Türkçe Doğal Dil İşleme için Arayüzler
User Interfaces for Turkish Natural Language Processing

Gökçe Uludoğan[1], Rıza Özçelik[1], Selen Parlar[1, 2], Gökhan Ercan[3], Olcay Taner Yıldız[3]
[1]Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye
[2]Starlang Yazılım Danışmanlık, İstanbul, Türkiye
[3]Işık Üniversitesi Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye
{gokce.uludogan, riza.ozcelik, selen.parlar}@boun.edu.tr, gokhan.ercan@isik.edu.tr, olcaytaner@isikun.edu.tr

***Özetçe*—Türkçe doğal dil işleme alanında var olan çalışmalardan en bilinen iki tanesi Zemberek ve İTÜ Doğal Dil İşleme Yazılım Zinciri'dir. Bu çalışmalardan Zemberek açık kaynak kodlu olup çeşitli doğal dil işleme bileşenlerinden oluşan bir yazılım kütüphanesidir. İTÜ Doğal Dil İşleme Yazılım Zinciri ise, çevrimiçi bir kullanıcı arayüzü sunmasına rağmen, açık kaynak kodlu değildir. Bu çalışmada, yine bir Türkçe doğal dil işleme aracı olan Türkçe Nlptoolkit için bir çevrimiçi arayüz sunuyoruz. Bu arayüz sayesinde, Nlptoolkit'teki yazım denetleyici, Türkçe karakter dönüştürücü, birimlendirici, cümle bölücü, biçimbirimsel çözümleyici ve belirsizlik giderici bileşenlerini çevrimiçi olarak kullanmak mümkündür.**

***Anahtar Kelimeler*—Biçimbilimsel Çözümleme, Biçimbilimsel Belirsizlik Giderme, Yazım Denetimi**

***Abstract*—Among the most popular works in Turkish natural language processing there are Zemberek and ITU Turkish Natural Language Processing Pipeline. The former is an open source software library that consists of various natural language processing components whereas the latter provides an online user interface, yet not open source. In this study, we present an online user interface for Turkish Nlptoolkit, which is another Turkish natural language processing tool. Thanks to this interface, it is possible to use asciifier, deasciifier, tokenizer, sentence splitter, morphological analyzer/disambiguator components of Nlptoolkit online.**

***Keywords*—Morphological Analysis, Morphological Disambiguation, Spell Checker**

I. GİRİŞ

Doğal Dil İşleme (DDİ), genel tanımıyla, hesaplamalı yöntemler kullanarak doğal dilde oluşturulan bir metin veya konuşmadan anlam çıkarmayı ya da konuşma veya metin sentezlemeyi hedefleyen bir yapay zeka koludur. Metin ve konuşmaların doğal dil ile üretiliyor olması, DDİ çalışmalarının sadece çalışılan dil üzerinde geçerli birçok özelliğe sahip olmasına neden olabilmektedir. Bu nedenle, bir konuşma dilinde DDİ alanında yapılan bir çalışma, başka bir dile doğrudan aktarılamayabilir. Bu da, dil özelinde yapılan çalışmaları daha önemli kılmaktadır.

DDİ alanında, birçok açık kaynak kodlu çalışma mevcuttur. Bu çalışmalar, spaCy [1], NLTK [2] veya Stanford CoreNLP [3] gibi programlama arayüzü sunan biçimlerde olabileceği gibi, Princeton WordNet [4] örneğinde olduğu gibi kullanıcı arayüzü de sunabilmektedir. Ancak, DDİ alanındaki pekçok çalışma İngilizce alanında yapıldığı için, bu çalışmaların sonuçlarının veya çalışmalarda ortaya çıkan araçların Türkçe'ye doğrudan aktarılması mümkün olmamaktadır. Bu da Türkçe DDİ çalışmaları için bir teşvik oluşturmuş; Zemberek ve İTÜ Türkçe DDİ Yazılım Zinciri gibi çalışmalara zemin hazırlamıştır. Bu çalışmalardan Zemberek, açık kaynak kodlu, Java dilinde yazılmış DDİ bileşenlerinden oluşan bir yazılım kütüphanesidir ve bir kullanıcı arayüzü sunmamaktadır. Zemberek'in aksine, İTÜ Türkçe DDİ Yazılım Zinciri, var olan araçlara çevrimiçi erişime olanak sağlayan, ancak kapalı kodlu bir sistemdir.

