



Açık Kaynak Kod Türkçe Doğal Dil İşleme Kütüphanelerinin Robotik Uygulamalarda Kullanımı

Kadir Aram^{a1,*}, Gökhan Erdemir^{b2}, Burhanettin Can^{a3}

^a Fatih Sultan Mehmet Vakıf Üniversitesi, Mühendislik Fakültesi Bilgisayar Mühendisliği, İstanbul, Türkiye

^b İstanbul Sabahattin Zaim Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, İstanbul, Türkiye

Istanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi (2021) 3 (1): 133-137

<https://doi.org/10.47769/izufbed.880143>

ORCID ¹0000-0002-5780-6334; ²0000-0003-4095-6333; ³0000-0002-0998-4371

YAYIN BİLGİSİ

Yayın geçmişi:

Gönderilen tarih: 14 Şubat 2021

Kabul tarihi: 12 Nisan 2021

Anahtar kelimeler:

Doğal Dil İşleme

Robotik Kontrol

Açık Kaynak

ÖZET

Dünya üzerinde doğal dil işleme ile ilgili birçok çalışma yapılmaktadır. Bunların bir kısmı "açık kaynak kodlu" olarak yayınlanırken, bazıları da kişilerin kullanımına izin vermekle beraber kaynak kodunu kapalı tutmaktadır. Günümüzde birçok farklı kullanım alanına sahip doğal dil işleme, özellikle mobil uygulamaların yaygınlaşmasıyla yazılım dünyasında kendisine daha fazla yer bulmaya başlamıştır. Mobil teknolojilerin günlük hayatta kullanımının kolaylaştırılması için insan-bilgisayar etkileşimi alanında araştırmalar hızla devam etmektedir. Dokunmatik ve küçük klavyelerin çok etkin kullanılmaması beraberinde sesli komut verme ve beraberinde komutların algılanması-işlenmesi süreçlerinde de gelişmelere yol açmıştır. Bu gelişmelere paralel olarak sadece mobil uygulamalarda değil diğer tüm uygulama alanlarında da doğal dil işleme konusunda çalışmalar yaygınlaşmıştır. Dil bilim, sinyal işleme vb. farklı uzmanlıklar gerektiren doğal dil işleme, yazılım projelerinde kütüphaneler veya yazılıma eklenmiş olan ek yazılımlar ile gerçekleştirilmektedir. Bu sayede birçok farklı türdeki uygulamaya entegrasyonu kolaylıkla sağlanabilmektedir. Türkçe üzerine yapılan çalışmalara baktığımızda birkaç tane önemli uygulama ve kütüphane karşımıza çıkmaktadır. Bu çalışmada, açık kaynak kodlu olarak sunulan Türkçe Doğal Dil İşleme çalışmaları ile ilgili öz bir değerlendirme yapılmış ve bu çalışmalardan yararlanarak kullanıcıdan alınan bir komutun, gezgin robot sisteminde kullanılması gösterilmiştir.

Use of Open Source Turkish Natural Language Processing Libraries in Robotic Applications

ARTICLE INFO

Article history:

Received: 14 February 2021

Accepted: 14 April 2021

Key words:

Natural Language Processing

Robotic Control

Open Source

ABSTRACT

There have been many studies on natural language processing over the world. While some of them are published as "open source", some of them allow the use of individuals but keep the source code closed. Natural language processing, which has many different usage areas, has started to find more place in the software world, especially with the widespread use of mobile applications. Researches continue in the field of human-computer interaction in order to facilitate the use of mobile technologies in daily life. The fact that touch and small keyboards cannot be used very effectively has led to improvements in the giving voice commands and the perception-processing of commands. In parallel with these developments, studies on natural language processing have become widespread not only in mobile applications but also in all other application areas. Linguistics, signal processing, etc. Natural language processing, which requires different expertise, is carried out in software projects with libraries or additional softwares. In this way, it can be easily integrated into many different types of applications. When we look at the studies on Turkish, we come across a few important applications and libraries. In this study, a self-evaluation has been made about the Turkish Natural Language Processing studies, which are presented as open source and by using these studies, a command received from the user has been shown to be used in a mobile robot system.

