

1-Giriş

Proje Amaçları:

Projenin ana amacı, kullanıcı tarafından seçilen Türkçe metin dosyası üzerinde belirli arama şablonlarını (RegEx), morfolojik analiz yöntemlerini ve kelime köklerini kullanarak işlemler gerçekleştirmektir. Bu süreç, kullanıcıya metin içeriği hakkında detaylı bilgiler sağlamak ve metindeki kelimelerin yapısal özelliklerini incelemek için kullanılır.

Projenin temel hedefleri şunlardır:

- Kullanıcıya, bilgisayarlarından txt uzantılı dosyalar seçme ve yükleme imkanı sağlamak.
- Kullanıcının girdiği RegEx ifadesi ile metin belgesinde arama yapma imkanı sunmak ve sonuçları kullanıcı dostu bir arayüzde göstermek.
- Metindeki hatalı yazımları düzeltme ve noktalama işaretlerini temizleme işlemlerini gerçekleştirerek metni hazırlamak.
- Metindeki benzersiz kelimelerin morfolojik analizini (kök, morfem vs.) yapmak ve kök alma işlemi sonrası eşsiz kelime sayısını karşılaştırmak.

Proje Özellikleri ve Fonksiyonları :

Projede yer alan temel özellikler şunlardır:

Dosya Seçme ve RegEx Arama: Projede kullanıcı, bilgisayarlarından bir metin dosyası seçebilir. Seçilen dosyanın adı ana form ekranında görüntülenir. Kullanıcı, bir RegEx ifadesi yazarak "Ara" butonuna tıklayabilir. Bu işlem sonucunda, metin dosyasında RegEx ifadesiyle eşleşen tüm sonuçlar yeni bir form ekranında gösterilir.

Morfolojik Analiz: Proje, metindeki hatalı yazımları düzeltme ve noktalama işaretlerini silme işlemlerini gerçekleştirir. Ardından, metindeki benzersiz (unique) kelimelerin morfolojik analizini yapar. Bu analizde, her kelimenin kökü ve morfemleri ayrıntılı bir şekilde incelenir. Ayrıca, kök alma işlemi gerçekleştirilerek, kök alma öncesi ve sonrası eşsiz kelime sayıları hesaplanarak kullanıcıya sunulur.

Kullanılan Teknolojiler ve Dil

Bu proje C# diliyle Visual Studio ortamında geliştirilmiştir. Metin işleme işlemleri için RegEx ifadeleri kullanılmış olup, hatalı yazımları düzeltme ve noktalama işaretlerini silme işlemleri

gerçekleştirilmiştir. Morfolojik analiz için uygun dil ve kütüphaneler kullanılarak kelime köklerinin ve morfemlerinin ayrıntılı analizi yapılmıştır.

2 -Proje Dosyaları

2.1 Ana Form Ekranı

Ana form ekranı kullanıcıya metin dosyası seçme, RegEx ile arama yapma ve morfolojik analiz gerçekleştirme imkanı sunar. Kullanıcı "Dosya Ekle" butonuna tıklayarak bilgisayarından bir .txt uzantılı dosya seçebilir. Bu butona basıldığında dosyaEkleBtn_Click fonksiyonu çalışır. Seçilen dosyanın adı form ekranında label(DosyaAdiLbl)'da yazılır, böylece kullanıcı seçtiği dosyayı doğrulayabilir. Formda bir textbox bulunur ve kullanıcı buraya aramak istediği RegEx ifadesini yazabilir. "Ara" butonuna tıkladığında rgxAraBtn_Click fonksiyonu çalışır, metin dosyasında RegEx ifadesiyle eşleşen sonuçlar yeni bir form ekranında gösterilir. Her arama sonrası, sonuç formundaki değerler temizlenir, böylece kullanıcı her yeni arama için taze ve doğru sonuçlar elde eder. Ayrıca, formda "Morfoloji Analiz Getir" butonu bulunur. Bu butona tıklandığında morfolojikAnalizBtn_Click fonksiyonu çalışacaktır. İlk olarak metindeki hatalı yazımlar düzeltilir ve noktalama işaretleri silinir. Ardından, metindeki tüm benzersiz kelimelerin morfolojik analizi yapılır. Bu analizde, her kelimenin kökü,

morfemleri gibi yapısal özellikleri ayrıntılı bir şekilde incelenir. Bu işlemlerin ardından, kök alma öncesi ve sonrası eşsiz kelime sayıları hesaplanarak kullanıcıya sunulur.

Ana Form Ekranı

dosyaEkleBtnClick() Fonksiyonu :

Bu metod, kullanıcıya bir dosya seçme penceresi açar (OpenFileDialog sınıfı kullanılarak) ve sadece .txt uzantılı dosyaların seçilebilmesini sağlar (Filter özelliği ile). Eğer daha önce bir dosya seçildiyse, bu dosyanın bulunduğu dizini varsayılan olarak açar (InitialDirectory özelliği ile). Kullanıcı bir dosya seçerse, seçilen dosyanın yolunu (dosyaYolu değişkeni) saklar ve dosya adını bir Label ögesine yazdırır (dosyaAdiLbl). Bu kod parçası, bir butona tıklandığında çalışacak olan ve kullanıcının bir metin dosyası seçmesine olanak tanıyan temel bir dosya seçme işlemini gerçekleştirir.

