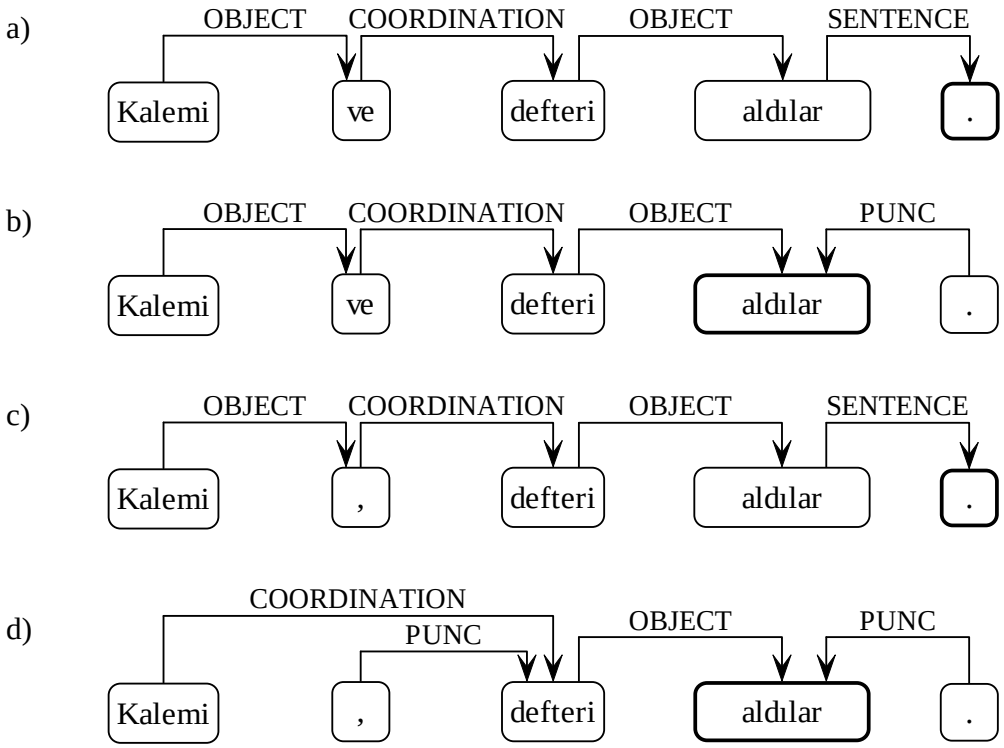
[21] Ortak Çalışma ile ilgili genel bilgiler tanıtım yazısından (Buchholz ve Marsi, 2006) alınmıştır.

genelde dolaylı söylemlerde veya aynı türden iki birimin bağlanması sırasında noktalama işaretinin bağlaç olarak kullanılmasında ortaya çıkmaktadır. Bunlara ek olarak tümcenin ana eylemi de tümce sonundaki noktalama işaretine uydu olarak ("*SENTENCE*" bağlılık türü ile) bağlanmaktadır. Ancak, derlemde bu tür bağlılıklarda belirli bir standart korunamamış bu yüzden birbirine çok benzer yapılar farklı şekillerde işaretlenmişlerdir. Bu nedenle, Conll-X ortak çalışmasında bu duruma geçici bir çözüm getirmek üzere derlem üzerinde bazı dönüşümler gerçekleştirilmiştir. Ancak köklü bir çözüm için hataların tek tek kontrol edilerek ve belirli bir standart korunarak düzeltilmesi gerekmektedir.

Derlem dönüştürülürken, bir noktalama işaretine uydu olan birimler, noktalama işareti yerine noktalama işaretinin bağlı olduğu birime bağlanmışlardır. Bu esnada bağlılık türü olarak da, noktalama işareti iye birime bağlanırken kullanılan bağlılık türü kullanılmıştır. Bu işlem sonrasında, değişime uğrayan bu noktalama işaretlerinin bağlılık türleri de *PUNC* adı verilen yeni bir bağlılık türüne dönüştürülmüşlerdir. Derlemde, hiçbir yere bağlı olmadan duran noktalama işaretleri ise içerisinde bulundukları bağlılık ağacında kesişen bağlılıklara izin vermeyecek şekilde en yüksek ara birime *PUNC* bağlılık türü ile bağlanmışlardır.

Şekil 4.8, noktalama işaretlerine ilişkin dönüşüm ile ilgi örnekler göstermektedir. Şekilde a) ve c) ile gösterilen ayrıştırma, tümcelerin derlemdeki esas gösterimlerini yansıtmaktadır. Görüldüğü üzere her iki ayrıştırmada da tümcenin ana eylemi olan "aldılar" sözcüğü tümce sonundaki noktalama işaretine bağlanmaktadır. Bu noktalama işaretleri aynı zamanda ayrıştırma ağacının da kökü olmaktadırlar. Şekillerde ayrıştırma ağacının kökü koyu renkte dikdörtgen içerisinde gösterilmiştir.

Şekilde b) ve d) ile gösterilen ayrıştırmalar ise a) ve b) ayrıştırmalarında gösterilen tümcelerin dönüşümden sonraki hallerini yansıtmaktadırlar. Görüldüğü gibi, b) satırındaki ayrıştırmada a)'dan farklı olarak noktalama işaretine bağlı olan "aldılar" sözcüğü bu noktalama işaretinden koparılarak, noktalama işaretinin iyesine bağlanmıştır. Bu durumda noktalama işaretinin iyesi olmadığı ve ayrıştırma ağacının kökü olduğu için "aldılar" sözcüğü koparılarak ağacın kökü haline getirilmiştir. Tümce sonundaki noktalama işareti ise oluşan ağaçta kesişen bağlılıklara izin vermeyecek



Şekil 4.8: Noktalama İşaretleri Dönüşüm

şekilde en yüksek noktaya, yani "aldılar" sözcüğüne ilişkilendirilmiştir. Şekilden görülebileceği gibi, bu bağlılık için yapılan işlem d) satırında da aynıdır.

a) ve c) satırlarındaki tümcelerin her ikisinde de "kalemi" ve "defteri" sözcükleri birbirlerine bağlanmışlardır. Bu birleştirme a) tümcesinde "ve" bağlacı ile c) tümcesinde ise "," (virgül) noktalama işareti ile gerçekleşmiştir. Derlem üzerinde yapılan noktalama işaretlerine özel dönüşüm sonucunda, c) tümcesi için d) satırında görülen ayrıştırma oluşmaktadır. Burada noktalama işaretine bağlı olan "kalemi" sözcüğü koparılarak, noktalama işaretinin iyesi olan "defteri" sözcüğüne doğrudan bağlanmıştır. Bağlılık türü de değiştirilerek, noktalama işaretinin bağlılık türü kullanılmıştır. Noktalama işareti ise ağaçta bağlanabileceği en yüksek nokta olan "defteri" sözcüğüne bağlanmış ve bağlılık türü olarak PUNC türü kullanılmıştır.

Ortak çalışmada kullanılan derlemlerin birbirleri ile tutarlı olması ve noktalama işaretlerinin tamamen değerlendirme dışı bırakılabilmesi amacıyla gerçekleştirilen bu dönüşüm, tamamen aynı yapıda olan tümcelerin farklı şekillerde ayrıştırılmasına yol açmaktadır. (Şekil 4.8 a) ve c)'deki tümceler b) ve d)'ye dönüşerek farklılaşmışlardır.)

Bu durum ayrıştırıcı açısından hem olumlu hem de olumsuz bir etki yaratarak genel başarımı çok etkilememesine rağmen, derlem içi tutarlılığın sağlanabilmesi açısından düzeltilmesi gereken bir durumdur. Şekilde, d) satırındaki gösterim aynı biçimbilimsel özelliklere (+Noun+A3sg+Pnon+Acc) sahip iki sözcüğü doğrudan bir bağlılık ile birleştirdiği için, ayrıştırıcı için öğrenmesi daha kolay bir yapı oluşturmaktadır. Ancak b) satırında olduğu gibi benzer yapıların esas gösterimde kalması, aynı durum için birden çok farklı örnek oluşturarak ayrıştırıcının bu tür durumları ayırt edebilmesini zorlaştırmaktadır.

