Türkçe Tümcelerin Sonunu Belirlemede Açık Kaynak / Ücretsiz Yazılımlar ve Performans Analizleri*

Özet: Tümce Sonu Belirleme Doğal Dil İşleme alanında önemli bir alt alan olarak pek çok araç ya da çalışmayla geniş bir yer tutmaktadır. Bu çalışmada Türkçe'de tümce sonu belirleme amacıyla kullanılan ya da kullanılabilecek çeşitli yazılımlar sınanacak ve başarımları değerlendirilecektir. Giriş ve alanyazının yer aldığı ilk bölümün ardından, belirtilen amaç için, Türkçe Ulusal Derlemi (TUD) [1] veritabanı kullanılarak oluşturulan alt-derlem ve çalışma kapsamında değerlendirilen yazılımlar tanıtılacaktır. Ardından, tümce sonu belirlenmiş, sınanmış ve iyileştirilmiş alt-derlemle karşılaştırılarak, sözkonusu yazılımların başarımları değerlendirilecektir. Sonuç bölümünde ise, ileride yapılacak benzer çalışmalar için önerilerde bulunulacaktır.

Anahtar Sözcükler: Doğal Dil İşleme, Tümce Sonu Belirleme, Noktalama İşaretleri, Belirginleştirme, Türkçe Ulusal Derlemi (TUD)

Open-source / free tools for detecting sentence boundaries in Turkish and their performance analyses.

Abstract: Sentence-boundary detection is an important part of Natural Language Processing with lots of tools and studies available in the literature. This study will analyze and evaluate the performances of the software currently used or to be used for sentence-boundary detection in Turkish. After a brief introduction and literature review, the sub-corpus derived from Turkish National Corpus (TNC) [1] for the above mentioned purpose and the software covered in the study will be presented. Then, by comparing the results of the given software on raw text, with the sentence-splitted and optimized sub-corpus, the analyses will be implemented. In the conclusion, suggestions for further research on the topic will be stated.

Keywords: Natural Language Processing, Sentence-Boundary Detection, Punctuation Marks, Disambiguation, Turkish National Corpus (TNC)

* Bu çalışma TÜBİTAK 113K039 no'lu proje kapsamında yapılmıştır.

1. Giriş

Doğal Dil İşleme (DDİ) günümüzde dilbilim, dil eğitimi, bilgisayar mühendisliği gibi pek çok farklı alanı birleştiren bir araştırma alanıdır. Grishman Doğal Dil İşleme ve Bilgisayarlı Dilbilim çalışmalarının 1950lerde bilgisayarla çeviri yöntemi kullanarak başladığını öne sürmektedir [8]. Bilgisayar teknolojisindeki ilerlemeler 1970-80lerde Doğal Dil İşleme çalışmalarının hız kazanmasını sağlamıştır.

Doğal Dil İşleme çalışmalarına temel sayılabilecek alanlardan biri olan Derlem İşlemleme (Corpus Processing) evrensel anlamda bilgisayarlı dilbilim çalışmalarına temel oluşturan bir alt alan olarak göze çarpmaktadır. Derlem, özel ya da genel amaçlı incelemeler yapmak için yazılı ve sözlü metinlerden oluşan metinler bütünü olarak tanımlanmaktadır. Bir başka deyişle derlem, elektronik veritabanında kayıtlı metinlerin veri bilgisiyle birleştirilmiş toplamıdır [4]. Bilgisayarların kullanımıyla oluşturulan kapsamlı derlem veritabanları dilin farklı yönlerinin araştırmacılar tarafından betimlenmesine olanak tanımaktadır.

Aktaş, tümce sonu belirleme işlemini derlem oluşturma işleminin ilk sırasına koymaktadır [2]. Tümce sonu belirleme çalışmalarının büyük bir çoğunluğu her ne kadar istatistiksel ve makine öğrenimine dayalı olsa da Türkçe için kural tabanlı bir çalışma ilk kez Aktaş ve Çebi tarafından ortaya konmuştur [3].

Tümce sonu belirleme çalışmalarında, nokta, ünlem, soru işareti vb. noktalama işaretleri sadece tümce ayırıcı olarak kullanılmazlar ve bu anlamda, tümce sonu belirleme, noktalama işaretlerinin belirginleştirilmesi olarak da özetlenebilir.

