

TÜRKÇE METİNLERDEKİ YAZIM YANLIŞLARINA YÖNELİK OTOMATİK DÜZELTME MODELİ

Ünal ÇAKIROĞLU¹

Özcan ÖZYURT²

¹Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, Fatih Eğitim Fakültesi
Karadeniz Teknik Üniversitesi, Söğütlü, Trabzon

²Teknik Programlar Bölümü, Beşikdüzü MYO

Karadeniz Teknik Üniversitesi, Beşikdüzü, Trabzon

¹e-posta: cakiroglu@ktu.edu.tr

²e-posta: oozyurt@ktu.edu.tr

Anahtar sözcükler: Doğal Dil İşleme, Morfolojik Analiz, Yazım Yanlış Düzeltme

ABSTRACT

The spelling errors are one of the most important problem in text editor programs and also in natural language processing studies. This paper presents an automatic spelling checking and correcting method for Turkish texts. The proposed method can correct many kinds of spelling errors. The frequencies of spelling errors are obtained by a statistical study and system modules are designed through these statistical data. Performance of the system is analyzed by corresponding system's performance and "Ms Word" word processor program. The proposed method is aimed to be an alternative method that can be used in morphological analysis phase of natural language processing studies.

1. GİRİŞ

Doğal Dil İşleme (DDİ), ana işlevi bir doğal dili çözümleme, anlama, yorumlama ve üretme olan bilgisayar sistemlerinin tasarımı konu alan bir mühendislik alanıdır [1-4]. DDİ, yapay zeka (bilgi gösterimi, planlama, akıl yürütme vb.) biçimsel diller kuramı (dil çözümleme), kuramsal dilbilim ve bilgisayar destekli dilbilim, bilişsel psikoloji gibi çok değişik alanlarda geliştirilmiş kuram, yöntem ve teknolojileri bir araya getirir.

DDİ ve yakın alanlarda yapılan araştırmalar, bir yanda işlenen dilin yapısal özelliklerinden bağımsız olma iddiasında kuramlar geliştirirken bir yandan da bunların geniş kapsamlı olarak uygulanması için işlenecek dillere özel kaynakların üzerine yoğunlaşmaktadır [3,5]. Ancak şu ana kadar geliştirilen kuramların çoğu İngilizce ve benzeri dilleri temel uygulama alanına aldığı için, çeşitli özellikleri ile bu tip dillerden farklı dillerde uygulamalarında sorunlar çıkabilmektedir [4,5,6].

Dilin işlenebilmesi için dilin kurallı veya kurlsız bütün yapılarının matematiksel (algoritmik) olarak

ifade edilebilmesi gerekir. Bu işlem, morfolojik, söz dizimsel ve anlamsal analiz aşamalarını kapsar [3,7].

Morfolojik analiz, amaç ne olursa olsun DDİ ile ilgili yapılacak tüm çalışmaların ön aşamasıdır. DDİ ile ilgili yapılan birçok çalışmada morfolojik analizin etkin performansa sahip olması gerekliliği üzerine vurgu yapılmaktadır [5,6,8]. Bu bildiride morfolojik analizdeki en önemli problemlerden olan yazım hatalarının tespiti ve düzeltilmesi ile ilgili yöntemler üzerine odaklanılmaktadır. Hataların tespitinde istatistiksel yöntemlerden yararlanılmıştır. Önerilen sistem gerek hata düzeltme, gerekse alternatif doğru sözcükleri önerme konularında yüksek düzeyde performans sağlamıştır. Sistem sık kullanılan metin editörlerindeki hata düzeltme modülleriyle karşılaştırılmış ve kullanılabilirliği test edilmiştir. Önerilen sistemin bu konuda yapılacak çalışmalar için bir alt yapı teşkil etmesi düşünülmektedir.

1. YAZIM HATALARININ TESPİTİ

Doğal dil işlemede karşılaşılan en önemli zorluklardan biri, doğal dilde girilen metinlerdeki ifadelerin hatalar içeriyor olmasıdır. Klasik metin editörleri, internet üzerinden iletişim ortamları, diyalog tabanlı sistemler ve çeviri sistemleri gibi akla gelebilecek birçok doğal dil işleme tabanlı sistemlerde bu durum önem taşımaktadır. Bu sebepten olayı, hataların tespiti ve düzeltilmesi gerekmektedir [5,9,10].