4.8.2 Değerlendirme

Ortak çalışmada değerlendirme sırasında, ana başarım ölçütü olarak etiketli bağlanma başarımı ($\acute{C}KB_E$) alınmış ve sıralama buna göre yapılmıştır. Ortak çalışmada farklı gruplar tarafından elde edilen sonuçlar, Türkçe için %37,8 (en kötü) ile Japonca için %91,7 (en iyi) arasında değişmektedir. Diller için elde edilen ortalama başarımlar %56,0 (Türkçe için) ile %85,9 (Japonca için) arasında, en yüksek başarımlar ise %65,7 (Türkçe için) ile %91,7 (Japonca için) arasında değişmektedir. Çıkan bu sonuçlar neticesinde, ayrıştırması en kolay derlem Japonca Verbmobil derlemi olarak görülmüştür (Buchholz ve Marsi, 2006). Bu durum Japonca'nın ayrıştırılması basit bir dil olduğu anlamına gelmemektedir. Derlemlerin özelliklerine bakıldığında, Japonca derlemin tek tür metinden (iş randevusu diyalogları) oluştuğu ve sadece yedi farklı türde bağlılık içerdiği görülmektedir. Ayrıca diyaloglardan oluşan bu derlemde, bazı tümcelerin "Evet", "Hayır" gibi çok kısa tümcelerden oluştuğu ve genelde aynı tür sözcükler kullanıldığı görülmektedir. Öte yandan, ortak çalışmanın işlenmesi en zor derlemi olarak görülen Türkçe derlem (Buchholz ve Marsi, 2006), sekiz farklı türden metin ve yirmibeş farklı bağlılık türü içermektedir. Ayrıca, sınama verisinde yer alan yeni sözcük (eğitim verisinde görülmeyen) oranı en yüksek dillerden biridir.[22]

Tablo 4.11 çalışmaya katılan onyedi grubun Türkçe Derlem üzerindeki ayrıştırıcı başarımlarını vermektedir. Önceki bölümlerde tanıtılan sınıflandırıcı tabanlı

[22] %41,4 yeni sözcük görülmesi ile en yüksek orana sahip dil, %13,2 yeni gövde görülmesi oranı ile ikinci en yüksek orana sahip dil.

ayrıştırıcımız Türkçe üzerinde en yüksek başarımı sağlamıştır (*ÇKB* = 75,8 ve $ÇKB_E$ = 65,7). (Nivre ve diğ., 2006b)[23]

Tablo 4.11: Conll-X Ortak Çalışması Türkçe Bölümü Sonuçları

Katılımcılar	*ÇKB*	$ÇKB_E$
Nivre ve diğ.	**75,8**	**65,7**
Johansson ve Nugues	73,6	63,4
McDonald ve diğ.	74,7	63,2
Corston-Oliver ve Aue	73,1	61,7
Cheng ve diğ.	74,5	61,2
Chang ve diğ.	73,2	60,5
Yüret	71,5	60,3
Riedel ve diğ.	74,1	58,6
Carreras ve diğ.	70,1	58,1
Wu ve diğ.	69,3	55,1
Shimizu	68,8	54,2
Bick	65,5	53,9
Canisius ve diğ.	64,2	51,1
Schiehlen ve Spranger	61,6	49,8
Dreyer ve diğ.	60,5	46,1
Liu ve diğ.	56,9	41,7
Attardi	65,3	37,8

Tablo 4.11'deki sonuçlarda, başarımı ortalamanın (%55,4) altına düşen grupların ortak özelliği hepsinin Türkçe için çok önemli yeri olan biçimbilimsel özelliklerin kullanımını gözardı etmeleridir.[24] Johansson ve Nugues (2006) ve Yüret (2006) ortak çalışmadaki genel başarımlarına göre Türkçe bölümünde çok daha yüksek bir başarım (gruplar arası sıralamada +7 sıra önde) göstermişlerdir. Bu ayrıştırıcıların özelliklerine bakıldığında her iki ayrıştırıcının da biçimbilimsel özellikleri küçük parçalara bölerek (bknz Bölüm 3.3.4) işlediği görülmektedir. Yüksek başarım elde eden grupların kullandığı ayrıştırma algoritmalarına bakıldığında, bunların büyük çoğunlukla Eisner

[23]Geliştirilen ayrıştırıcı aynı zamanda ortak çalışmada Türkçe dışındaki tek sağa bağımlı türdeki dil olan Japonca için ve İsveççe için diğer tüm sonuçlardan istatistiksel olarak belirgin farkla en iyi sonucu üretmiştir. Bunun dışında diğer altı farklı dil için üretilen sonuçlarda da en iyi küme ile arada istatistiksel olarak belirgin bir fark bulunmamaktadır.

[24]Bu küme içerisinde iki ayrıştırıcıda (Bick, 2006; Attardi, 2006) biçimbilimsel özelliklerin bir kısmı bazı durumlarda kullanılmaya çalışılmıştır.

(1996), Nivre (2003) ve Yamada ve Matsumoto (2003)'nun algoritmalarından biri olduğu görülmektedir. Kullanılan öğrenme yöntemleri ise çoğunlukla KDM tarzı aralık büyükleme sınıflandırıcılarıdır.

Ayrıştırıcılarda bağlılık türleri bulunmaya çalışılırken farklı yaklaşımlar benimsenmiştir. Bunlar:

- önce iye birimleri bulmak,
- önce bağlılık türlerini bulmak,
- sıradaki birime geçmeden önce üzerinde bulunulan birimin hem iye birimini hem de bağlılık türünü bulmak
  - bu işlemi yaparken ikisini aynı anda bulmak veya
  - önce bağlılığı sonra türünü bulmak

olarak sıralanabilir. Bölüm 3.3'de anlatıldığı üzere, sınıflandırıcı tabanlı ayrıştırıcı her adımda bağlılıkları belirlerken hem bağlılığı hem de türünü aynı anda belirlemektedir. İlk aşamada, sınıflandırıcının ayırt etmesi gereken sınıf sayısını arttırarak, sınıflandırıcının işini karmaşıklaştırdığı izlenimini veren bu yaklaşımın, diğer yöntemlere göre daha başarılı olduğu düşünülmektedir. Katılımcıların kullandığı yöntemler ile ilgili ayrıntılı bilgiye Conll-X konferans kitabından ulaşılabilir.

4.9 Bölüm Sonucu

Bu bölümde, tez kapsamında tanıtılan farklı türde ayrıştırıcıların eniyileştirilmiş halleri sunulmuş ve bunlar üzerinde yapılan değerlendirmeler verilmiştir. Ayrıştırıcılarda biçimbilimsel özellikleri ve görünüm bilgilerini kullanmanın etkisi ayrıntılı olarak incelenmiş ve her iki kullanımın da Türkçe'nin ayrıştırmasındaki başarıma çok önemli katkısı olduğu gösterilmiştir. Türkçe'nin ayrıştırmasının farklı yöntemler ile gerçekleştirildiği Conll-X ortak çalışmasında, biçimbilimsel özellikleri kullanmayan grupların başarımlarının ortalamanın altında kaldığı görülmüştür. Tez kapsamında yapılan incelemelerde, bu özellikleri kullanmanın etkisi gözardı edilemezken aynı zamanda bazı sistemler için özellikleri kısmi olarak kullanmanın daha faydalı

olacağı değerlendirilmektedir. Farklı bağlılık türleri üzerinde yapılan başarım ölçümlerinde, uydu ve iye birimlerin birbirlerine uzak konumlarda bulunduğu bağlılıkların başarımlarının daha düşük olduğu gösterilmiştir.