Dilbilimciler açısından, tümce sonu belirleme aşaması, çok sözcüklü birimlerin tümce temelinde çıkarımı, derlem temelli/çıkışlı yapılacak analizlerin tümce boyutunda yapılması, sözcük türü işaretleme (part-of-speech tagging) çalışmalarında ortaya çıkan belirsizliklerin tümce boyutunda ele alınan bağlamla en aza indirgenmesi, sözdizimsel ayrıştırma (syntactic parsing) için oluşturulacak olası tümce öğelerinin diziliminin çıkarımı gibi konularda bir ön-gereksinim olarak karşımıza çıkmaktadır.

Bilgisayar mühendisliği açısından tümce sonu belirleme ele alındığında, sözdizimsel ayrıştırma (syntactic parsing), bilgi çıkarımı (information extraction), makine çevirisi (machine translation), metin hizalama (text alignment), belge özetleme (document summarization), istatistiksel ya da makine öğrenmesi yöntemiyle sözcük türü belirginleştirme çalışmaları için önemli olduğu söylenebilir.

2. Alanyazın

Tümce sonu belirleme çalışmaları daha çok kurallı ifadeler, kısaltma listeleri aracılığıyla tümce sonlarını belirleyen betikler veya pek çok aracı içinde barındıran araç takımları (toolkits) aracılığıyla gerçekleşmektedir.

Bilgisayarlı dilbilim alanyazınında tümce sonu belirleme problemi iki farklı yöntemle çözümlenmeye çalışılmıştır. Bunlardan ilki kural tabanlı yaklaşımdır. Kural tabanlı tümce sonu belirleme yaklaşımının başlı başına anlaşılmasının zor olması ve veri setlerinin yalnızca kullanılan metinlerle sınırlı kalması eksiklikleri olarak sıralanabilir [5]. Tümce sonu belirleme sorununun çözümüne bir diğer yaklaşım ise, Makine Öğrenmesine dayalı yaklaşımdır. Reynar ve Ratnaparkhi [12] tarafından hazırlanan ve maksimum entropi yaklaşımı kullanan çalışma, Riley [13] tarafından izlenen Karar Ağacı Sınıflandırıcısı (Decision Tree Classifier), Palmer ve Hearst [11] tarafından ortaya konan Sinir Ağı (Neural Network) Yaklaşımı ve bu çalışmalara ek olarak hem Hidden Markov modelini hem de Maksimum Entropi yaklaşımını birleştirerek melez bir yaklaşım kullanan Mikheev [10] makine öğrenmesine dayalı yapılan tümce sonu belirleme çalışmalarına örnek olarak verilebilir [5].

Tümce sonu belirlemede kullanılan uygulamalar kendi veri setlerine uygun olarak

tasarlandığından farklı metin alanlarındaki başarım oranları büyük farklılıklar gösterebilmektedir. Bu uygulamaların sınandığı veri setlerini, daha çok özel amaçlı derlemler ya da farklı dil kullanımlarını içermeyen gazete metinleri oluşturmaktadır. Ancak tümce sonu belirlemede başarım oranlarını arttırmak için farklı alanlardan alınarak hazırlanmış veri setlerini kullanmak gerekmektedir.

Apache OpenNLP kütüphanesi (http://opennlp.apache.org/) tek başına tümce sonu belirleme aracı olmaktan çok, pek çok doğal dil işleme aracını barındıran, makine öğrenmesi yöntemiyle doğal dil metinlerini işlemleyebilen bir araç takımı olarak araştırmacılara sunulmuştur. Uygulama içerisinde; sözcükbirim belirleme (tokenization), tümce sonu belirleme, sözcük türü işaretleme, isim verilmiş varlık (named entity) çıkarımı gibi pek çok farklı işlevle kullanılan araçları barındırmaktadır. Open NLP uygulaması maksimum entropi ve perceptron tabanlı makine öğrenmesini de beraberinde sunmaktadır.

Tomanek vd. tarafından hazırlanan Julie Sentence Boundary Detector (JSBD) biyoloji ve tıp alanında yazılmış metinlerin tümce sonu belirlemesi hedefiyle oluşturulmuş açık kaynak kodlu bir uygulamadır [14]. JSBD makine öğrenmesi yöntemiyle eğitici bir model veri seti yardımıyla tümce sonlarını işaretlemeyi hedeflemektedir. JSBD Java programlama dili ile hazırlanmıştır.