Biz de bu çalışmada, Türkçe DDİ alanında, açık kaynak kodlu bir araç derlemesi olan Türkçe Nlptoolkit için geliştirdiğimiz arayüzü sunuyoruz [5]. Böylelikle, şu ana kadar Türkçe için geliştirilmiş olan DDİ araçlarının aksine, hem açık kaynak kodlu, hem de çevrimiçi arayüze sahip olan bir Türkçe DDİ paketini kullanıma açmış oluyoruz.

Ortaya koyduğumuz bu arayüz, Türkçe Nlptoolkit'in birçok bileşenine teknik nedenlerden dolayı kısıtlı da olsa, çevrimiçi erişim sağlamaktadır. Bu bileşenleri, Türkçe karakter dönüştürücü (asciifier/deasciifier) [6], birimlendirici/cümle bölücü (tokenizer/sentence splitter) [7], yazım denetleyici (spell checker) [8] ve biçimbilimsel çözümleyici/belirsizlik giderici (morphological analyzer/disambiguator) [9], [10] olarak sıralayabiliriz. Bu bileşenlerden, Türkçe karakter dönüştürücü, birimlendirici/cümle bölücü ve yazım denetleyici 10000 karakter uzunluğundaki girdilere kadar desteklese de biçimbilimsel çözümleyici/belirsizlik giderici 1000 karaktere kadar uzunluktaki girdilere kadar cevap vermektedir.

Bu makale şöyle düzenlenmiştir: II. kısımda, şu ana kadar geliştirilen DDİ araçları hakkında bilgi verilmiştir. III. kısımda arayüz tarafından desteklenen bileşenler hakkında detaylar verilirken; IV. kısımda ise, sunduğumuz arayüz, çeşitli yönleriyle ele alınmış ve kullanım örnekleri sunulmuştur. Son tahlilde, V. kısımda, çalışmamız özetlenmiş ve yapılabilecek iyileştirmeler tartışılmıştır.

978-1-5090-6494-6/17/$31.00 ~©~2017 IEEE

II. LİTERATÜR ÖZETİ

Doğal dil işleme alanında çok sayıda araç ve arayüz geliştirilmiştir. Stanford CoreNLP [3], NLTK [2], Apache OpenNLP [11], spaCy [1] birçok dili destekleyen açık kaynaklı ve popüler araçlardandır. Türkçeye özgü olan araçlar ise Zemberek [12] ve ITU Türkçe Doğal Dil İşleme Yazılım Zinciridir (ITU Turkish NLP Pipeline) [13].

Zemberek açık kaynaklı bir kütüphane olup şu temel işlevleri sağlamaktadır: yazım denetleyici (spell checker), biçimbilimsel çözümleyici (morphological analyzer), Türkçe karakter dönüştürücü (asciifier/deasciifier), birimlendirici (tokenizer). Zemberek'in bir kullanıcı arayüzü bulunmamaktadır. Dolayısıyla hitap ettiği kullanıcı kitlesi yazılım geliştiricilerden oluşmaktadır.

ITU Türkçe Doğal Dil İşleme Yazılım Zinciri, Türkçe karakter dönüştürücü (asciifier/deasciifier), birimlendirici/cümle bölücü (tokenizer/sentence splitter), yazım denetleyici (spell checker), biçimbilimsel çözümleyici/belirsizlik giderici (morphological analyzer/disambiguator), varlık ismi tanıma (named entity recognizer) ve bağımlılık çözümlemesi (dependency parser) gibi araçlardan oluşan bir platformdur. Bu platform, hem bir web arayüzüne hem de bir uygulama programlama arayüzüne (API) sahiptir. Böylece farklı seviyelerdeki kullanıcılar bu platformdan faydalanabilir. Fakat bu platform açık kaynaklı değildir, dolayısıyla bu araçları değiştirme ve geliştirme imkanı tanımamaktadır.

Sunduğumuz Nlptoolkit, hem açık kaynaklı olup hem de çevrimiçi bir kullanıcı arayüzüne sahiptir, böylece bir yandan geliştiricilere bu araçları geliştirme ya da kendi projelerinde kullanma imkanı verirken öte yandan arayüzü ile yazılım alt yapısı olmayan kullancıların bu araçlardan faydalanmasına imkan tanımaktadır.