Eğitim kümesi boyutunun başarıma etkisinin incelenmesi sonucunda, özellikle görünüm bilgilerini kullanan en iyi modelimizin eğitim kümesi boyutunun artması ile birlikte başarımının da arttığı gösterilmiştir. Derlem boyutunun, özellikle farklı sözcük gövdeleri içerecek şekilde arttırılmasının, yeni karşılaşılan tümcelerin başarımında artışa neden olacağı öngörülmektedir.

5. SONUÇLAR VE ÖNERİLER

Bağlılık ayrıştırması, tümceyi oluşturan birimler arasında bağlılık ilişkileri kurmayı amaçlayan bir tümce çözümlemesi yöntemidir. Sözcükler arası ikili bağlılıklık ilişkilerinin ayrıştırmanın başarımındaki önemli etkisinin anlaşılması ile birlikte, son yıllarda bağlılık ayrıştırması konusuna olan ilgi gittikçe artmıştır. Veri güdümlü araştırmalar için insan tarafından çözümlemesi yapılmış derlemlere gereksinim duyulmasından ötürü, çalışmalar ilk olarak derlemleri hazır olan diller üzerinde başlamıştır. Bu diller İngilizce gibi ayrıştırma konusunda üzerinde yoğun olarak çalışılmış dillerdir. Ancak bu diller için geliştirilen modellerin, yapıları farklı olan diller üzerinde aynı başarımları sağlamadıkları saptanmıştır. Bu farklı diller için yeni modeller geliştirilmesine gereksinim duyulmaktadır.

Türkçe tümce içi öğe dizilişleri serbest, çok zengin bitişken biçimbilimsel yapıda olan bir dildir. Türkçe ağaç yapılı derlemin yakın zamanda kullanıma açılması ile birlikte derlem üzerinde yapılan çalışmalar da hız kazanmıştır. Bu tez çalışmasında, yukarıdaki özellikleri ile ilgili yayınlarda üzerinde yoğun olarak çalışılmış dillerden farklılık gösteren Türkçe'nin bağlılık çözümlemesi konusunda araştırmalar yapılmıştır. İncelemeler sonucunda biçimbilimsel özelliklerin ve görünüm bilgilerinin bu tür bitişken bir dilin ayrıştırmasında vazgeçilemez unsurlar olduğu görülmüştür. Aynı zamanda, bu özelliklerden faydalanan veri güdümlü gerekirci bir ayrıştırıcı ve çekim kümelerinin ayrıştırma birimleri olarak kullanılmasıyla Türkçe'nin ayrıştırma başarımında önemli bir artış elde edilebileceği gösterilmiştir. Bu teknikleri kullanarak oluşturulan ayrıştırıcı ile Türkçe derlem üzerinde benzer çalışmalar içerisindeki en yüksek başarımlar elde edilmiştir.

Biçimbilimsel özelliklerin ve görünüm bilgilerinin, zengin biçimbilim ve serbest sözcük dizilişine sahip diller için ayrıştırmanın başarımını arttırabileceğinin gösterilmesinin yanı sıra, deneyler bu etkinin farklı sınıflar üzerinde düzgün bir dağılım göstermediğini belirtmektedir. Bu tezde başlatılan türde ayrıntılı incelemelerin, ilgili

yayınlarda rastlanan, görünüm bilgilerinin (özellikle farklı diller üzerindeki) etkisi ile ilgili çelişkili sonuçlara ışık tutacağına inanılmaktadır.

Önerilen yöntemler farklı ayrıştırıcılar üzerinde denenmiş ve etkileri incelenmiştir. Tez sırasında geliştirilen olasılık tabanlı ayrıştırıcı Türkçe'nin veri güdümlü bağlılık ayrıştırması konusunda yapılan ilk çalışma olma niteliğindedir. Karşılaştırmalar için geliştirilen kural tabanlı ayrıştırıcı ile beraber Türkçe'nin bağlılık ayrıştırması için yapılacak çalışmalara önemli bir temel çizgi oluşturmaktadırlar.

Tez süresince, Türkçe derlemde bulunan hatalar düzeltilerek bu derlemin yeni bir sürümü araştırmacıların hizmetine sunulmuştur. Ancak geliştirilen ayrıştırıcıların başarımlarının artırılabilmesi için derlem boyutunun arttırılması ve bu düzeltmelere devam edilmesi gerekmektedir.

Özetle, bu çalışmayla bilime yapılmış katkılar şunlardır:

- Türkçe'nin bağlılık ayrıştırmasında, ayrıştırma birimi olarak sözcüklerden daha küçük olan çekim kümelerini kullanmanın ayrıştırma başarımını arttırdığı gösterilmiştir.
- Biçimbilimsel özellikleri kullanmanın, Türkçe'nin bağlılık ayrıştırması başarımında önemli artışa neden olduğu ve bunun yanısıra bu özelliklerin hangilerinin, ayrıştırma için değerli bilgi taşıdığı gösterilmiştir.
- Görünüm bilgisi özelliklerini kullanmanın, Türkçe'nin bağlılık ayrıştırması başarımında önemli artışa neden olduğu gösterilmiştir.
- Yukarıda sıralanan yaklaşımların birleştirilmesi ile oluşturulan sınıflandırıcı tabanlı ayrıştırıcı ile Türkçe derlem üzerindeki en yüksek başarım elde edilmiştir.

Bu tezde geliştirilen ayrıştırıcıda ve Türkçe üzerinde yüksek başarım gösteren diğer bağlılık ayrıştırıcılarında bağlılıklar bulunurken bunların tümce içi bağlılıklardan bağımsız oldukları varsayılmıştır. Geçmişe dayalı modellerin kurulan kısmi ağacın bağlılıklarını özellik olarak kullanmalarına karşın, bu modeller bile ağacın bütünü üzerinde dilin bağlılık yapısına uygun kısıtlar getirmemekte ve sadece komşu birimlerin bağlılıklarını kullanmaktadırlar. Bu tür kısıtlar yeni ayrıştırma algoritmalarının tasarlanması ihtiyacını doğurmaktadır. Bu gereksinim sadece

Türkçe'ye özel olmayıp tüm diller için geçerlidir. Bağlılık ağacı üzerinde kısıtlar koyan yeni algoritmalar tasarlamak gelecekte üzerinde çalışılması gereken önemli araştırma konularından biri olarak görülmektedir.

Gelecek çalışma olarak önerilen araştırma konularından bir diğeri de, denetimli öğrenme yöntemlerinin yanısıra yarı denetimli öğrenme yöntemleri üzerinde incelemeler yapılmasıdır. Derlem geliştirmenin çok maliyetli bir iş olması nedeni ile, işaretlenmesi yapılmamış doğal dil metinleri kullanılarak ayrıştırıcıların başarımlarının nasıl arttırılabileceği konusunda çalışmalar yapılmalıdır.

KAYNAKLAR

Afonso, S., Bick, E., Haber, R. ve Santos, D., 2002. "Floresta sintá(c)tica": a treebank for Portuguese, *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, Canary Island, 29–31 May, 1698–1703.

Arun, A. ve Keller, F., 2005. Lexicalization in crosslinguistic probabilistic parsing: the case of French, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, MI, 25-30 June, 302–313.

Atalay, NB., Oflazer, K. ve Say, B., 2003. The annotation process in the Turkish treebank, *Proceedings of the 4th International Workshop on Linguistically Interpreteted Corpora (LINC)*, Budapest, 13-14 April, ?

Attardi, G., 2006. Experiments with a multilanguage non-projective dependency parser, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 166–170.

Bick, E., 2006. Lingpars, a linguistically inspired, language-independent machine learner for dependency treebanks, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 171–175.