1. Giriş

Robotlar, ev işlerinde, sağlık sektöründe, eğitim alanında, ofis binalarında, değişik hizmet alanlarında, seri üretim hatlarında olmak üzere günlük hayatın hemen hemen her alanında

kullanılmaya başlanmıştır. Özellikle insan ile iletişimin yoğun olduğu alanlarda kullanım gittikçe artmaktadır (Asif & Yannier, 2003). Robotların uygulama alanları genişledikçe, insanlar/kullanıcılar ile robotlar arasındaki etkileşim daha da karmaşık hale gelmektedir. Bu nedenle İnsan Robot Etkileşimi

* Sorumlu yazar: karam@fsm.edu.tr

üzerinde yapılan araştırmalar, çok yaygınlaşmıştır. İnsanlar ve robotlar arasındaki iletişim sağlanması için değişik yöntemler kullanılmaktadır. Bunlara örnek olarak el kol hareketleri (Yang, Park, Lee, & Member, 2007) ve doğal dillerin kullanımı verilebilir (Kılıçaslan & Tuna, 2015).

Doğal Dil İşleme (DDİ), temel olarak bir doğal dili işleyerek onu anlayabilen ve yorumlayabilen bilgisayar sistemlerinin tasarımını inceleyen bir bilim dalı ve dil biliminin bir alt alanıdır (Çakıroğlu & Özyurt, 2006). Her dilin kendine has özellikleri olmasından dolayı bir dil üzerinde yapılan çalışma diğer bir dile birebir uygulanamamaktadır. Bu sebepten her farklı dil için uygun çalışmaların yapılması gerekmektedir.

Bu alanda yapılan çalışmalar sonucunda birçok dili destekleyen kütüphaneler oluşturulmuştur. Özellikle Türkçe dışında diller için kullanılabilir kütüphanelere örnek olarak NLTK (Loper & Bird, 2004), Stanford CoreNLP (Manning, Bauer, Finkel, & Bethard, n.d.), Gensim (Řehůřek & Sojka, 2011), SpaCy ve TextBlob (Loria, 2020) verilebilir. Bu kütüphaneler bu alanlarda çalışma yapacak kimselere programlama için bir ara yüz sağlarlar.

DDİ üzerine yapılan çalışmaların çoğalması, Türkçe özelinde böyle bir ara yüze olan ihtiyacı artırmıştır. Bu alanda ortaya çıkan yazılımların içerisinde açık kaynaklı olanlarla birlikte, kodu kapalı olup sadece kullanım için sunulanlar da vardır. İTÜ Doğal Dil İşleme Yazılım Zinciri (Eryiğit, 2014) araştırmacılara çevrimiçi olarak birçok özellik sunmasına rağmen açık kaynak kodlu değildir. Açık kaynak kodlu kütüphanelere örnek olarak Zemberek verilebilir (Akın & Akın, 2007). Zemberek Java dili ile yazılmış bir kütüphanedir ve daha çok yazılımcılara hitap etmektedir. Kullanıcı ara yüzü yoktur. Yine Türkçe NLPtoolkit açık kaynak kodlu olup, Zemberek'ten farklı olarak Türkçe doğal dil işleme alanında çalışan kullanıcılar için hazırlanmış bir ara yüzü de vardır (Uludo, Özçelik, Parlar, Ercan, & Yıldız, 2019).

Bu çalışmada, açık kaynak kodlu çalışmalardan birisi olan Türkçe Cümle Çözümleyici (Zafer, 2011) kullanılarak, sesli Türkçe komutlar ile verilen görevleri icra edebilen bir gezgin robot sisteminin kontrolü gerçekleştirilmiştir. Çalışmamızın 2. bölümünde Türkçe DDİ için kullanılabilir bazı çalışmalara yer verilmiştir. 3. bölümde bu çalışmalardan yararlanarak oluşturduğumuz uygulama sunulmuştur. 4. bölümde simülasyon sonuçları ve son olarak sonuç bölümünde elde edilen çıktılar bulunmaktadır.

2. Materyal ve Yöntem

Türkçe doğal dil işleme üzerine yapılan çalışmalara bakıldığında en sık karşılaşılan açık kaynak kodlu olarak hazırlanmış olan Zemberek uygulamasıdır. Zemberek kendi içerisinde normalizasyon modülü, morfoloji analizi, özel isim bulma, cümle sonu bulma, düzeltme, dil tanıma gibi özellikler barındırmaktadır (Akın & Akın, 2007).