GENIA tümce sonu belirleme aracı (GeniaSS) [9] Unix ve benzeri platformlarda çalışabilen, Ruby programlama dili ile hazırlanmış, biyoloji ve tıp metinlerini tümcelere ayırmak için tasarlanmış açık kaynak kodlu bir uygulamadır. Virgül, tek ya da çift tırnak işareti, parantezler vb. noktalama işaretlerine bakarak üye tümceleri otomatik olarak tanımlar.

GeniaSS biyoloji ve tıp metinleri için hazırlanmış bir uygulama olduğundan oluşturulan alt-derlem üstünde belirgin hatalarla tümce sonlarını işaretleme işlemi tamamlanabilmektedir. Soru işareti ve ünlem işaretinin uygulama için bir ayırıcı olmaması, uygulamanın hali hazırda bir kısaltma sözlüğü kullanmaması en belirgin eksikleri olarak sayılabilir. Bununla birlikte Unix tabanlı işletim sistemlerinde terminal aracılığıyla araştırmacıların ekstra yazılım bilgisine ihtiyaç duymadan uygulamayı kullanabilmesi bir artı olarak değerlendirilebilir. Çalışmanın üçüncü bölümünde GeniaSS ile yapılan uygulamaya ilişkin ayrıntılı bir betimleme sunulacaktır.

Gillick [7] tarafından hazırlanan Splitta, Destekçi Vektör Makinesi (Support Vector Machine (SVM)) kullanılarak İngilizce için hazırlanmış açık kaynak kodlu bir diğer tümce sonu belirleme aracıdır. Python programlama dili kullanılarak hazırlanmış bu uygulama temsil yeterliliği olduğu düşünülen Brown Derlemi [6] ve Wall Street Journal gazetesi verilerinin bir birleşimi üstünde yüksek başarım oranına sahip sonuçlar ortaya çıkarmıştır. İngilizce metinlerde uygulamanın ortaya çıkardığı hata oranı yaklaşık % 0,25'tir.

Türkçe için yapılan tümce sonlarını belirleme çalışmaları incelendiğinde, çalışmaların hem istatistiksel hem de makine öğrenmesine dayalı yöntemler kullanılarak yapıldığı görülmüştür.

Tür [15] Türkçe için *İstatistiksel Bir Bilgi Çıkarım Sistemi* adlı çalışmasında, istatistiksel dil işleme modeli kullanarak Türkçe metinlerden bilgi çıkarımı üzerine yaptığı bir dizi çalışmada Cümlelere Ayırma Sistemini de adapte etmiştir. Cümlelere Ayırma Sistemiyle, verilen bir dizi sözcüğü sözdizimsel bağlamda tümcelere bölmeyi amaçlamıştır. Kullanılan veri seti ve destekleyici öğeler yardımıyla yapılan bu çalışmadaki başarım oranı %91,56 olarak belirtilmiştir [15].

Dinçer ve Karaoğlan [5] Türkçede tümce sonu belirleme çalışmalarında Türkçe seslemleme ve Türkçenin fonetik özelliklerini kullanarak noktaların belirsizliğini gidermeye çalışmışlardır. Algoritmalarının başarım oranı %96,02 olarak belirtilmiştir. Çalışmalarının alanyazına katkısını sözlük kullanmadan

tümce sonu belirleme problemine çözüm getiren bir yöntem olarak tanımlamaktadırlar [5].

Aktaş ve Çebi [3] tarafından hazırlanan uygulama, diğer çalışmaların aksine kural tabanlı bir yöntem izleyerek güncel Türkçe metinleri tümcelerine ayırmayı hedeflemektedir. Uygulama kural listeleri, kısaltma listeleri ve girdi metin kullanarak tümce sonu işaretlemesini XML dosyası biçiminde sunmaktadır. Güncel Türkçe gazete metinlerinin köşe yazılarını temel alarak sınanan uygulamanın başarım oranı % 99,60 ile %99,80 arasında bulunmuştur [3].

3. Açık Kaynak / Ücretsiz Yazılımlarla Türkçe Tümcelerin Belirlenmesi

Çalışmanın bu bölümünde açık kaynak kodlu ve/veya ücretsiz yazılımlarla TUD veritabanından çekilerek hazırlanmış, dengeli, dili temsil yeterliliğine sahip 10 milyon sözcükten oluşan alt-derlemin kapsamı, çalışma süresince kullanılan açık kaynak kodlu uygulamalar ve bunların performans analizleri betimlenecektir.