Bikel, DM., 2004. A distributional analysis of a lexicalized statistical parsing model, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, 25-26 June, 182–189.

Bikel, DM. ve Chiang, D., 2000. Two statistical parsing models applied to the Chinese treebank, *Proceedings of the 2nd Chinese Language Processing Workshop*, Hong Kong, 7-8 October, 1–6.

Black, E., Jelinek, F., Lafferty, JD., Magerman, DM., Mercer, RL. ve Roukos, S., 1992. Towards history-based grammars: Using richer models for probabilistic parsing, *Proceedings of the 5th DARPA Speech and Natural Language Workshop*, New York, NY, 23-26 February, 31–37.

Bosco, C., 2004. A grammatical relation system for treebank annotation. *Ph.D. thesis*, University of Torino, Torino.

Bozşahin, C., 1996. Ulamsal dilbilgisi ve Türkçe, *Dilbilim Araştırmaları*, **7**(1), 230–244.

Bozşahin, C., 2000. Gapping and word order in Turkish, *Proceedings of the 10th International Conference on Turkish Linguistics*, Istanbul, 16-18 August, 58–66.

Bozşahin, C., 2002. The combinatory morphemic lexicon, *Computational Linguistics*, **28**(2), 145–186.

Brants, S., Dipper, S., Hansen, S., Lezius, W. ve Smith, G., 2002. The TIGER treebank, *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT)*, Sozopol, 20-21 September, ?

Buchholz, S. ve Marsi, E., 2006. Conll-X shared task on multilingual dependency parsing, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 149–164.

Çakıcı, R., 2005. Automatic induction of a CCG grammar for Turkish, *Proceedings of the student research workshop of 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, MI, 73–78.

Canisius, S., Bogers, T., van den Bosch, A., Geertzen, J. ve Sang, ETK., 2006. Dependency parsing by inference over high-recall dependency predictions, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 176–180.

Carreras, X., Surdeanu, M. ve Marquez, L., 2006. Projective dependency parsing with perceptron, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 181–185.

Cebiroğlu, G. 2002. Sözlüksüz köke ulaşma yöntemi, Master's thesis, İstanbul Teknik Üniversitesi.

Çetinoğlu, Ö. ve Oflazer, K., 2006. Morphology-syntax interface for Turkish LFG, *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (Coling/ACL)*, Sydney, 17-21 July, 153–160.

Chang, CC. ve Lin, CJ., 2001. LIBSVM: A Library for Support Vector Machines. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Chang, MW., Do, Q. ve Roth, D., 2006. A pipeline model for bottom-up dependency parsing, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 186–190.

Charniak, E., 2000. A maximum-entropy-inspired parser, *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Seattle, Washington, 132–139.

Cheng, Y., Asahara, M. ve Matsumoto, Y., 2006. Multi-lingual dependency parsing at naist, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 191–195.

Chomsky, N., 1957. Syntactic Structures, Mouton, Berlin.

Chung, H. ve Rim, HC., 2004. Unlexicalized dependency parser for variable word order languages based on local contextual pattern, *Proceedings of the 5th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, Seoul, 15-21 February, 109–120.

Civit Torruella, M. ve Martí Antonín, MA., 2002. Design principles for a Spanish treebank, *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT)*, Sozopol, 20-21 September, ?

Collins, M., 1996. A new statistical parser based on bigram lexical dependencies, *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, Santa Cruz, CA, 24-27 June, 184–191.

Collins, M., 1997. Three generative, lexicalised models for statistical parsing, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, Madrid, 7-12 July, 16–23.

Collins, M., 1999. Head-driven statistical models for natural language parsing. *Ph.D. thesis*, University of Pennsylvania, Philadelphia, PA.

Collins, M., Hajic, J., Ramshaw, L. ve Tillmann, C., 1999. A statistical parser for Czech, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, University of Maryland, 20–26 June, 505–518.

Corazza, A., Lavelli, A., Satta, G. ve Zanoli, R., 2004. Analyzing an Italian treebank with state-of-the-art statistical parsers, *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT)*, Tübingen, 10-11 December, 39–50.

Corston-Oliver, S. ve Aue, A., 2006. Dependency parsing with reference to Slovene, Spanish and Swedish, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 196–200.

Daelemans, W. ve Bosch, AV., 2005. Memory-Based Language Processing, Cambridge University Press, Cambridge.

Dreyer, M., Smith, DA. ve Smith, NA., 2006. Vine parsing and minimum risk reranking for speed and precision, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 201–205.

Dubey, A. ve Keller, F., 2003. Probabilistic parsing for German using sister-head dependencies, *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, 7-12 July, 96–103.

Dzeroski, S., Erjavec, T., Ledinek, N., Pajas, P., Zabokrtsky, Z. ve Zele, A., 2006. Towards a Slovene dependency treebank, *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, 24-26 May, ?

Eisner, J., 1996. Three new probabilistic models for dependency parsing: An exploration, *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, 5-9 August, 340–345.

Eryiğit, G., Adalı, E. ve Oflazer, K., 2006a. Türkçe cümlelerin kural tabanlı bağlılık analizi, *Proceedings of the 15th Turkish Symposium on Artificial Intelligence and Neural Networks*, Muğla, 21-24 June, 17–24.

Eryiğit, G., Nivre, J. ve Oflazer, K., 2006b. The incremental use of morphological information and lexicalization in data-driven dependency parsing, *Proceedings of the 21st International Conference on the Computer*

Processing of Oriental Languages (ICCPOL), Singapore, 17-19 December, accepted for publication.

Eryiğit, G. ve Oflazer, K., 2006. Statistical dependency parsing of Turkish, *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, 3-7 April, 89–96.

Gildea, D., 2001. Corpus variation and parser performance, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Pittsburgh, PA, 3-4 June, 167–202.

Güngör, T., 2004. Generation of sentence parse trees using parts of speech, *Proceedings of the Advances in Artificial Intelligence, 27th Annual German Conference on AI (KI)*, Ulm, 20-24 September, 56–66.

Güngördü, Z. ve Oflazer, K., 1994. Parsing turkish using the lexical functional grammar formalism, *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, Kyoto, 5-9 August, 494–500.

Hajic, J., Smrz, O., Zemánek, P., Snaidauf, J. ve Beska, E., 2004. Prague Arabic dependency treebank: Development in data and tools, *Proceedings of the International Conference on Arabic Language Resources and Tools*, Cairo, 22-23 September, 110–117.

Hajič, J., Hajičová, E., Pajas, P., Panevová, J., Sgall, P. ve Hladká, B., 2001. Prague dependency treebank 1.0 (final production label). CDROM CAT: LDC2001T10., ISBN 1-58563-212-0.

Hakkani-Tür, D., Oflazer, K. ve Tür, G., 2002. Statistical morphological disambiguation for agglutinative languages, *Journal of Computers and Humanities*, **36**(4), 381–410.

Haruno, M., Shirai, S. ve Ooyama, Y., 1998. Using decision trees to construct a practical parser, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistics*, San Francisco, California, 10-14 August, 505–512.

Hengirmen, M., 2005. Türkçe Dilbilgisi, Engin Yayınevi, Ankara.

Hoffman, BA., 1995. The computational analysis of the syntax and interpretation of free word order in turkish. *Ph.D. thesis*, University of Pennsylvania, Philadelphia, PA.

Huang, CR., Chen, FY., Chen, KJ., ming Gao, Z. ve Chen, KY., 2000. Sinica treebank: design criteria, annotation guidelines, and on-line interface, *Proceedings of the 2nd Workshop on Chinese Language Processing*, Morristown, NJ, 7-8 October, 29–37.

Johansson, R. ve Nugues, P., 2006. Investigating multilingual dependency parsing, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 206–210.

Jurafsky, D. ve Martin, JH., 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall, New Jersey.