İTÜ Doğal Dil İşleme Yazılım Zinciri içerisinde Türkçe karakter dönüştürücü, yazım denetleyici, bağımlılık analizi, cümle bölücü, yazım denetleyici gibi araçlardan oluşmaktadır (Eryiğit, 2014) Bu platform, kullanıcılara bir web ara yüzü ve programlama ara yüzü sunmaktadır; fakat açık kaynak kodlu değildir.

NLPToolkit de açık kaynak kodlu olup, yazılım bilgisi olmayanlar için bir ara yüz de sunmaktadır (Uludo et al., 2019). NLPToolkit içerisinde Türkçe karakter dönüştürücü, cümle bölücü, yazım denetleyici, biçimbirimsel çözümleyici gibi modüller bulunmaktadır.

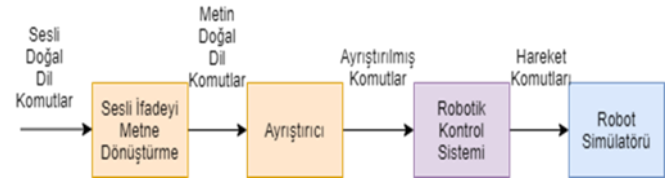
Bu çalışmalardan bir diğeri ise Türkçe üzerinde cümle analizi yapan "Türkçe Cümle Çözümleyici" isimli uygulamadır (Zafer, 2011). Bu uygulama Türkçe ve Türki diller için cümle analizi yapılması için başlanmış olup bu konu hakkında

çalışma yapmak isteyenler için bir referans olabilir. Çalışma açık kaynak olarak sunulmuştur.

TRMorph, biçimbirimsel analiz yapan açık kaynak kodlu araçtır. Ayrıca biçimbirimsel parçalama, bilinmeyen kelimeleri tahmin etme, gövdeleme ve kök çözümleme, heceleme, biçimbirimsel belirsizlik giderme gibi özellikler sunar (Çöltekin, 2014). Nüve ise biçimbirimsel analiz, N-gram ayıklama, gövdeleme, cümle sonu belirleme işlevlerini barındırır ("Türkî Diller için Doğal Dil İşleme Kütüphanesi: Nüve," n.d.).

Bu çalışmada, Türkçe Doğal Dil İşleme üzerine yapılmış açık kaynak kodlu araçları kullanarak, kullanıcıdan bir robota gönderilen yönlendirme komutları analiz edilmiş ve robotun hareket ettirilmesi sağlanmıştır. Zemberek ve Türkçe Dil Çözümleyici üzerinde çalışılarak cümle analizi yapılmış ve simülasyon ortamında denemesi yapılmıştır. Çalışma, Robot Operating System (ROS) içerisinde bulunan TurtleSim simülasyon ortamında denemiştir. ROS, adı işletim sistemi olsa da aslında robot yazılımları geliştirmek için birtakım kütüphanelerin bulunduğu bir platform olarak anlatılabilir. İçerisinde kütüphanelerin dışında simülasyon ve tasarım araçları da bulundurmaktadır ("Robot Operating System," n.d.). TurtleSim, kaplumbağa şeklinde bir robotu barındıran, grafik penceresi şeklinde bir uygulamadır ("TurtleSim," n.d.).

Şekil 1'de gerçekleştirilen sistemin genel yapısı görülmektedir.



Şekil 1. Sistemin blok diyagramı

2.2. Sistem Metine Dönüştürme

Sisteme verilen sesli komutlar bu modül ile metine dönüştürülür. Metine dönüştürme işlemi için SpeechRecognition kütüphanesi kullanılmıştır. ("SpeechRecognition," n.d.).

2.3. Ayrıştırıcı

Robot sistemine gönderilecek olan Türkçe doğal dil ile ilgili ifadeler, öncelikle ayrıştırıcı ile analiz edilmektedir. Bu nedenle bu kısımda komut cümlesi alınıp analiz edilerek yüklem ve tümleci belirlenmektedir. Burada analiz için Zemberek ve Türkçe Cümle Çözümleyici (TCÇ) kullanılmıştır ("Türkçe Dil Çözümleyici Fatih Parser," n.d.). TCÇ ve rilen cümledeki kelimelerin analizi için Zemberek kullanılır.