Metinlerin metin editörüne girilmesi esnasında oluşabilecek hatalar çeşitlilik göstermektedir. Bu durum, yetersiz imla bilgisi ya da klavye kullanım hatalarından kaynaklanmaktadır. Bu hataları tespit amacıyla en az 500 kelimelik 15 farklı metin 40 kişiye yazdırılmış ve bu doğrultuda istatistik yapılmıştır. Seçilen 40 kişinin eğitim durumlarının hata yapma olasılıklarını etkileyeceği düşünülerek ilgili örneklem Tablo-1'de sunulmuştur. İstatistikten elde edilen sonuçlar Tablo-2'de verilmiştir.

Tablo-1. Seçilen örneklemin eğitim düzeyi

Okur yazar	İlköğretim	Ortaöğretim	Üniversite
4	8	18	10

Tablo-2. Türkçe metinlerdeki yazım hatası türleri ve sıklıkları

Hata Türü	Hata Kaynağı		
	Tek Karakter	İki Karakter	İkiden fazla Karakter
1	% 4,8	% 2,6	% 0,02
2	% 0	% 4,3	% 1,2
3	% 3,5	% 0,67	% 0,003
4	% 3,7	% 0,32	% 0,001

İstatistikler hesaplanırken

Hata oranı = Hatalı kelime / Toplam kelime
Toplam Kelime = 500x 15x 40 = 300000

eşitlikleri kullanılmıştır.

Hata Türleri:

- (1) Yanlış yazılan karakter:
kitap yerine kirap gibi (Kaynak: tek karakter)
- (2) Yer değiştiren komşu karakterler:
kalem yerine kalme gibi (Kaynak: iki karakter)
- (3) Düşen karakter:
defter yerine dejt gibi (Kaynak: iki karakter)
- (4) Eklenen karakter:
örnek yerine önernk gibi (Kaynak: tek karakter)

Burada belirtilenler dışında kalan hatalar, bu hatalarla kıyaslandığında, görülme sıklıkları oldukça azdır. Dolayısıyla yapılan çalışmada bu hatalar dikkate alınmamıştır.

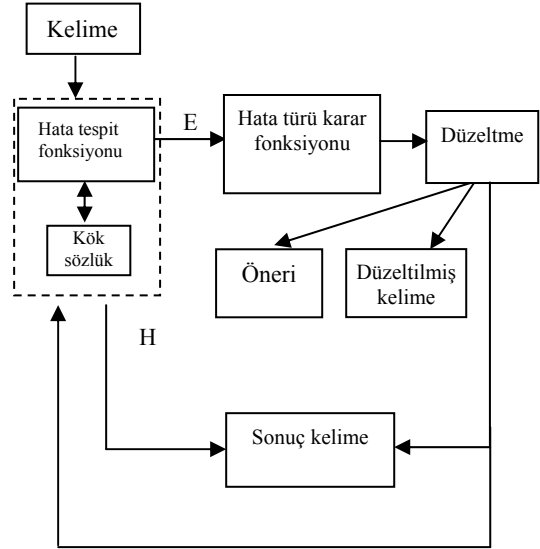
2. HATA DÜZELTME YÖNTEMLERİ

Türkçe, kural tabanlı bir dildir. Bir kelime, kök ve kökün sonuna eklenmiş eklerden oluşur [7,9,11]. Ek sayısı ortalama 3–5 arasındadır. Kelimede yapılmış hatalar, kelimenin ek almış ya da almamış oluşuna göre farklı özellikler gösterebilmektedir. Bunun yanı sıra, kelime eğer ek almış bir kelime ise, hatanın ekte veya yoksa kökte yapıldığı da ayrıca tespit edilmesi gereken bir durumdur. Bunların tespiti ve düzeltilmesi için, önerilen sistemde kök kelimelerden oluşan sözlük (Kök_sözlük) kullanılmaktadır. Bu sözlük 10000 civarında kelime içermekte olup güncellenmeye açıktır.

3. EK ALMAMIŞ KELİMELER İÇİN HATA DÜZELTME YÖNTEMLERİ

Ek almamış (yalın haldeki) kelimelerde yapılan hataların tespitinde Kök_sözlük kullanılır. Bunun için her bir kelime sözlükte aranır. Bulunursa, kelime hatasız olarak kabul edilir. Bulunamazsa kelime hatalı olarak işaretlenir ve kelimede olabilecek hata türleri

göz önüne alınır. Hata tespit ve düzeltme işlemleri için önerilen sistemin yapısı aşağıdaki gibidir.