Kawata, Y. ve Bartels, J., 2000. Stylebook for the Japanese treebank in VERBMOBIL, Verbmobil-Report **240**, Seminar für Sprachwissenschaft, Universität Tübingen.

Klein, D. ve Manning, CD., 2003. Accurate unlexicalized parsing, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, 7-12 July, 423–430.

Kromann, MT., 2003. The Danish dependency treebank and the DTAG treebank tool, *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*, Vaxjo, 14-15 November, 117–128.

Kudo, T. ve Matsumoto, Y., 2000. Japanese dependency analysis based on support vector machines, *Joint Sigdat Conference On Empirical Methods In Natural Language Processing and Very Large Corpora*, Hong Kong, 7-8 October, ?

Kudo, T. ve Matsumoto, Y., 2002. Japanese dependency analysis using cascaded chunking, *Proceedings of the 6th Conference on Computational Natural Language Learning (CoNLL-2002)*, Taipei, 31 August-1 September, 63–69.

Lepage, Y., Shin-Ichit, A., Susumu, A. ve Hitoshi, I., 1998. An annotated corpus in japanese using Tesniere's structural syntax, *Proceeding of the Content Visualization and Intermedia Representations COLING-ACL'98*, Montreal, 10-14 August, ?

Levy, R. ve Manning, C., 2003. Is it harder to parse Chinese, or the Chinese treebank?, *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, 7-12 July, 439–446.

Liu, T., Ma, J., Zhu, H. ve Li, S., 2006. Dependency parsing based on dynamic local optimization, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 211–215.

Magerman, DM., 1995. Statistical decision-tree models for parsing, *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Cambridge, MA, 26-30 June, 276–283.

Marcus, MP., Santorini, B. ve Marcinkiewicz, MA., 1993. Building a large annotated corpus of English: The Penn Treebank., *Computational Linguistics*, **19**(2), 313–330.

McDonald, R., Pereira, F., Ribarov, K. ve Hajic, J., 2005a. Non-projective dependency parsing using spanning tree algorithms, *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing(HLT-EMNLP)*, Vancouver, 6-8 October, 523–530.

McDonald, R., Crammer, K. ve Pereira, F., 2005b. Online large-margin training of dependency parsers, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, MI, 25-30 June, 91–98.

McDonald, R., Lerman, K. ve Pereira, F., 2006. Multilingual dependency analysis with a two-stage discriminative parser, *Proceedings of the 10th*

Conference on Computational Natural Language Learning (CoNLL-X), New York, NY, 8-9 June, 216–220.

Nilsson, J., Hall, J. ve Nivre, J., 2005. MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity, *Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA) Special Session on Treebanks*, Joensuu, 20-21 May, ?

Nivre, J., 2003. An efficient algorithm for projective dependency parsing, *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, Nancy, 23-25 April, 149–160.

Nivre, J., 2006a. Inductive Dependency Parsing, Springer.

Nivre, J., 2006b. Inductive dependency parsing. *Ph.D. thesis*, Växjö University, Sweden.

Nivre, J., Hall, J. ve Nilsson, J., 2004. Memory-based dependency parsing, *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL-2004)*, Boston, MA, 6-7 May, 49–56.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryiğit, G., Kübler, S., Marinov, S. ve Marsi, E., 2006a. Maltparser: A language-independent system for data-driven dependency parsing, *Accepted for publication in Natural Language Engineering Journal*.

Nivre, J., Hall, J., Nilsson, J., Eryiğit, G. ve Marinov, S., 2006b. Labeled pseudo-projective dependency parsing with support vector machines, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 221–225.

Nivre, J. ve Nilsson, J., 2003. Three algorithms for deterministic dependency parsing, *14th Nordic Conference of Computational Linguistics (NODALIDA)*, Reykjavik, 30-31 May, ?

Nivre, J. ve Nilsson, J., 2005. Pseudo-projective dependency parsing, *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, MI, 25-30 June, 99–106.

Nivre, J. ve Scholz, M., 2004. Deterministic dependency parsing of english text, *The 20th International Conference on Computational Linguistics (COLING)*, Geneva, 23-27 August, 64–70.

Oflazer, K., 1994. Two-level description of Turkish morphology, *Literary and Linguistic Computing*, **9**(2), 137–148.

Oflazer, K., 2003. Dependency parsing with an extended finite-state approach, *Computational Linguistics*, **29**(4), 515–544.

Oflazer, K., Say, B., Hakkani-Tür, DZ. ve Tür, G., 2003. Building a Turkish treebank, In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*. Kluwer, Dordrecht/Boston/London, 261–277.

Ratnaparkhi, A., 1997. A linear observed time statistical parser based on maximum entropy models, *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Providence, RI, 1-2 August, 1–10.

Riedel, S., Çakıcı, R. ve Meza-Ruiz, I., 2006. Multi-lingual dependency parsing with incremental integer linear programming, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 226–230.

Sagae, K. ve Lavie, A., 2005. A classifier-based parser with linear run-time complexity, *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, Vancouver, 9-10 October, 125–132.

Say, B., 2004. Odtü-Sabancı Türkçe ağaç yapılı derlemi kullanma kılavuzu.

Schiehlen, M. ve Spranger, K., 2006. Language independent probabilistic context-free parsing bolstered by machine learning, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 231–235.

Şehitoğlu, O. ve Bozşahin, C., 1996. Morphological productivity in the lexicon, *Proceedings of the ACL SIGLEX Workshop on Breadth and Depth of Semantic Lexicons*, Santa Cruz, 4 August, 153–160.

Sekine, S., Uchimoto, K. ve Isahara, H., 2000. Backward beam search algorithm for dependency analysis of Japanese, *Proceedings of the 18th Conference on Computational linguistics*, Morristown, NJ, USA, 31 July - 4 August, 754–760.

Shimizu, N., 2006. Maximum spanning tree algorithm for non-projective labeled dependency parsing, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 236–240.

Simov, K., Popova, G. ve Osenova, P., 2002. HPSG-based syntactic treebank of Bulgarian (BulTreeBank), In Andrew Wilson, Paul Rayson, and Tony McEnery, editors, *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*. Lincom-Europa, Munich, 135–142.

Tapanainen, P. ve Järvinen, T., 1997. A non-projective dependency parser, *Proceedings of the 5th conference on Applied natural language processing*, San Francisco, CA, USA, 64–71.

Tesnière, L., 1959. Eléments de syntaxe structurale, Klincksieck, Paris.

Uchimoto, K., Sekine, S. ve Isahara, H., 1999. Japanese dependency structure analysis based on maximum entropy models, *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Bergen, Norway, 196–203.

van der Beek, L., Bouma, G., Malouf, R. ve van Noord, G., 2002. The Alpino dependency treebank, *Proceedings of the 12th Meeting of Computational Linguistics in the Netherlands (CLIN)*, Enschede, ?

Vapnik, VN., 1995. The Nature of Statistical Learning Theory, Springer, New York, NY.

Veenstra, J. ve Daelemans, W., 2000. A memory-based alternative for connectionist shift-reduce parsing, *Technical Report,* **ILK-0012**, Tilburg University, Tilburg.

Wu, YC., Lee, YS. ve Yang, JC., 2006. The exploration of deterministic and efficient dependency parsing, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 241–245.

Yamada, H. ve Matsumoto, Y., 2003. Statistical dependency analysis with support vector machines, *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, Nancy, 23-25 April, 195–206.

Yüret, D., 2006. Dependency parsing as a classification problem, *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, New York, NY, 8-9 June, 246–250.

Yüret, D. ve Türe, F., 2006. Learning morphological disambiguation rules for Turkish, *Proceedings of the Human Language Technology conference and North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL)*, New York, NY, 5-7 June, 328–334.