Buradan elde edilen morfolojik analize göre ayrıştırma yapılmaktadır. Bu ayrıştırma işlemi gerçekleştirmek için İçerikten Bağımsız Gramer (Context Free Grammar (CFG)) kullanılmıştır. CFG, doğal dillerin yapısını ve söz dizimi parçalarını açıklayabilmek için kullanılan sistemlerden biridir (Jurafsky, D. James, Martin 2009). Fakat anlam ifadeleri tanımlanmadığı için doğal dilin tüm özelliklerini çözümleyemez. Çoğunlukla programlama dilleri, içerikten bağımsız gramerler kullanılarak tasarlanmıştır. Ayrıştırıcı içerisinde bu yapı kullanılarak, cümle için oluşabilecek tüm olası kombinasyonlar ortaya çıkarılmaktadır. Sistem, ayrıştırma işlemi yaparken bunların bir kısmını elemektedir. Buna rağmen birden fazla olası çözüm mevcuttur. Yapılan uygulama için ise sadece eylem cümleleri kullanılmıştır ve öncelikle elde edilen olası çözümlerin frekansları belirlenmiştir. Her benzer çözüm

aynı sınıfın elemanı olarak düşünülmüş ve çözüm sayısı fazla olmaması sebebi ile sınıf sayısını bir sınırlandırma getirilmemiştir. Her bir sınıfın frekans yüzdesi aşağıda gösterilen (1) numaralı denkleme göre hesaplanmıştır.

$$p_i = \frac{f_i}{n} \quad i = 1,2,3 \dots, k \quad \sum_{i=1}^k p_i = 1 \quad (1)$$

Denklemden p_i ilgili sınıfın frekans yüzdesini, f_i frekansını, n toplam veri sayısını, k ise sınıf sayısını göstermektedir. Bulunan çözümler içerisinde oranı en yüksek olan doğru ayrıştırma olarak kabul edilerek sistem için kullanılmıştır.

Aşağıda "Robot ileriye git" cümlesi için sistem tarafından gerçekleştirilen analiz verilmektedir:

"robot/isim", "ileriye/isim", "git/fiil"

"özne/robot", "tümleç/ileri", "yüklem/git"

Bu şekilde yapılan analiz ile cümlelerin yüklemine göre yapılacak iş, cümlelerin tümlecisi ile de yönü elde edilmektedir. Bu adımın ardından ayrıştırma işlemine, iki işin sıralı olarak yapılabilmesi özelliği için bağlaç kullanımı eklenmiştir. Verilen komut cümlesi içerisinde "sonra" bağlacı eklenerek eylemlerin iş sırası belirlenmiştir. Örnek olarak "Robot ileriye git sonra sola dön" cümlesi için aşağıdaki gibi bir analiz yapılmaktadır.

Birinci yapılacak iş: "özne/robot", "tümleç/ileri", "yüklem/git"

İkinci yapılacak iş: "tümleç/sol", "yüklem/dön"

Örnekte görüldüğü üzere "sonra" zaman ve sıra bildiren bir ifadedir. Ayrıştırıcı da bu ifade ile karşılaştığında eylemlerin sırasını belirler ve robotik kontrol sistemine veriler gönderilirken bu sıra ile işlerin gerçekleşmesi sağlanır.

2.4. Robotik Kontrol Sistemi

Robotik Kontrol Sistemi (RKS), ayrıştırıcıdan gelen ifadeleri alarak onları robot üzerinde çalışacak ifadelere dönüştürür. Ayrıştırıcı doğal dildeki komutları ayrıştırırsa da bunlar hala robotun işleyebileceği komutlar değildir. Bu aşamada sadece yapılması gereken işlem ve yön bilgisi elde edilmiştir. Bunun için yeni bir dönüştürme işlemine ihtiyaç vardır. Bunların eyleme dönüştürülmesi için bir kısım fonksiyonlar gerekir. RKS, sistemden alınan komutlar ile robot sürücüsü arasında köprü vazifesi görür. Kendine gelen komutları ilgili fonksiyonlara aktararak robot sisteminin yapacağı işlevi belirler.

İşlevler kullanılacak robot sistemine göre düzenlenebilir. Bu çalışmada bir düzlem üzerinde hareket işlevi gerçekleştirildiğinden gitme ve dönme eylemleri yerine getirilmektedir.