Doğal Dil İşleme çalışmaları kapsamında hem evrensel anlamda hem de Türkçe metinlerde kullanılan veri setleri ne yazık ki dili temsil yeterliliğine sahip değildir. Tümce sonu belirleme çalışmalarında kullanılan veri setleri ya belli bir konu alanına odaklanmış ya da karmaşık yapılı tümceler üstünde denenmemiştir.

3.1. TUD-Alt Derlemi

Çalışma süresince tümce sonu belirleme uygulamaların performans analizlerinin yapıldığı veritabanının içeriği ile ilgili ayrıntılı bir döküm, çalışmanın bu bölümünde sunulmaktadır. Çalışmaya konu olan derlem, TUD dağılım ölçütleri kullanılarak hazırlanmış [1], günümüz Türkçesinin metin örneklerinden oluşan, 20 yıllık bir dönemi (1990-2009) kapsayan, çok farklı alan ve türden yazılı ve sözlü metin örneklerini içeren, dengeli ve temsil yeterliliğine sahip bir alt-derlemdir. Çalışmanın veri setinin dağılımı Tablo 1, Tablo 2, Tablo 3 ve Tablo 4'te gösterilmiştir.

Alan	Oran	Toplam Sözcük Sayısı	Hedeflenen Sözcük Sayısı
1.Kurgusal Düzyazı	%19	1.901.174	1.900.000
2. Bilgilendirici Metinler	%81	7.956.406	8.100.000

Tablo 1. Alana göre Dağılım

Türev Metin Biçimi	Oran	Toplam Sözcük Sayısı
1. Akademik Düzyazı	%95	1.806.708
2. Kurgu ve Şiir	%2	37.059
3. Dram, Tiyatro	%3	57.407

Tablo 2. Kurgusal Düzyazı Metinlerinin Türev Metin Biçimine göre Dağılımı

Media	Oran	Toplam Sözcük
1. Kitaplar	%46,1	3.667.944
2. Süreli Yayınlar	%37,1	2.951.859
2.1. Bilim.Dergileri	%14,9	1.185.466
2.2. Gazeteler	%11,1	883.176
2.3. Dergiler	%11,1	883.217
3. Diğer Basılmış Metinler	%6,09	484.550
4. Basılmamış Yazılı Metinler	%2,5	198.912
5. Sözlü Metinler	%8,21	653.228

Tablo 3. Bilgilendirici Metinlerin Medyaya göre Dağılımı

Alan	Oran	Toplam Sözcük Sayısı
1. Bilgilendirici: Doğa ve Temel Bilimler	%5,03	400.207
2. Bilgilendirici: Uygulamalı Bilimler	%10,21	812.349
3. Bilgilendirici: Sosyal Bilimler	%20,08	1.597.646
4. Bilgilendirici: Dünya Sorunları	%22,57	1.795.761
5. Bilgilendirici: Sanat	%8,78	698.572
6. Bilgilendirici: Düşünce ve İnanç	%5,00	397.820
7. Bilgilendirici: Serbest	%18,29	1.455.226
8. Bilgilendirici: Ticaret ve Finans	%10,04	798.823

Tablo 4. Bilgilendirici Metinlerin Alanlara göre Dağılımı

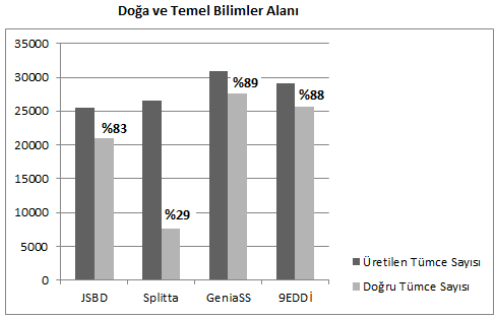
3.2. Açık Kaynak Kodlu / Ücretsiz Uygulamaların Performans Analizleri