A. EK: Kural Tabanlı Ayrıştırıcılarda Kullanılan Kurallar

Tablo A.1: Kurallar, Uygulanış Sayıları ve Derlemin Bütünü Üzerindeki Başarımları

Kurallar	**n**	***ÇKB***	***SB***
◇ "dA" ekleri, kendilerinden önce gelen *ÇK*'ye bağlanırlar.	785	91.1	97.6
◇ "ki" eki, kendisinden önce gelen *ÇK*'ye bağlanır.	112	78.6	91.1
◇ "mU" gövdesine sahip soru ekleri, kendilerinden önce gelen *ÇK*'ye bağlanırlar.	223	91.0	98.2
◇ "değil" gövdesine sahip bağlaçlar, kendilerinden önce gelen *ÇK*'ye bağlanırlar.	35	85.7	94.3
◇ Zarflar, hemen sağındaki sözcüğün edat olan *ÇK*'sine bağlanırlar.	30	90.0	90.0
◇ Zarflar, sıradaki, bulunduğu sözcüğün eylem türündeki son *ÇK*'si olan *ÇK*'ye bağlanırlar.	2149	61.6	70.5
◇ Her sözcük, hemen sağındaki,öncesinde veya sonrasında noktalama işareti bulunmayan, "ama, ancak, ya, ve, veya, ile, yada, hem" sözcüklerine veya virgüle bağlanır.	740	95.7	95.7
◇ Tamlayan durumdaki isim, sıradaki, iyelik eki almış *ÇK*'ye bağlanır.	1347	84.9	87.8
◇ Tamlayan durumdaki isim, aynı sözcük içerisinde kendinden sonra iyelik eki almış bir *ÇK* bulunana sıradaki eylem *ÇK*'ye bağlanır.	530	51.9	94.3
◇ Yalın durumdaki isim, hemen sağındaki sözcüğün iyelik eki almış *ÇK*'sine bağlanırlar.	2126	87.8	90.5
◇ İsimler ve zamirler, sıradaki eylem *ÇK*'ye bağlanırlar.	12463	60.6	80.7
◇ İsimler ve zamirler, hemen sağındaki sözcüğün edat *ÇK*'sine bağlanırlar.	893	96.3	97.9
◇ Eylemler, sıradaki noktalama işaretine bağlanırlar.	6365	90.6	90.6
◇ Eylemler, hemen sağındaki sözcüğün edat *ÇK*'sine bağlanırlar.	70	91.4	94.3
◇ Eylemler, sıradaki, bulunduğu sözcüğün eylem türündeki son *ÇK*'si olan *ÇK*'ye bağlanırlar.	451	62.3	64.7
◇ Sayılar, sıradaki, bulunduğu sözcüğün son *ÇK*'si isim olmayan sayı *ÇK*'ye bağlanırlar.	39	92.3	97.4

Tablo A.1: devam

Kurallar	n	*ÇKB*	*SB*
◇ Sayılar, sıradaki isim *ÇK*'ye bağlanırlar.	543	87.3	93.7
◇ Sıfatlar, sıradaki isime ve zamire bağlanırlar.	3909	83.4	86.9
◇ "kadar" sözcüğü, sıradaki sıfat veya isim *ÇK*'ye bağlanır.	74	52.7	59.5
◇ "bir" sözcüğü, sıradaki zarf veya isim *ÇK*'ye bağlanır.	967	87.9	92.3
◇ Sıfatlar, sıradaki, bulunduğu sözcüğün eylem türündeki son *ÇK*'si olan *ÇK*'ye bağlanırlar.	614	61.4	85.5
◇ Sıfatlar, sıradaki edat *ÇK*'ye bağlanırlar.	78	94.9	94.9
◇ Belirteçler, sıradaki isim *ÇK*'ye bağlanırlar.	1013	88.7	90.7
◇ Edatlar, hemen sağındaki sözcüğün eylem *ÇK*'sine bağlanırlar.	417	73.9	95.0
◇ Birleştirme bağlaçları (ve, veya, ile, yada, hem, virgül), oluşmakta olan ağaçta iyesi olduğu bir *ÇK* varsa ve bu *ÇK* ile sıradaki *ÇK* uyum gösteriyorsa[a], sıradaki *ÇK*'ye bağlanır.	1393	78.5	82.3
◇ "daha, en, pek, çok, öyle" gövdesine sahip zarflar, hemen sağındaki sözcüğün sıfat *ÇK*'sine veya hemen sağlarındaki sözcük "dA, mU" ise sağ tarafındaki ikinci sözcüğün sıfat *ÇK*'sine bağlanırlar.	252	81.0	96.4
◇ Noktalama işaretleri, sıradaki *ÇK*'ye bağlanırlar.	3986	70.8	72.4
◇ Tümce içerisinde kendilerinden önce eylem bulunmayan isimler ve zamirler, sıradaki noktalama işareti cümle sonundaysa, bu noktalama işaretine bağlanırlar.	488	77.9	77.9
◇ Sözcükler, hemen sağ taraflarındaki noktalama işaretlerine bağlanırlar.	3231	33.8	33.8
◇ Yukarıdaki kurallar kullanılarak yığının üzerindeki ve sıradaki *ÇK* arasında bir bağlılık kurulamamışsa, sıradaki bulunduğu sözcüğün ilk *ÇK*'si olan isimler ve zamirler, eğer hemen sağ taraflarında bir noktalama işareti bulunuyorsa, kendilerinden önce gelen ilk eylem *ÇK*'ye bağlanırlar.	729	79.0	86.8
◇ Yukarıdaki kural sonucunda, sıradaki *ÇK*'nin bağlandığı yığındaki eylem *ÇK*'nin üzerinde yer alan *ÇK*'ler yine bu eylem *ÇK*'ye bağlanırlar.	83	47.0	50.6

[a]- İkisi de eylemse veya - ikisi de isimse ve durum ekleri aynıysa veya - ikisi de isimse ve ilk *ÇK* yalın haldeyse

B. EK: ODTÜ-Sabancı Türkçe Ağaç Yapılı Derlemi'nde Kullanılan Biçimbilimsel Terimlerin Açıklamaları

Derlemde yer alan sözcüklerin biçimbilimsel çözümlemeleri Oflazer (1994)'in çift yönlü biçimbilimsel çözümleyicisi tarafından yapılmıştır. Bu nedenle derlemde kullanılan notasyon bu çözümleyici ile aynıdır. Bu bölümde, tez içerisinde kullanılan biçimbilimsel terimlerin açıklamaları verilecektir. Kullanılan notasyon ile ilgili daha ayrıntılı bilgiye Say (2004), Oflazer (1994) veya "http://www.hlst.sabanciuniv.edu/TL/" kaynaklarından ulaşılabilir.

Tablo B.1: Biçimbilimsel Terimler

+Noun	: İsimler
+A3sg	: Kişi ve Sayı, 3. tekil
+A3pl	: Kişi ve Sayı, 3. çoğul
+Pnon	: İyelik, yok
+P3sg	: İyelik, 3. tekil
+Loc	: Hal, -de hali
+Gen	: Hal, sahip olma hali
+Nom	: Hal, yalın hali
+Pastpart	: Geçmiş zaman ortacı
+Adj	: Sıfatlar
+With	: -li isimden sıfat türetme eki
+Rel	: İlişkilendirici
+Adv	: Belirteçler
+Verb	: Eylemler
+Zero	: Ek almadan türetme
+Pos	: Olumlu
+Prespart	: Şimdiki zaman ortacı
+Pres	: Şimdiki zaman
+Cop	: Koşaç
+Become	: Oluşmak
+Caus	: Ettirgen
+A1sg	: 1. tekil kişi
+A3sg	: 3. tekil kişi

Derlemde bazı ana sınıf bilgileri alt sınıf bilgisine de sahiptirler. Tablo B.2 bu yapıyı göstermektedir. Derlemin Conll-X biçiminde, alt sınıf bilgisine sahip olmayan ana sınıflar için, alt sınıf bilgisi aynı isim verilerek belirtilmiştir.