III. ARAÇLAR

Nlptoolkit'te var olan araçlar şunlardır: Türkçe karakter dönüştürücü (asciifier/deasciifier), birimlendirici/cümle bölücü (tokenizer/sentence splitter), yazım denetleyici (spell checker) ve biçimbilimsel çözümleyici/belirsizlik giderici (morphological analyzer/disambiguator). Bu bölümde bu araçların işleyişinden bahsedilecektir.

A. Türkçe Karakter Dönüştürücü

Nlptoolkit'in bu bileşeni, Türkçe karakterler içeren metinleri Türkçe karakterlerden arındırmak veya Türkçe karakterlerden arındırılmış bir metnin Türkçe karakterlerini geri kazandırmak için kullanılabilir. Bu araç, bir metni Türkçe karakterlerden arındırmak için oldukça basit bir yol takip etmektedir ve her bir Türkçe karakteri karşılık gelen Latince haline dönüştürmektedir. Örneğin, Ç harfleri C'ye çevrilirken, ı harfi i harfine çevrilmektedir.

Bir metne Türkçe karakterleri geri kazandırmak için ise araç iki farklı yöntem içermektedir. Bu yöntemlerin ilki basit geri dönüştürücüdür (simple deasciifier). Bu yöntem, bir kelimenin var olabilecek bütün Türkçe karakter karşılıklarını oluşturmakta ve bu seçeneklerden biçimbirimsel olarak çözümlenebilen bir tanesini rassal olarak seçip sonuç olarak sunmaktadır. Örneğin, *cocuk* girdisi için, basit geri dönüştürücü *cocuk*, *çocuk*, *coçuk* ve *çoçuk* seçeneklerini yaratacak ve sadece *çocuk* biçimbirimsel olarak çözümlenebileceği için *çocuk* kelimesini çıktı olarak verecektir. Diğer yöntem ise n-karakter geri dönüştürücüdür (ngram deasciifier). Bu yöntemde ise, yine her bir kelime için ilk yöntemdeki gibi bir aday listesi oluşturulur ve ardından her bir kelimenin Türkçe'deki n-karakter olasılıkları hesaplanarak en muhtemel aday çıktı olarak verilir.

B. Birimlendirici/Cümle Bölücü

Nlptoolkit'in birimlendirici/cümle bölücü bileşeni, bir serbest metnin birimlerini ve/veya cümlelerini saptamak için kullanılabilir. Bu bileşen, kural tabanlı bir bileşen olup girdiyi önceden belirlenmiş bir kural kümesini takip ederek cümlelere ve birimlerine ayırır. Bu kural kümesi, bir sonraki karakterin küçük/büyük harf olması gibi cümle düzeyinde kurallar içerdiği gibi, bir girdinin Türkçe'deki yaygın kısaltmalar arasında olup olmadığını kontrol etmek gibi dil düzeyinde kurallar da içerir[1]. Özetle, birimlendirici/cümle bölücü bileşeni bir girdi olarak serbest metin alır ve çıktı olarak birimlerine ayrılmış bir cümle kümesi verir.

C. Yazım Denetleyici

Yazım denetleyici, verilen metindeki yazım hatalarını bulup düzelten Nlptoolkit bileşenidir. Her kelime için hatayı tespit edip olası doğru adaylar arasından seçim yapar. Bu bileşen iki farklı yazım denetleyici içermektedir. Bunlar basit yazım denetleyici (simple spell checker) ve n-karakter yazım denetleyicidir (ngram spell checker).

Basit yazım denetleyici, basit geri dönüştürücü ile benzer bir yöntem kullanır. Girdideki her kelime için her karakter gezilip bu karakter olası bütün karakterlerle değiştirilerek mümkün olabilecek bütün kelimeler oluşturulur ve bunlardan biçimbilimsel olarak çözümlemenebilenlerden bir tanesi rassal olarak seçilir.

N-karakter yazım denetleyici, benzer şekilde n-karakter geri dönüştürücü ile aynı mantığı kullanmaktadır. Önce, basit yazım denetleyicide olduğu gibi kelimeler için aday listeleri hazırlanır. Daha sonra ise n-karakter modelinden bu adaylar için olasılıklar hesaplanarak, her kelime için olasılığı en yüksek olan aday çıktı olarak verilir.