Şekil-1. Ek almamış kelimeler için hata düzeltme sisteminin yapısı

Sistemin Modülleri:

Hata tespit fonksiyonu: Kelimenin hatalı kelime olup olmadığının kontrol edildiği modüldür. Bu işlemde kelime Kök_sözlük ile karşılaştırılır. Kök_sözlükte var olan kelime hatasız kabul edilir. Aksi durumda kelime hatalı kabul edilerek, hata türü karar fonksiyonu devreye sokulur.

Hata türü karar fonksiyonu: Bu modülde hatalı kelimenin hata türüne karar verilir. Bu işlemi gerçekleştirmek amacıyla Tablo1 deki istatistiklerden yararlanılır. Bu istatistikler doğrultusunda hata türleri öncelik sıralarına göre sıranır. İlk rastlanılan hata türü düzeltilmek üzere düzeltme modülüne gönderilir.

Düzeltilme: Hata türü tespit edilen kelimenin ilgili hata türüne göre düzeltme fonksiyonlarından geçirildiği modüldür. Hata türlerine göre düzeltme yöntemleri aşağıda kısaca özetlenmektedir.

Yanlış yazılan karakter: kitap yerine kirap gibi

Burada, hatalı kelime olarak “sayak” kelimesini göz önüne alalım. Hatalı kabul edilen bu kelimede, herhangi bir karakterin yanlış yazılmış olabileceği kabul edilerek verilen kelime için aşağıdaki yapılar çıkarılır:

“_ayak”, “s_yak”, “sa_ak”, “say_k”, “saya_” ; “_” tek bir karakter yerine kullanılmaktadır.

Bu yapıları uyan tüm kelimeler sözlükten listelenir. Bu durumda tablodaki alternatifler üretilir.

Tablo-3. Bir karakterin yanlış yazılmasına önerilen doğru alternatifler

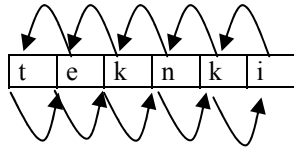
ayak	dayak
	kayak
s yak	
sa ak	saçak
	salak
	sapak
say k	sayık
saya	sayaç

Yer değiştiren karakter: *kalem yerine kalme* gibi

Bu tip hataların düzeltilmesi için, hatalı kabul edilen kelimedeki birinci karakterden itibaren karakterler ikili olarak yer değiştirilerek tüm alternatifler bulunur. Bu tür hata örneği için “teknki” kelimesini göz önüne alalım. Bu kelime için aşağıdaki yapılar oluşturulur:

“etknki”, “tkenki”, “tenkki”, “tekni”, “teknik”, “teknki”

Bu yapıya uyan tüm kelimeler sözlükten listelenir.



Şekil-2. Yer değiştiren karakterlerin düzeltilmesi

Düşen karakter: *defter yerine deftr* gibi

Bu tip hataların düzeltilmesinde, yazılan kelimenin herhangi bir karakteri yazılmamış olduğu için olası tüm durumlar çıkarılır. Örnek olarak “klak” ifadesini alalım. Bu ifade için;

“_klak”, “k_lak”, “kl_ak”, “kla_k”, “klak_”

yapıları çıkarılır. Bu yapıya uyan tüm kelimeler sözlükten listelenir.

Eklenen karakter: *örnek yerine örnerk* gibi

Bu tip hataların düzeltilmesinde, hatalı kelimenin herhangi bir karakteri fazla yazılmış kabul edilip her bir karakter tek tek atılır ve Kök sözlüğe uygun tüm alternatif durumlar üretilir. Örneğin, “hayast” ifadesine bakalım. Bu ifade için;

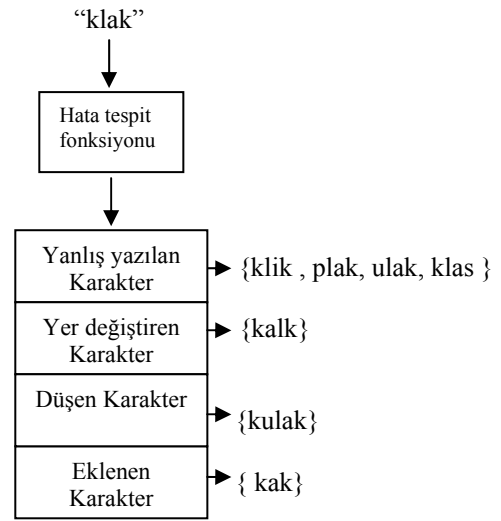
“ayast”, “hyast”, “haast”, “hayst”, “hayat”, “hayas”

Yapıları üretilir ve sözlükte aranır. Sözlükten uygun kelimeler listelenir.