Tablo B.2: Sınıf Bilgileri

Ana Sınıf	**Alt Sınıf**
Adj (Sıfat)	Adj (Sıfat) AFutPart (Gelecek Zaman Ortacı) APastPart (Geçmiş Zaman Ortacı) APresPart (Şimdiki Zaman Ortacı)
Adv (Belirteç)	Adv (Belirteç)
Conj (Bağlaç)	Conj (Bağlaç)
Det (Belirleyen)	Det (Belirleyen)
Dup (Tekrar)	Dup (Tekrar)
Interj (Ünlem)	Interj (Ünlem)
Noun (İsim)	Noun (İsim) NFutPart (Gelecek Zaman Ortacı) NInf (Mastar) NPastPart (Geçmiş Zaman Ortacı) NPresPart (Şimdiki Zaman Ortacı) Prop (Özel isim)
Num (Sayı)	Num (Sayı) Card (Miktar) Distrib (Üleştirme) Ord (Sıra) Range (Aralık) Real (Gerçek)
Postp (İlgeç)	Postp (İlgeç)
Pron (Adıl)	Pron (Adıl) DemonsP (İşaret) PersP (Kişi) ReflexP (Dönüşlü) QuesP (Soru)
Punc (Noktalama İşareti)	Punc (Noktalama İşareti)
Ques (Soru)	Ques (Soru)
Verb (Eylem)	Verb (Eylem)

Tablo B.3: Bağlılık Türleri

ABLATIVE.ADJUNCT	Çıkma (-den) Tümleç
APPOSITION	İlave açıklama
CLASSIFIER	Sınıflandırıcı
COLLOCATION	Eşdizimli öbekler
COORDINATION	Bağlaçlar
DATIVE.ADJUNCT	Yönelme (-e) Tümleç
DETERMINER	Belirleyen
EQU.ADJUNCT	Tarafından Tümleç
ETOL	Bileşik Eylemler
FOCUS.PARTICLE	Ayrı yazılan de da ise
INSTRUMENTAL.ADJUNCT	Yardımcı (ile) Tümleç
INTENSIFIER	Vurgulayıcı
LOCATIVE.ADJUNCT	Kalma (-de) Tümleci
MODIFIER	Niteleyici
NEGATIVE.PARTICLE	Olumsuzluk
OBJECT	Nesne
POSSESSOR	Sahipleyici
QUESTION.PARTICLE	Soru
RELATIVIZER	İlişkilendirici
S.MODIFIER	Söylemsel bağlılık
SENTENCE	Cümle
SUBJECT	Özne
VOCATIVE	Ünleme

C. EK: Derlem Üzerinde Yapılan Değişiklikler

"bir" sözcüğünün biçimbilimsel çözümlemesine ve bağlılık türüne dair uyumluluğun sağlanması

Hengirmen (2005)'in Türkçe Dilbilgisi kitabında sıfatlar niteleme ve belirtme sıfatları olmak üzere ikiye bölünmüşlerdir. Belirtme sıfatları, işaret, soru, belgisiz ve sayı sıfatları olmak üzere dörde ayrılırlar. Derlem oluşturulurken belgisiz sıfatların +Det sözcük sınıfı ile etiketlenmelerine ve DETERMINER bağlılık türü ile bağlanmalarına karar verilmiştir. Sayı sıfatları ise +Num ile etiketlenmiş ve MODIFIER bağlılık türü ile bağlanmışlardır. Derlemde "bir" sözcüğü +Det, +Num ve +Adj sınıflarıyla etiketlenmiş üç farklı şekilde karşımıza çıkmaktadır. Ancak derlemin farklı kişiler tarafından işaretlenmesi yapılırken hangi etiketin ve bağlılığın hangi durumda kullanılacağına dair uyumluluk korunamamıştır. Bu nedenle aşağıdaki mantıkla "bir" sözcüğünü içeren tüm tümceler taranarak düzeltilmiştir.

"bir" sözcüğü:

- Sayı sıfatı olarak kullanılıyorsa +Num ile etiketlenir ve MODIFIER bağlılık türü ile bağlanır
  Bu kadar parayla bir bisiklet alabildim.
- Belgisiz sıfat olarak kullanılıyorsa +Det ile etiketlenir ve DETERMINER bağlılık türü ile bağlanır
  Bugün beni bir kadın aradı.
- Yukarıdaki iki sınıfa girmeyen çok az örnekte +Adj ile etiketlenir ve MODIFIER bağlılık türü ile bağlanır
  Bu resimlerin hepsi bir.

"var" ve "yok" sözcüğünün biçimbilimsel belirsizlik giderimindeki uyumsuzluğun düzeltilmesi

Bu sözcükler ad soylu sözcüklerdir. Yüklem gibi kullanılır ve çoğu zaman ek eylemin dört kipiyle çekime girerler (Hengirmen, 2005). Asıl derlemde "var" ve "yok" sözcükleri için biçimbilimsel çözümleyicinin kendisinden de kaynaklanan ve ayrıştırmayı yapan kişilerin de etkisi ile oluşmuş büyük bir uyumsuzluk söz konusudur. Bu amaçla bu sözcükleri ve varyasyonlarını (vardır, yoktur vb..) içeren tüm tümceler elden geçirilerek aşağıdaki mantığa uygun biçimde değiştirilmişlerdir.

Biçimbilimsel çözümleyici "var" ve "yok" sözcükleri için aşağıdaki çözümlemeleri üretmektedir:

“yok”:

- yok+Adv
- yok+Adj

“var”:

- var+Adj
- var+Verb+ Pos+ Imp+ A2sg

Buradaki “var+Verb+Pos+Imp+A2sg” çözümlemesi “varmak” eyleminin 2.tekil kişi emir halini belirtmektedir.

Derlemde, iki sözcük için de sözcüklerin eylem olarak kullanılan hallerini belirtmek için (1,"var+Adj")(2,"Verb+Zero+A3sg") ve (1,"yok+Adj")(2,"Verb+Zero+A3sg") etiketleri kullanılmıştır. Ancak çoğu yerde bu karar uygulanamamış ve farklı hatalı girişler yapılmıştır. Düzeltme sonucunda, bu sözcüklerin eylem olarak kullanıldığı ve hatalı gösterildiği tüm tümcelerde ilgili değişiklik yapılmıştır.

Bu düzeltmelerden sonra dahi derlemde bu sözcüklerle ilgili hatalar görülmektedir ve düzeltilmeleri gerekmektedir. Buna örnek olarak “var” ve “vardır” sözcükleri gösterilebilir. Her iki sözcük de aynı şekilde kullanılmasına ve aynı anlamı ifade etmesine rağmen derlemin şu anki versiyonunda farklı etiketler ile gösterilmektedirler:

var : (1,"var+Adj")(2,"Verb+Zero+A3sg")

vardır : (1,"var+Adj")(2,"Verb+Zero+Pres+Cop+A3sg")

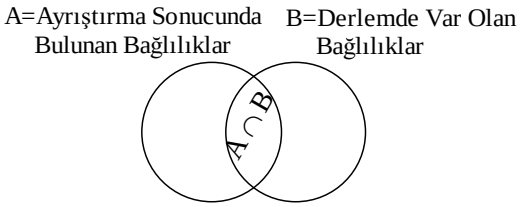
Noktalama işaretleri ile ilgili hataların bir kısmının düzeltilmesi

Derlemde noktalama işaretleri genel olarak bir yere bağlanmamış halde bulunmaktadırlar. Ancak bazı noktalama işaretleri bir baş sözcüğe bağlı ve kendisine bağlı bir uydu sözcük içerir halde bulunmaktadırlar. Bunlar genelde birleştirme veya dolaylı anlatım yapılarında görülmektedirler. Bunlara ek olarak, tümcenin ana eylemi de tümcenin en sonunda bulunan noktalama işaretine *SENTENCE* bağlılık türü ile bağlanmaktadır. Ancak derlemde noktalama işaretlerinin bağlanması ile ilgili uyumluluk maalesef yoktur. Bu nedenle öncelikle bu noktalama işaretlerinden kaynaklanan kopuk tümceler (noktalama işaretine bağlanan bir uydunun olduğu ancak akabinde bu noktalama işaretinin herhangi bir iye sözcüğe bağlanmadığı tümceler) düzeltilmiştir. Yaklaşık olarak derlemdeki her tümce noktalama işareti içerdiğinden, bu hataların tümünün düzeltilebilmesi için derlemin tümünün gözden geçirilmesi gerekmektedir.