Bu çalışmada açık kaynak kodlu Julie Sentence Boundary Detector (JSBD) [14], GeniaSS [9], Splitta [7], yazılımları ile ücretsiz Web servisi şeklinde çalışan ve Dokuz Eylül Üniversitesi Doğal Dil İşleme Araştırma Grubu (9EDDİ) [3] tarafından Türkçe metinler için geliştirilmiş tümce ayırma sistemi karşılaştırılmıştır. İlk olarak herhangi bir alan sınırlaması olmadan 10 milyon sözcükten oluşan alt-derlem üzerinde tümce ayırma yazılımları denenmiş, yazılımların elde ettikleri tümceler ile bu alt-derlem üzerinde daha önce yarı-otomatik olarak oluşturulmuş ve el ile kontrol edilmiş doğru tümceler karşılaştırılmıştır. Kullanılan alt-derlemde yarı-otomatik yöntemle toplam 774.449 tümce elde edilmiş olup, denemesi yapılan yazılımlar ile elde edilen toplam tümce sayıları ve doğru tümce sayıları Tablo 5'te verilmiştir. Yazılımlar tarafından üretilen doğru tümce sayısının, alt-derlem üzerinde yarı-otomatik olarak oluşturulmuş doğru tümce sayısına oranı hesaplandığında, JSBD %70, Splitta %22, GeniaSS %88 ve 9EDDİ %75 oranında başarılı olmuştur.

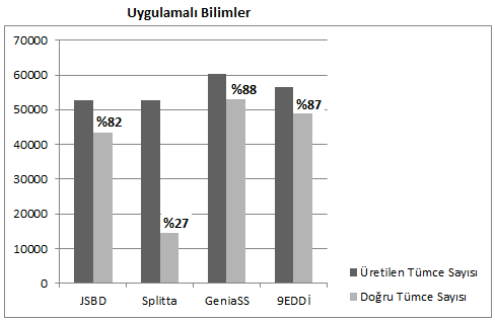
Yazılım	Bulunan Toplam Tümce Sayısı	Doğru Tümce Sayısı	Doğruluk Oranı
JSBD	690.998	539.628	%70
Splitta	664.769	171.467	%22
GeniaSS	893.401	681.850	%88
9EDDİ	683.609	576.920	%75

Tablo 5. Tümce Sonu Belirleme Yazılımlarının Alt-derlem Üzerindeki Başarımı

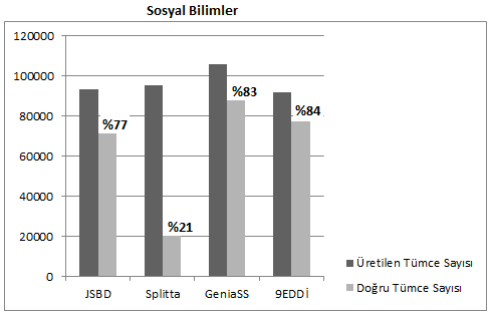
Tablo 5'de görüldüğü gibi alt-derlem bir bütün olarak ele alındığında en başarılı yazılımın GeniaSS olduğu görülmektedir. Splitta yazılımının başarı oranının düşük olmasının nedeni, alt-derlemde başlık satırları gibi tümce sonunu belirleyen bir noktalama işareti ile bitirilmemiş olan satırların bulunmasıdır. Denemesi yapılan yazılımların, kullanılan alt-derlemdeki başarı oranının düşük olması, alt-derlemde 8 farklı alana ait metinlerin bulunması ve alanlara göre tümce yapılarının farklılık gösterebilmesinden kaynaklanmaktadır. Alanyazın bölümünde de belirtildiği gibi, JSBD ve GeniaSS, genellikle tıp ve biyoloji metinleri temel alınarak geliştirilmiş; Splitta Brown Derlemi ve Wall Street Journal gazetesi verileri üzerinde yüksek başarı göstermiş; 9EDDİ ise gazete köşe yazıları üzerinde yüksek başarıma sahip olmuştur. Bu deneyde kullanılan alt-derlemde ise Tablo 4'te belirtildiği gibi çok farklı alanlarda metinler olduğundan yazılımların başarım oranı oldukça düşmüştür. Bu nedenle, deneyde kullanılan alt-derlem alanlara göre de bölünmüş ve her alan için her bir yazılımın başarım oranları tekrar hesaplanmıştır. Şekil 1-8 yazılımların her alan için ürettikleri toplam tümce sayılarını, bu tümcelerden kaç tanesinin doğru olduğunu ve üretilen toplam tümcelerin yüzde kaçının doğru olduğunu göstermektedir.