D. Biçimbilimsel Çözümleyici/Belirsizlik Giderici

Biçimbilimsel çözümleyici/belirsizlik giderici bileşeni, önce verilen girdinin çözümlemesini yapmakta, sonrasında ise bu çözümlemedeki belirsizlikleri gidermektedir. Örneğin, *Yarın doktora gidecekler.* cümlesinde biçimbilimsel çözümleyici her kelime için farklı çözümlemeler bulur. *Yarın* için olası iki çözümlemeye karşılık gelen anlamlar, *ertesi gün* ve *ikinci ya da üçüncü tekil kişinin yarısı* şeklinde olabilir. Biçimbilimsel belirsizlik giderici, bu belirsizliği ortadan kaldırarak *Yarın* için *ertesi gün* anlamına gelen çözümlemeyi çıktı olarak verir.

[1]Bütün kuralları görmek için [7] ziyaret edilebilir.

Biçimbilimsel çözümleyici, Türkçe dil kurallarından oluşan bir sonlu durum makinesi kullanarak verilen metini çözümler. Girdi için olabilecek belirsizlikler düşünülmeden olası bütün çözümlemeler verilir.

Biçimbilimsel belirsizlik giderici, girdi olarak çözümleme listesi alır ve belirsizliği gidererek sadece doğru çözümlemeleri verir. Bu bileşen, n-karakter modeller kullanmaktadır. Kelimeler ve çözümlemeleri için iki ayrı modelden faydalanmaktadır. Öncelikle, her çözümleme için kelime ve çözümlemesinin n-karakter olasılıkları hesaplanarak en iyi kök kelime seçilir. Sonrasında, çözümlemeler bu kökü içerenlere indirgenir. Son olarak da kalan bu çözümlemelerden n-karakter modeline göre en olası olanlar bulunur ve üstlerde yer alacak şekilde çıktı olarak verilir.

IV. ARAYÜZ

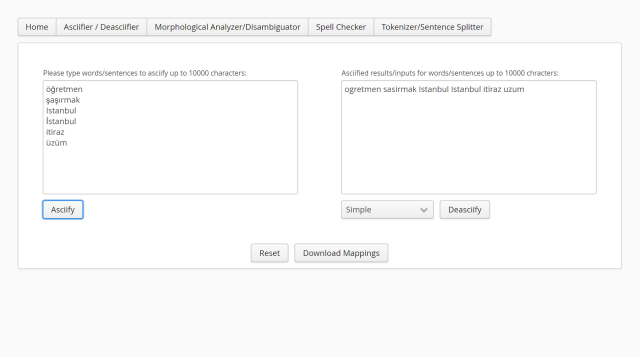
Nlptoolkit'in arayüzü temel olarak dört parçadan oluşmaktadır. Bu arayüzleri Türkçe karakter dönüştürücü (asciifier/deasciifier), birimlendirici/cümle bölücü (tokenizer/sentence splitter), yazım denetleyici (spell checker) ve biçimbilimsel çözümleyici/belirsizlik giderici (morphological analyzer/disambiguator) olarak sıralayabiliriz. Her bir arayüzün işlevi ve kullanımı aşağıda bölümler halinde anlatılmaktadır.

A. Türkçe Karakter Dönüştürücü

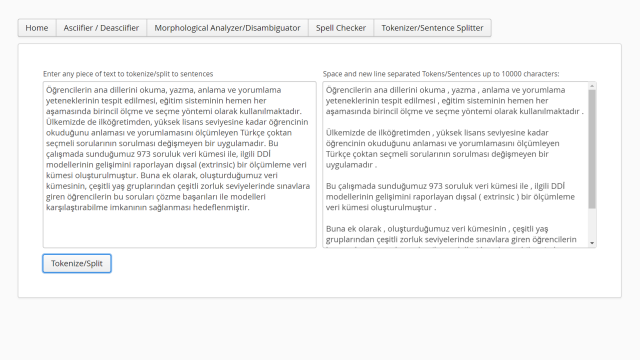
Nlptoolkit arayüzünün bu bileşeni, Türkçe karakterler içeren metinlerdeki Türkçe karakterlerden arındırmak veya Türkçe karakterlerden arındırılmış bir metnin Türkçe karakterlerini geri kazandırmak için kullanılabilir. Bir metni Türkçe karakterlerden arındırmak için, sol taraftaki girdi alanına metin girilip *Asciify* butonuna basılabilir. Bu metni girdi olarak alan Nlptoolkit, metni önce birimlerine ayıracak ve daha sonrasında Türkçe karakterlerden arındırılmış halini sağdaki alanda gösterecektir. Tersi işlemi yapmak için ise, metin sağdaki alana girilip alttaki seçiciden *Simple* veya *NGram* seçildikten sonra *Deasciify* butonuna basılabilir. İlk işlemde olduğu gibi, Nlptoolkit girdiyi öncelikle birimlerine ayıracak ve ardından metnin Türkçe karakterlerini yerine koyacaktır. Her iki işlemin sonunda da, elde edilen kelime dönüşümleri *Download Mappings* butonuna basarak indirilebilir. Bu arayüz Şekil 1'de görülebilir. Dikkat edilmelidir ki, bu dönüşüm için girilebilecek girdi uzunluğu 10000 karakter ile sınırlıdır. Daha uzun bir girdi girilmesi halinde, 10000. karakterden sonraki kısımlar yoksayılacaktır.