R.SENTENCE bağlılık türünün kaldırılması

Asıl derlemde sadece altı defa kullanılan bu bağlılık türü değiştirilerek “SENTENCE” bağlılık türü yapılmıştır.

D. EK: Kesinlik, Geriçağırım, F ölçütü



P (Kesinlik[×]): Ayrıştırma sonucunda bulunan bağlılıkların ne kadarı doğru?

$$P = \frac{|A \cap B|}{|A|}$$

R (Gerigetirim[×]): Doğru olan bağlılıkların ne kadarı bulundu?

$$P = \frac{|A \cap B|}{|B|}$$

F Ölçütü: P ve R'nin harmonik ortası

$$F = \frac{2.P.R}{P+R} = \frac{2.|A \cap B|}{|B| + |A|}$$

E. EK: Terimler Sözlüğü

Bu bölümde tez içerisinde kullanılan Türkçe terimlerin literatürde yer alan İngilizce karşılıkları verilmektedir.

Ağaç Birleştiren Gramerler	Tree Adjoining Grammar
Anlamsal Bilgi	Semantics
Ayırdedici	Discriminative
Ayrıştırma	Parsing
Aradeğerlemek	Interpolate
Bağlamdan Bağımsız Gramer	Context Free Grammar
Bağlı	Connected
Bağlılık Çözümlemesi	Dependency Analysis
Bağlılık Ayrıştırması	Dependency Parsing
Basamaklı Birleştirme	Cascaded Chunking
Baş-sürümlü Öbek Yapısal Gramerlerdir	Head-Driven Phrase Structure Grammar
Bellek Tabanlı Öğrenme	Memory Based Learning
Biçimbilim	Morphology
Bilişimsel Dilbilim	Computational Linguistic
Bilgi Çıkarımı	Information Extraction
Birleşenli Ulamsal Gramer	Combinatory Categorial Grammar
Çekim Kümesi	Inflectional Group
Çekirdek Fonksiyonları	Kernel Functions
Denetimli Öğrenme	Supervised Learning
Doğal Dil Ayrıştırması	Natural Language Parsing
Doğal Dil İşleme	Natural Language Processing
Doğal Dil Öğrenmesi	Natural Language Learning
Dünya Bilgisi	Discourse
Düşürerek Düzleştirme	Backed-off smoothing
Düzgelenmiş	Normalized
Düzleştirme Algoritması	Smoothing Algorithm
En Büyük Bilgi Değeri	Maximum Entropy
En Büyük Olabilirlik Kestirimi	Maximum Likelihood Estimation
Eniyileştirme	Optimization
Etiketli	Labeled
Etiketsiz	Unlabeled
Geçmişe Dayalı	History-Based
Gerekirci	Deterministic
Gerigetirim	Recall
Geriye Doğru Demetli Arama	Backward Beam Search

Gramer Güdümlü	Grammar Driven
Görünüm Bilgisi	Lexical
Görünüm Bilgisi Ekleme	Lexicalization
Hevesli Yay	Arc-Eager
İkilileştirme	Binarization
İlgili Yayınlar	Literature
K En Yakın Komşu	K Nearest Neighborhood
Kapsamlı Öbek Yapısal Gramerler	Generalized Phrase Structure Grammar
Karar Destek Makineleri	Support Vector Machines
Kesinlik	Precision
Kesişmeyen	Projective
Kesişen	Non-projective
Koşullu	Conditional
Kullanım Bilgisi	Pragmatics
Makine Öğrenmesi	Machine Learning
Maksimum Kapsayan Ağaç	Maximum Spanning Tree
Mantıksal Tipli Gramer	Type Logical Grammar
Olağan Yay	Arc-Standard
Olasılık tabanlı	Probabilistic
Öbek Yapısal Gramer	Phrase Structure Grammar
Sağa bağımlı	Head-final
Sentaks Bilgisi	Syntax
Sesbilimi	Phonology
Sola bağımlı	Head-initial
Sözcük Etiketleyici	POS Tagger
Sözlüksel İşlevsel Gramerler	Lexical Functional Grammar
Tümevarımsal Çıkarım	Inductive Inference
Türetim Sınırı	Derivational Boundary
Uydu-İye	Dependent-Head veya Subordinate-Governor
Üretimsel	Generative
Üretimsel Dönüşümlü Dilbigisi	Generative Transformational Grammar
Veri Güdümlü	Data Driven
Yetkin	Gold-standard
10 Katlı Çapraz Doğrulama	10 Fold Cross Validation

ÖZGEÇMİŞ

Gülşen Cebiroğlu Eryiğit 1995 yılında Özel Saint-Michel Fransız Lisesinden mezun olmuştur. 2000 yılında Marmara Üniversitesi Bilgisayar Mühendisliği Bölümünden lisans derecesini, 2002 yılında İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği bölümünden yüksek lisans derecesini almış ve aynı bölümde araştırma görevlisi olarak doktora öğrenimine başlamıştır. 2000-2002 yılları arasında Garanti Teknoloji bursu ile "Türkçe Doğal Dil İşleme" konusunda başladığı araştırmalarına doktora çalışmaları süresince devam etmiştir. Bu süre zarfında "Tübitak Yurt İçi Doktora Burs Programı" tarafından desteklenmiştir. Ocak - Haziran 2006 tarihleri arasında İsveç Växjö Üniversitesi Dil Teknolojileri Grubu'nda, yine Tübitak (Yurt Dışı Araştırma Burs Programı) desteği ile, ziyaretçi araştırmacı olarak bulunmuştur.

Doktora çalışmaları sırasında konu ile ilgili çıkardığı yayınlarının listesi şöyledir:

Dergi Makaleleri:

- **Eryiğit, G.**, Nivre, J. and Oflazer, K., 2006. "Dependency Parsing of Turkish", Submitted to Computational Linguistics, MIT Press.
- Nivre, J., Hall, J., Nilsson, J., Chanev, A., **Eryiğit, G.**, Kübler, S., Marinov, S., and Marsi, E., 2006. "MaltParser: A Language-Independent System for Data-Driven Dependency Parsing", Accepted for publication in Natural Language Engineering Journal, Cambridge Press.

Konferans Bildirileri:

- **Eryiğit, G.**, and Oflazer, K., 2006. Statistical dependency parsing of Turkish. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, April.
- Nivre, J., Hall, J., Nilsson, J., **Eryiğit, G.** and Marinov, S., 2006. Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines. Proceedings of the Tenth Conference on Computational Natural Language Learning, New York, USA, June.
- **Eryiğit, G.**, Adalı, E. and Oflazer, K., 2006. Türkçe Cümlelerin Kural Tabanlı Bağlılık Analizi. In Proceedings of the 15th Turkish Symposium on Artificial Intelligence and Neural Networks, Muğla, Turkey, June.
- **Eryiğit, G.**, Nivre, J. and Oflazer, K., 2006. "The incremental use of morphological information and lexicalization in data-driven dependency parsing", Proceedings of the 21st International Conference on the Computer Processing of Oriental Languages, Sentosa, Singapore, December.