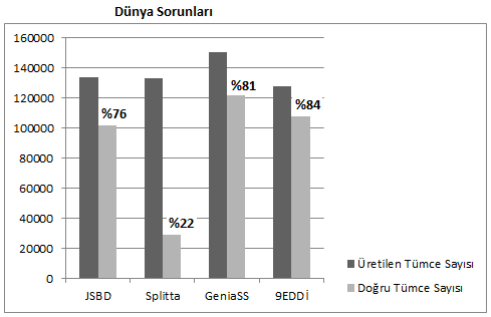
Şekil 1. Yazılımların Doğa ve Temel Bilimler Alanındaki Metinler Üzerindeki Başarımı



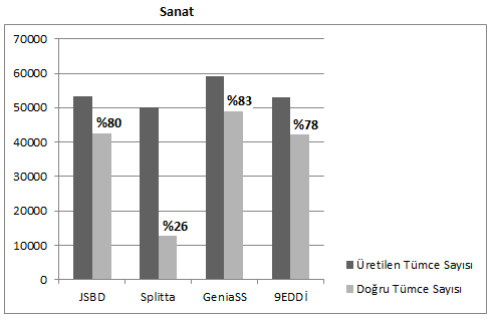
Şekil 2. Yazılımların Uygulamalı Bilimler Alanındaki Metinler Üzerindeki Başarımı



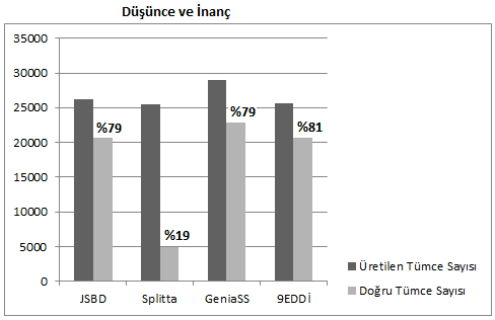
Şekil 3. Yazılımların Sosyal Bilimler Alanındaki Metinler Üzerindeki Başarımı



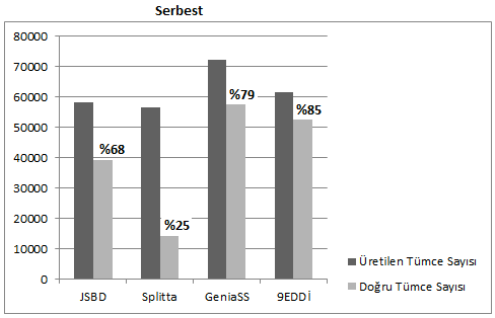
Şekil 4. Yazılımların Dünya Sorunları Alanındaki Metinler Üzerindeki Başarımı



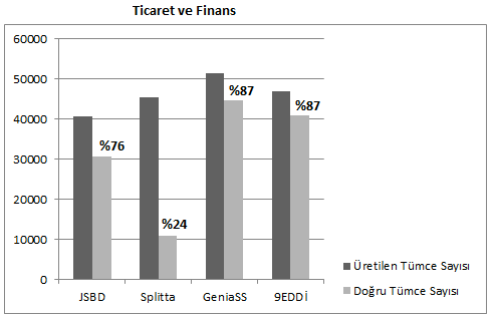
Şekil 5. Yazılımların Sanat Alanındaki Metinler Üzerindeki Başarımı



Şekil 6. Yazılımların Düşünce ve İnanç Alanındaki Metinler Üzerindeki Başarımı



Şekil 7. Yazılımların Serbest Alanındaki Metinler Üzerindeki Başarımı



Şekil 8. Yazılımların Ticaret ve Finans Alanındaki Metinler Üzerindeki Başarımı

Şekil 1-8'de yer alan yüzdelik ifadeler, yazılımın o alan için ürettiği doğru tümce sayısının, yazılımın o alan için ürettiği toplam tümce sayısına bölünmesi ile elde edilmiştir. Tablo 5 ile Şekil 1-8 karşılaştırıldığında, yazılımların ürettiği doğru tümce oranlarının, yazılımlarım başarım oranlarından daha yüksek olduğu görülmektedir. Bunun nedeni ise, yazılımların bazı tümce grupları için hiçbir sonuç döndürmemiş olmasıdır. Bir diğer deyimle yazılımlar bazı tümce gruplarını görmezden gelmiş, ancak ele alıp işlediği metinleri de (Splitta hariç) genellikle %75 ile %89 arasında değişen oranlarda doğru olarak tümcelere ayırabilmiştir. Tablo 5 ve Şekil 1-8'de görüldüğü gibi en fazla