2.5. Robot Simülasyonu

Çalışmanın gerçekleştirilmesi için Robot Operating System (ROS) üzerinden Turtlesim simülatörü kullanılmıştır. ROS, bilgisayar kullanarak robotların kontrol edilmesini sağlayan açık kaynak kodlu bir yazılımdır. Kullanıcı ve robot arasında bir ara yüz oluşturur. Turtlesim ise ROS üzerinde bulunan basit bir simülasyon ortamıdır. Yapılan projenin hızlı bir şekilde test edilmesi için bu platform kullanılmıştır. Simülasyonda robotun hızı sabit tutularak 0,5 m/s olarak

deneme yapılmıştır. Aldığı mesafe ise kullanıcı tarafından komut ifadesinde belirtilmemişse 1 m olarak alınmıştır.

3. Bulgular ve Tartışma

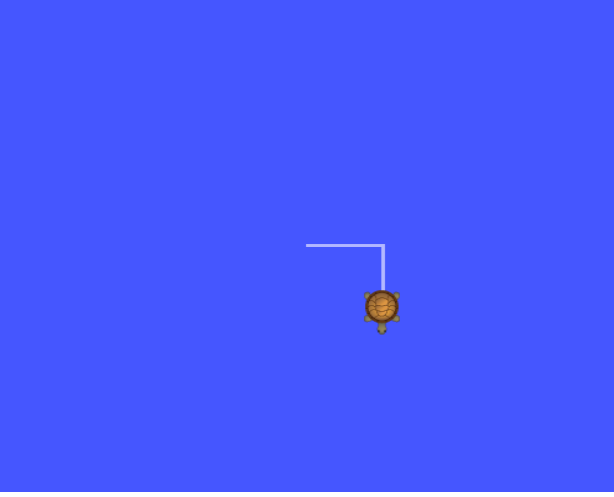
Bu çalışmada sınırlı bir test ortamı kullanılarak belirli eylemler gerçekleştirilmiştir. Test aşamasında yapılan her bir denemede robotun başlangıç konumu sabittir ve merkezde yer almaktadır. Robotun alacağı mesafe belirtilmediği sürece 1 m, hızı ise her zaman 0,5 m/s olarak belirlenmiştir. Ayrıştırıcı genel olarak tüm eylemleri ayrılabilmesine rağmen RKS üzerinde sadece "gitmek" ve "dönmek" eylemleri dikkate alınarak bu eylemler için gerekli fonksiyonlar hazırlanmıştır. Gitme işlevi kendi içerisinde yön bilgisine göre gerçekleşmektedir. Bu yön bilgileri "ileri" ve "geri" şeklinde belirtilmiştir. Dönme işlevi ise "sol" ve "sağ" şeklinde yön bilgisi almaktadır.

İlk test işleminde, tek işlevli bir komut olarak "Robot ileriye git" ifadesi kullanılmıştır. Bu ifade, önce ayrıştırıcı modülüne girer. Ayrıştırıcıda yapılacak eylemin gitme işlevi olduğu ve tümleç olarak da yön bilgisi içeren ileri kelimesi olduğu belirlenir. Bu ifadeler RKS içerisinde gerekli fonksiyonlar üzerinde çalıştırılarak simülasyon ortamında gerçekleştirilir. Şekil 2'de bu işlemin simülasyon sonucu görülmektedir.



Şekil 2. "Robot ileriye git" komutu için simülasyon görünümü

İki işlevli ifade için "Robot ileriye git sonra sağa dön" örneği sisteme verilmiştir. Bu komut incelendiğinde "sonra" ifadesi algılanmaktadır. Burada RKS'ye birinci işlevin ileri gitme olduğu ikinci işlevin ise sağa dönme olduğu bilgisi gönderilir. Şekil 3'te iki işlevli komutun simülasyon ortamında art arda çalıştırılmasının sonucu gösterilmiştir.



Şekil 3. "Robot ileriye git sonra sağa dön" komutu için simülasyon görünümü.

Başlangıçta belirtildiği üzere robotun alacağı mesafe komutta belirtilmediği sürece 1 m olarak alınmıştır. Fakat kullanıcı robota bunu da söyleyebilmektedir. Örnek olarak "Robot iki metre ileri git sonra iki metre sola git" komutu ele alındığında kullanıcı tarafından bildirilen bir mesafe söz konusudur. Burada yine ayrıştırıcı tarafından alacağı mesafe ve yönü belirlenip RKS ile robota iletilmiştir. Şekil 4'te mesafe belirtilerek robotun ilerlemesi istenen komutun sonucu görülmektedir.