B. Birimlendirici/Cümle Bölücü

Arayüzün birimlendirici/cümle bölücü bileşeni, bir serbest metnin birimlerini ve/veya cümlelerini saptamak için kullanılabilir. Türkçe karakter dönüştürücüye benzer olarak, birimlerine ve/veya cümlelerine ayırılacak metin soldaki alandan girdi olarak verilebilir. *Split/Tokenize* butonuna basıldığında, Nlptoolkit metni birimlerine ve cümlerine ayıracak ve sağdaki ekranda gösterecektir. Elde edilen sonuçlar, birimlerin boşlukla, cümlelerin ise ayrı ayrı satırlarda gösterildiği bir biçimde sağdaki ekrandan kopyalanabilir. Önceki bileşene



Şekil 1: Türkçe Karakter Dönüştürücü



Şekil 2: Birimlendirici/Cümle Bölücü

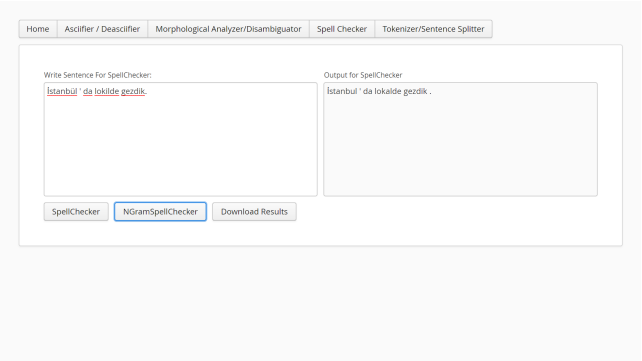
benzer şekilde, bu işlemler girdinin 10000. karakterine kadar uygulanacak, girdinin geri kalanı ihmal edilecektir. Bahsedilen arayüz Şekil 2'de incelenebilir.

C. Yazım Denetleyici

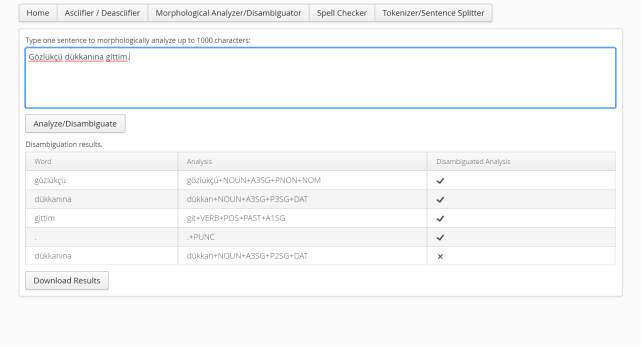
Yazım denetleyici arayüzü bir metindeki yanlış yazılmış kelimelerin ve karakterlerin tespiti ve karşılık gelen doğru yazımları bulmak için faydalanılabilecek bir arayüzdür. Nlptoolkit'in bu bileşeni, *NGram* ve *Simple* olmak üzere iki farklı şekilde tespit yapabilmektedir ve istenen seçenek arayüzden seçilebilir. Daha önceden anlatılan bileşenlere benzer olarak, seçim yapıldıktan sonra, girdi önce birim/cümle ayırıcı kullanılarak birimlerine ve cümlelerine ayırılacak ve yazım denetleyici çalıştırıldıktan sonra sonuçlar sağdaki alanda gösterilecetir. Sonuçlar, *Download Results* butonuna basılarak indirileblir. İndirilen sonuçlarda, her yanlış yazılan birim Levenshtein uzaklık yardımıyla doğru biçimiyle hizalanmış şekildedir. Tıpkı önceki bileşenlerde olduğu gibi, bu arayüzde de karakter sayısı 10000 ile sınırlandırılmıştır. Bu bileşen de Şekil 3'te görülebilir.