sayıda tümce oluşturan yazılım GeniaSS olup, en fazla sayıda doğru tümceyi üreten yazılım da yine GeniaSS olmuştur. 9EDDİ ise, GeniaSS'a göre daha az sayıda tümce üretmiş, ancak Sosyal Bilimler, Dünya Sorunları, Düşünce ve İnanç, Serbest alanlarında GeniaSS'a göre daha yüksek doğrulukla tümce ayırma yapabilmiştir. Bunun en büyük nedeni 9EDDİ'nin güncel Türkçe metinler için, GeniaSS'ın ise İngilizce tıp ve biyoloji metinleri için geliştirilmiş yazılımlar olmalarıdır.

Şekil 1-8'de görüldüğü gibi, Sanat, Serbest, Düşünce ve İnanç alanlarında yazılımların genel olarak daha düşük oranda doğru tümce üretebildikleri gözlenmektedir. Bu alanlar, çok farklı konular üzerine yazılan metinleri içerdiği için (şiirler, konuşma metinleri, hobiler, biyografiler, bahçecilik vb.) tümce yapıları çok farklılık göstermektedir. Örneğin derlemde bulunan

Prof. Dr. Onur Erol; "Estetik olmak için beklemeye tahammülleri yok. Elinde Angelina Jolie'nin resmiyle geliyor. Henüz 14 yaşında ve doktora resmi uzatıp "beni de böyle yap" diyor."

tümcesi yazılımların hiçbiri tarafından doğru şekilde belirlenememiştir. Bu tümce JSBD, Splitta ve GeniaSS tarafından

Tümce 1: *Prof. Dr. Onur Erol; "Estetik olmak için beklemeye tahammülleri yok.*

Tümce 2: *Elinde Angelina Jolie'nin resmiyle geliyor.*

Tümce 3: *Henüz 14 yaşında ve doktora resmi uzatıp "beni de böyle yap" diyor."*

şeklinde 3 tümceye bölünürken; tümcenin sonu nokta yerine tırnak işareti ile bittiği için 9EDDİ tarafından boş tümce olarak döndürülmüş ve belirlenememiştir. Derlemde yer alan bu gibi tümceler nedeniyle, hem 9EDDİ, hem de diğer yazılımların doğruluk oranları oldukça düşük çıkmıştır.

Yazılımların ürettikleri tümceler incelendiğinde, 9EDDİ'nin, sonunda nokta olmayan satırları (şiir gibi) hiç işleme almadan boş olarak döndürdüğü, çift tırnak arasında yer alan ve sonu nokta gibi tümce sonunu belirleyici bir işaretle biten tümceler (konuşma alıntısı gibi) için doğru ayrımı kısmen yapıp, tırnaklar arasındaki metinde yer alan noktalama işaretlerini ise ya tamamen attığı ya da ilave noktalama işaretleri koyduğu ve bu nedenle başarı oranın düştüğü gözlenmiştir. JSBD, Splitta ve GeniaSS yukarıdaki örnek tümcede de görüldüğü gibi genellikle benzer hataları yapmaktadır. Bu üç yazılım, özellikle tırnak işareti arası ifadelerde ve isim kısaltmalarında yanlış tümce belirlemesi yapmaktadır. Bunun yanı sıra Splitta, satır sonunu belirleyici bir noktalama işareti olmadığı durumda bu satırı ardından gelen satır ile birleştirmekte, böylece her iki tümcenin de yanlış şekilde belirlenmesine neden olmaktadır. JSBD ise farklı olarak madde imleri olan satırlarda hatalı çalışmaktadır.