Bazı durumlarda, hatalı yazılan bir kelime için birden fazla hata türü uygun düşebilmektedir. Diğer bir deyişle, herhangi bir hatalı kelime için bu düzeltme

yöntemlerinden birden fazlası sonuç üretmektedir. Bu tür durumlarda, hata türü karar fonksiyonu tarafından belirlenen hata öncelik sırası dikkate alınır. Bu amaçla yazılan yazılımda iki şekilde çözüm üretilebilmektedir. Birinci olarak; üretilen tüm kelimeler önerilmektedir. İkincisinde ise yapılan hata istatistikleri oranında hata türlerine öncelik tanınmakta ve önerilecek kelimelerin ihtimali hesaplanmaktadır. Bu şekilde önerilecek kelime sayısı azaltılarak doğru kelime önerilmeye çalışılmaktadır. Bu durumlar için örnekler aşağıda gösterilmiştir.

Hatalı kelime: “klak”



Şekil-3. Hata tespit ve hata türleri belirleme fonksiyonlarının birlikte işleyişi

4. EK ALMIŞ KELİMELER İÇİN HATA DÜZELTME YÖNTEMLERİ

Ek almış kelimelerde hata düzeltme teknikleri Türkçe'deki kelimelerin ek yapılarıyla ilişkilidir. Türkçe kural tabanlı bir dil olup, dilin gramer yapısı gereğince alabildiği ekler belirlidir. Diğer bir deyişle herhangi bir ekin hangi eklerden sonra gelebileceği (ek sıralamaları) belirlenmiştir. Bu yapıların dışında ekler rasgele olarak kelimelerin sonuna eklenmez. Aşağıda, Türkçe isim ve fiillerin genel türetilme yapısı verilmiştir [3].

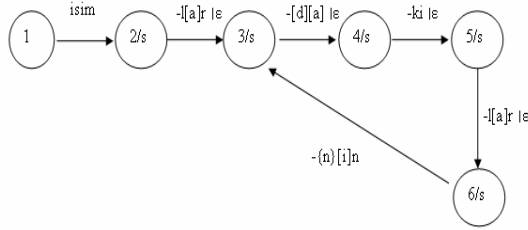
İsim: isim kökü + çoğulluk eki + iyelik eki + durum eki + ki eki

Fiil: fiil kökü + çatı eki + olumsuzluk eki + bileşik fiil eki + ana zaman eki + soru eki + yardımcı zaman eki + kişi eki

Dil matematiksel olarak modellenirken bilgisayar bilimlerinde tanıma amaçlı olarak sıkça kullanılan sonlu otomatalardan yararlanılır. Sonlu otomata kullanım alanlarına göre farklı makineler olarak

karşımıza çıkabilirler. Genişletilmiş Geçiş Ağları (GGA) bunlar arasında en çok kullanılanlardır.

Aşağıdaki şekil, sistemde kullanılan belirli sayıdaki isim türlerinin morfolitiklerini ayırıp, tanıyabilen bir GGA'yı göstermektedir:



Şekil-4. Türkçe isimsel morfolitikleri tanıyabilen basit bir GGA yapısı

Bu yapıda, /s sonlanabilme durumunu; [], içerisindeki harf için alternatif seslerin varlığını; ε; ilgili geçiş olmayabileceğini; {} ise içerisindeki harfin ses durumlarına göre hiç olmayabileceğini göstermektedir. Buradaki yapı ile tanımlanabilecek kelimelere bir örnek aşağıda verilmiştir:

Arabadakiler: Araba da ki ler
İsim [d][a] ki l[a]r

Ayrıca, Türkçe dilbilgisi ses uyumu kurallarına göre, bir önceki hecedeki sesli harf kalın ise ekteki sesli harfte kalın olacaktır. Tersisi durumunda, ekteki sesli harfte ince sesli olacaktır [12]. Örneğin, çoğulluk eki “-ler ve lar” dır. “ayak” kelimesi çoğul yapılırken, ses kurallarına göre “-lar” eki kullanılarak “ayaklar” kelimesi oluşturulacaktır. “ayaklar” şeklinde bir kelime geçerli bir kelime olmayacaktır. Benzeri durumda, “iyi” kelimesinin çoğulu yapılırken aynı kural gereği “-ler” eki kullanılarak “iyiler” kelimesi oluşturulur.