D. Biçimbilimsel Çözümleyici/Belirsizlik Giderici

Biçimbilimsel çözümleyici ve belirsizlik giderici, bir girdideki birimlerin eklerin ve köklerinin bulunması ve olası sonuçlar arasından doğrusunun seçilmesinden sorumlu bileşendir. Bu bileşende, üst tarafta bulunan girdi alanına yazılan cümleler, *Analyze/Disambiguate* butonuna basıldığında çözümlenecek ve aşağıdaki tabloda gösterilecektir. Çözümleme sonucunda doğru olduğu tespit edilen çözümlemeler bu



Şekil 3: Yazım Denetleyici



Şekil 4: Biçimbilimsel Çözümleyici/Belirsizlik Giderici

tabloda en üstte yer almaktadır ve yanlarına doğru olduklarını belirtecek bir imge konmuştur. Bu tablo da, *Download Results* butonu ile biçimli bir şekilde indirilip işlenebilir. Diğer bileşenlerin aksine, bu arayüzde girdi 1000 karakter ile sınırlandırılmıştır. Bu arayüz de Şekil 4'de incelenebilir.

V. SONUÇ

Bu çalışmada, şu ana kadar geliştirilmiş olan Türkçe DDİ araçlarına açık kaynak kodlu ve çevrimiçi arayüze sahip bir yazılım paketi ekledik. Sunduğumuz bu arayüz, belirli kısıtlarla, bir tarayıcı üzerinden Türkçe karakter dönüştürücü, birimlendirici/cümle bölücü ve yazım denetleyici ve biçimbilimsel çözümleyici/belirsizlik giderici bileşenlerini kullanmaya olanak sağlamakta ve işlemin sonucunu bir bilgisayar programı tarafından işlenebilecek bir biçimde sunmaktadır.

Bu arayüzü geliştirmek adına, arayüz hali hazırda yapılmış olan diğer araçlarla da uyumlu çalışacak bir yapıya sokulabilir. Bu şekilde, bütün Türkçe DDİ yazılımları tek bir çatıda toplanacak ve kolaylıkla erişilebilecektir.

Sunduğumuz bu yazılıma katkı sağlamak için atılabilecek bir başka adım ise, var olan bileşenlerin performansını iyileştirmektir. Her yapay zeka yazılım gibi Nlptoolkit de mükemmel değildir ve açık kaynak kodlu olmasının da yardımıyla, Türkçe DDİ geliştiricilerinin katkılarıyla daha iyi çalışacak hale getirilebilir.

TEŞEKKÜRLER

Bu çalışma Tübitak 116E104 nolu proje tarafından desteklenmiştir.

KAYNAKLAR

[1] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, 2017.

[2] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[3] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[4] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[5] "Nlptoolkit," http://haydut.isikun.edu.tr/nlptoolkit.ui-1.0/, accessed: 2019-02-08.

[6] "Türkçe karakter dönüştürücü," https://github.com/olcaytaner/Deasciifier, accessed: 2019-02-07.

[7] "Birimlendirici/cümle bölücü," https://github.com/olcaytaner/Corpus, accessed: 2019-02-07.

[8] "Yazım denetleyici," https://github.com/olcaytaner/SpellChecker, accessed: 2019-02-07.

[9] "Biçimbirimsel çözümleyici," https://github.com/olcaytaner/MorphologicalAnalysis, accessed: 2019-02-07.

[10] "Biçimbirimsel belirsizlik giderici," https://github.com/olcaytaner/MorphologicalDisambiguation, accessed: 2019-02-07.

[11] "opennlp," https://github.com/apache/opennlp, accessed: 2019-02-06.

[12] "Zemberek," https://github.com/ahmetaa/zemberek-nlp, accessed: 2019-02-06.

[13] G. Eryiğit, "ITU Turkish NLP web service," in *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Gothenburg, Sweden: Association for Computational Linguistics, April 2014.