4. Sonuç ve Öneriler

Bu çalışmada açık kaynak kodlu tümce sonu belirleme sistemlerinden JSBD, GeniaSS, Splitta, ile ücretsiz Web servisi şeklinde çalışan Dokuz Eylül Üniversitesi Doğal Dil İşleme Araştırma Grubu (9EDDİ)'nun tümce sonu belirleme yazılımı TUD veritabanından çekilerek hazırlanmış, dengeli, dili temsil yeterliliğine sahip 10 milyon sözcükten oluşan alt-derlem üzerinde denenmiş ve elde edilen sonuçlar karşılaştırılmıştır. Tüm yazılımlar, geliştirildikleri metin grupları için yüksek başarım oranlarına sahip oldukları halde, bu çalışmada kullanılan ve sekiz farklı alandan metinlerin bulunduğu derlemde başarı oranları daha düşük olmuştur. Bunun en büyük nedeni olarak, bu çalışmada kullanılan derlemin çok farklı tümce yapılarına sahip metinlerden oluşması gösterilebilir. GeniaSS yazılımı İngilizce için geliştirilmiş olmasına rağmen, bu çalışmada kullanılan Türkçe derlem için de en yüksek başarım oranını veren yazılım olmuş, daha sonra güncel Türkçe metinler için geliştirilmiş 9EDDİ yazılımı başarılı sonuçlar vermiştir. 9EDDİ, metni daha az sayıda tümceye böldüğü için ürettiği doğru tümce sayısı da daha az olmuş, ancak üretilen doğru tümce oranı açısından GeniaSS'a göre daha başarılı olmuştur.

Bu çalışma çeşitli alanlarda yazılmış Türkçe metinler için daha etkin tümce sonu belirleme sistemlerine ihtiyaç olduğunu göstermiştir. Makine öğrenmesi tabanlı ya da kural tabanlı bir tümce sonu belirleme yöntemi geliştirilirken TUD alt-derlemi gibi dili temsil etme yeteneğine sahip bir derlem ile çalışmanın daha etkin sistemlerin geliştirilmesine yardımcı olacağı düşünülmektedir.

5. Kaynaklar

[1] Aksan, Y. et al., "Construction of the Turkish National Corpus (TNC)", **Proceeding of the Eight International Conference on Language Resources and Evaluation (LREC 2012),** İstanbul, (2012).

[2] Aktaş, Ö., "Türkçe için Verimli bir Cümle Sonu Belirleme Yöntemi", **Proceeding of the Akademik Bilişim 2006,** Pamukkale, Türkiye, (2006).

[3] Aktaş, Ö., Çebi, Y., "Rule-Based Sentence Detection Method (RBSDM) for Turkish", **International Journal of Language and Linguistics,** 1 (1), 1-6, (2013).

[4] Baker, P., Hardie, A., and McEnery, T., "A glossary of corpus linguistics", Edinburgh: Edinburgh University Press. (2006).

[5] Dinçer, T., Karaoğlan, B., "Sentence Boundary Detection in Turkish", **Proceeding of the Advances in Information Systems: Third International Conference**, İzmir-Turkey, (2004).

[6] Francis, W. N. and Kućera, H., *Brown corpus manual. Unpublished manuscript,* Brown University, Rhode Island, US, (1964).

[7] Gillick, D., "Sentence boundary detection and the problem with the U.S.", **Proceeding of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics,** 241-244, (2009).

[8] Grishman, R. "Computational linguistics: an introduction." Cambridge Cambridgeshire/New York: Cambridge University Press, (1986).

[9] Kim, J. D., Ohta T., Tateishi Y., and Tsujii J., GENIA corpus - a semantically annotated corpus for bio-textmining. Bioinformatics, 19, 180-182, (2003).

[10] Mikheev, A. *Tagging Sentence Boundaries*. Language Technology Group, University of Edinburgh, (1997).

[11] Palmer, D. D. and Hearst, M. A., *Adaptive multilingual sentence boundary disambiguation. Computational Linguistics*, (1997).

[12] Reynar, J. C. and A. Ratnaparkhi. "A maximum entropy approach to identifying sentence
Boundaries", **Proceeding of the Fifth A CL Conference on Applied Natural Language Processing (ANLP'97),** Washington, D.C., (1997).

[13] Riley, M.D. "Some applications of tree-based modeling to speech and language indexing", **Proceeding of the DARPA Speech and Natural Language Workshop**, 339-352, (1989).

[14] Tomanek, K., Wermter, J. and Hahn, U. "Sentence and token splitting based on conditional random fields", **Proceeding of the 10th Conference of the Pacific Association for Computational Linguistics**, Melbourne, Australia, 49-57, (2007).

[15] Tür, G., A Statistical Information Extraction System. PhD Thesis, Bilkent University, Ankara, Turkey, 2000.