Bu kurallara göre, “ayaklarım” kelimesi, Türkçe dilbilgisi kurallarına göre geçerli bir kelime olup, “ayağımlar” kelimesi geçerli bir kelime değildir. Sistem bu şekilde tanımlanmış genişçe bir GGA yapısını bünyesinde barındırmaktadır. Bu GGA tanımlamalarıyla ek sıraları dikkate alınarak hatalı olan ek öbeği bulunmakta, bunun için doğru ek önerilebilmektedir.

Örneğin “okulumuzdeki” kelimesi için tanımlanan GGA ile kelime okul- (u)muz-(deki) biçiminde ayrıştırılmaktadır. Burada (deki) hatalı ek öbeği şeklinde işaretlenmekte ve tanımlanan GGA yapısından “(u)muz ” ekinden sonra “-de” eki gelemeyeceği bilgisi elde edilmektedir. Bu ek için “-a , -da, -dan, -u, -ca, ...” gibi alternatifler önerilir. Bu ekin ardından gelen “-ki” eki GGA daki tanımlamadan dolayı ilgili ekin “-da” olabileceğini kuvvetlendirir.

Yapılan deneysel çalışmalarda, ek almış kelimeler için önerilen yöntemin belirli bir başarı düzeyine ulaştığı gözlenmiştir. Bazı sözcüklerde aynı anda kelimenin hem kök hem de ek kısmında hata olabilmektedir. Bu durumda, Levenshtein uzaklık algoritması (Levenshtein Distance Algorithm) devreye sokularak sistem performansı artırılmıştır.

Bu yöntem, iki karakter dizisinin arasındaki benzerliğin bulunmasına dayanır. Kaynak ve hedef karakter dizisi olarak adlandırılan iki parametreden, hedef karakter dizisinde, kaynak karakter dizisinden silinen, eklenen ya da değişime uğrayan karakterler farkı bulunur. Örneğin,

Kaynak “test” ve hedef “test” olsun. Bu durumda, uzaklık (kaynak, hedef) = 0 olacaktır. Çünkü hedefle kaynak arasında hiçbir değişim olmamıştır.

Kaynak “test” ve hedef “tent” olsun. Bu durumda ise uzaklık (hedef, kaynak) = 1 olacaktır. Çünkü hedefle kaynak arasında s harfi n’ye dönüşmüştür.

Kelimenin alabileceği ekler belirli olduğu için, her seferinde, kelimeye eklenebilecek ekler kelimeye eklenir ve yanlış olarak işaretlenen kelime ile karşılaştırılır. İki karakter dizisi arasındaki fark 1 ise, bu durumda o kelime doğru kelime olarak önerilir. Aşağıda, bu türden hataların yapıldığı ve önerilen kelimeler verilmiştir:

Hatalı kelime: “ayağm”

kelimenin kökü “ayak” olarak işaretlenir. Burada kelimenin uzunluğu da karşılaştırılacak karakter dizisi için önem taşımaktadır. Diğer bir deyişle, kelimenin uzunluğuna göre kelimenin köküne eklenecek eklerin sayısı belirlenmektedir. Bu durumda, ayak kelimesinin alabileceği ek sayısı 1 olacaktır. Bu ekler aşağıda listelenmiştir:

“-i”, “-ım”, “-a”, “-ta”, “-la”, “-lar”

Bu eklerin eklenmesiyle aşağıdaki kelimeler oluşacaktır.

“ayağı”, “ayağım”, “ayağa”, “ayakta”, “ayakla”, “ayaklar”

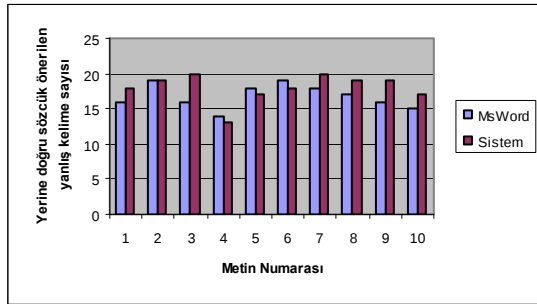
Bu kelimeler, “ayağm” kelimesi ile karşılaştırılarak, aralarındaki uzaklık değeri bulunur. Bu kelimeler sırasıyla aşağıdaki uzaklık değerlerini üretir.