Şekil 4. "Robot iki metre ileri git sonra iki metre sola git" komutu için simülasyon görünümü

Simülasyon sonuçlarından görüldüğü gibi sistem, komutları başarıyla gerçekleştirmiştir.

4. Sonuç

Yapılan çalışma üç ana aşamada gerçekleştirilmiştir. Öncelikle alınan sesli komutlar metine dönüştürülmüştür. Ardından bu komutlar açık kaynak kodlu Türkçe doğal dil işleme çalışmalarından yararlanılarak işlenmiş ve ayrıştırılmıştır. Son olarak ayrıştırılan bu komutlar, gezgin robot sisteminde denenmiştir. Simülatör üzerinde başarılı sonuçlar elde edilmiştir.

Çalışma belirli kısıtlar dâhilinde gerçekleştirilmiştir. Belirli

bir düzlem kullanılmış ve sadece "gitme" ve "dönme" olarak iki adet işlev gerçekleştirilmiştir. Bu işlevlere katkı olarak ileri, geri, sağ ve sol olmak üzere yön bilgileri ile gidilecek mesafe bilgisi de işlenmiştir.

Gelecek çalışmalarda simülasyon ortamının daha da genişletilmesi ve robot üzerinde kullanılan eylemlerin çeşitlendirilmesi düşünülmektedir.

Kaynaklar

- Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source NLP framework for Turkic Languages. *Structure*, 10, 1–5.
- Asif, Ş., & Yannier, S. (2003). Robotlar: Sosyal Etkileşimli Makineler. *Journal of Thermal Sciences and Technology*, 1–6. Retrieved from https://www.researchgate.net/publication/255615207_Robotlar_Sosyal_Etkileşimli_Makineler_Robots_as_Socially_Interacting_Machines
- Çakıroğlu, Ü., & Özyurt, Ö. (2006). Türkçe Metinlerdeki Yazım Yanlışlarına Yönelik Otomatik Düzeltme Modeli. *Elektrik, Elektronik ve Bilgisayar Mühendisliği Sempozyumu ve Fuarı (ELECO 2006)*, Aralık, 6-10., 1–5.
- Çöltekin, Ç. (2014). A set of open-source tools for Turkish natural language processing. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik.
- Eryiğit, G. (2014). ITU Turkish NLP Web Service. *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 1–4. Gothenburg.
- Kılıçaslan, Y., & Tuna, G. (2015). An Nlp-Based Approach for Improving Human-Robot Interaction. *Journal of Artificial Intelligence and Soft Computing Research*, 3(3), 189–200. <https://doi.org/10.2478/jaiscr-2014-0013>
- Loper, E., & Bird, S. (2004). NLTK: the Natural Language Toolkit. *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 214–217. <https://doi.org/10.3115/1118108.1118117>
- Loria, S. (2020). *textblob Documentation*.
- Manning, C. D., Bauer, J., Finkel, J., & Bethard, S. J. (n.d.). *The Stanford CoreNLP Natural Language Processing Toolkit*.
- Řehůřek, R., & Sojka, P. (2011). Gensim — Statistical Semantics in Python. *EuroScipy 2011- The 4th European Meeting on Python in Science*, 6611, 6611.
- SpeechRecognition. (n.d.). Retrieved November 10, 2020, from <https://pypi.org/project/SpeechRecognition/>
- Türkçe Dil Çözümleyici Fatih Parser. (n.d.). Retrieved November 1, 2020, from <https://github.com/hrzafer/fatih-parser>
- Türkî Diller için Doğal Dil İşleme Kütüphanesi: Nuve. (n.d.). Retrieved November 28, 2020, from <http://hrzafer.com/turki-diller-icin-dogal-dil-isleme-kutuphanesi-nuve>

TurtleSim. (n.d.). Retrieved November 28, 2020, from <http://wiki.ros.org/turtlesim>

Uludo, G., Özçelik, R., Parlar, S., Ercan, G., & Yıldız, O. T. (2019). Türkçe Doğal Dil İşleme için Arayüzler User Interfaces for Turkish Natural Language Processing. 27. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU2019), (April).

Yang, H., Park, A., Lee, S., & Member, S. (2007). Gesture

Spotting and Recognition for Human – Robot Interaction. 23(2), 256–270.

Zafer, H. R. (2011). A Generic Syntactic Parser for Turkic Languages.