Uzaklık(ayağı,ayağm) = 1
Uzaklık(ayağım, ayağm) = 1
Uzaklık(ayağa,ayağm) = 1
Uzaklık(ayakta,ayağm) = 3
Uzaklık(ayakla,ayağm) = 3
Uzaklık(ayaklar,ayağm) = 4

Yukarıdaki değerlere göre, “ayağm” kelimesine karşılık olarak “ayağı”, “ayağım”, “ayağa” kelimeleri üretilir.

5. SONUÇLAR

Yapılan deneysel çalışma sonucunda sistem hata düzeltmede % 71, doğru kelime önermede %98 başarı sağlamaktadır. İçerisinde 20 adet yanlış yazılmış (her hata türünden farklı sayılarda) 500 kelimelik 10 metin için yapılan analizde en sık kullanılan Ms. Word yazılımıyla önerilen sistemin Türkçe hatalı sözcüklere karşılık gelen doğru sözcükleri önerme performansları aşağıdaki gibi olmuştur.



Şekil-5. Sistemin “doğru kelime önerisi” bakımından Ms. Word ile karşılaştırılması

Buradaki performanslar bulunurken hatalı kelimelerin yerine önerilen doğru sözcük sayıları iki sistem için birbirine oranlanmış, her iki sistem arasında en çok %15 oranındaki fark olması durumunda o sözcük için performans eşit kabul edilmiştir. Farkın %15’ten yüksek olduğu durumlarda, performansı iyi olan sistem için “doğru öneri” diğeri için “yanlış öneri” sunduğu kabul edilmiştir. Örneğin “Başakı” sözcüğü için Ms. Word 5 sözcük önerirken, sistem 6 sözcük önermektedir. Aradaki fark % 15 ten büyük olduğundan sistemin daha doğru öneri bulunduğu düşünülmüştür.

Sistemi gerçekleştirmek için yazılan yazılım esnek bir yapıya sahip olduğundan birçok metin editörüne adapte edilebilir durumdadır. Bu da, sistemin avantajları arasında sayılabilir.

Burada modüllerinden kesitler sunulan çalışma gerçek zamanlı olarak koşturulmuş ve olumlu sonuçlar elde edilmiştir. Bu durumda, özellikle morfolojik analiz için girdi olan kelimenin yapısal özellikleri değerlendirildikten sonra en uygun yöntemin seçilerek devreye sokulmasıyla hata düzeltme işlemi gerçekleştirilebilmektedir.

KAYNAKLAR

- [1] Rich E. Artificial Intelligence, McGraw Hill Inc., Second Edition, Newyork, 1991.

- [2] Öztaner, M. A Word Grammar of Turkish with Morphophonemic Rules, Yüksek Lisans Tezi, ODTÜ, Fen Bilimleri Enstitüsü, 1996
- [3] Bozşahin, C., Zeyrek, D., Pembeci, İ., “Computer-aided Learning of Turkish Morphology”, Proceedings of Joint Conf. Of ACH and ALLC, Debrecen, Hungary, 1998
- [4] Çakıroğlu, Ü., Doğal Dil İşlemede Anlamsal Ağlar Yardımıyla Bilgi Modelleme, Yüksek Lisans Tezi, KTÜ Fen Bilimleri Enstitüsü, 2001
- [5] Güngördü, Z. And Oflazer, K., “Parsing Turkish using the Lexical-Functional Grammar Formalizm, Machine Translation” Vol.10, Number 4, 1995
- [6] Cohen, A., Introduction to Computer Theory, John Wiley & Sons Inc., Newyork, 1986.
- [7] Banguoğlu, T., Türkçe’nin Grameri, Türk Tarih Kurumu Basımevi, Ankara, 1990.
- [8] Vagifoğlu, V., “Doğal Dil İşleme”, Yapay Zeka Problemler-Yöntemler-Algoritmalar, Seçkin Yayıncılık, Ankara, 2004
- [9] Hengirmen, M., Türkçe Temel Dilbilgisi, Engin Yayınevi, Ankara, 1990.
- [10] Solak, A., Oflazer, K.: Parsing agglutinative word structures and its application to spelling checking for Turkish. In Proceeding of the 16th International Conference on Computational Linguistic, Vol. 1. (1992) 39 – 46
- [11] Eryigit, G. and Adali, E., An Affix Stripping Morphological Analyzer For Turkish, Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Innsbruck, Austria, 2004.
- [12] Oflazer, K., Two-level Description of Turkish Morphology. Literary and Linguistic Computing, Vol. 9. (1994) 137 – 148