Sözde Kod:

- 1-openFileDialog oluştur
2. openFileDialog başlığını "Metin dosyası seçiniz.." olarak ata
3. openFileDialog nesnesine ".txt" filtresi uygula
4. nesneyi göster
5. Nesne üzerinde seçili dosyanın varlığını kontrol et, doğruysa git 6. adıma değilse 2.adıma git
6. DosyaAdiLbl'da dosya adına eşitle
- 7.Seçilen Dosya Yolu Belirlenir
- 8.Bitir.

```
private void dosyaEkleBtn_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Metin Dosyaları|*.txt";

    if (!string.IsNullOrEmpty(dosyaYolu) && File.Exists(dosyaYolu))
    {
        openFileDialog.InitialDirectory = Path.GetDirectoryName(dosyaYolu);
    }

    dosyaYolu = openFileDialog.ShowDialog() == DialogResult.OK ?
openFileDialog.FileName : dosyaYolu;
    dosyaAdiLbl.Text = openFileDialog.ShowDialog() == DialogResult.OK ? "Seçilen
Dosya: " + openFileDialog.SafeFileName : dosyaAdiLbl.Text;
```

}

Dosya Ekleme Fonksiyonu kodu

Açıklama:

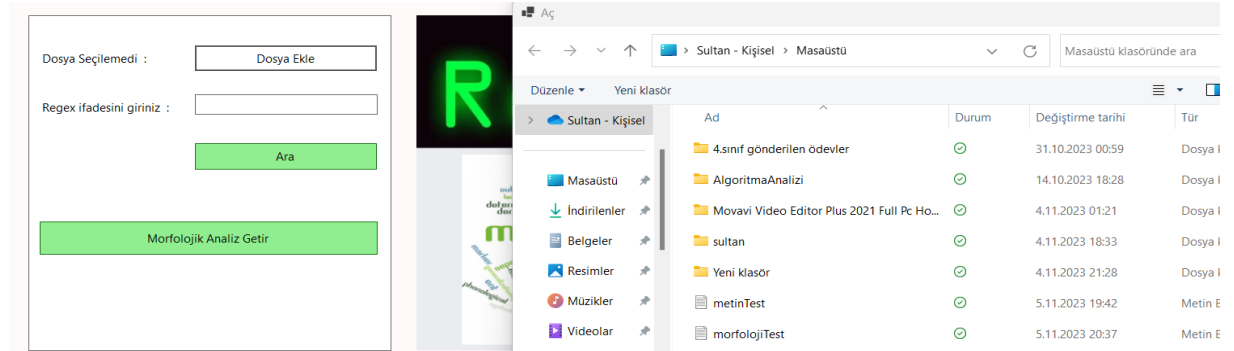
Bir OpenFileDialog nesnesi oluşturulur ve metin dosyalarını (.txt uzantılı dosyalar) seçmeyi sağlayacak bir filtreye sahip olacak şekilde yapılandırılır.

Eğer dosyaYolu adında bir değişken dolu ise ve bu dosya hala mevcutsa, openFileDialog nesnesinin başlangıç dizini bu dosyanın bulunduğu dizin olarak ayarlanır.

Kullanıcıya bir dosya seçme penceresi gösterilir. Eğer kullanıcı bir dosya seçer ve "OK" düğmesine tıklarsa, seçilen dosyanın yolu dosyaYolu değişkenine atanır; aksi takdirde dosyaYolu değişkenin mevcut değeri korunur.

Aynı şekilde, kullanıcı "OK" düğmesine tıklarsa, bir etiket (label) olan dosyaAdiLbl'nin metni "Seçilen Dosya: " + seçilen dosyanın adı ile güncellenir. Eğer kullanıcı dosya seçmez ya da dosya seçme işlemi iptal edilirse, etiketin metni değiştirilmez.

Bu kod parçası, kullanıcının bir metin dosyasını seçmesini ve seçilen dosyanın yolunu ve adını görüntülemesini sağlar.



Dosya Ekle Butonu



Seçilen dosyanın adını görüntüleme

rgxAraBtnClick() Fonksiyonu :

Kullanıcının bir metin dosyasını seçmesini ve seçilen dosya üzerinde bir RegEx ifadesi kullanarak metin içinde eşleşen desenleri bulmasını sağlamaktır.

Sözde Kod :

1. Başla
2. Dosya Yolu ve Varlığı Kontrol Et: Eğer dosya yolu boş değilse ve dosya mevcutsa:
- 3-Dosya İçeriğini Oku
4. RegEx İfadesini Al ve Eşleşmeleri Bul:
Eğer RegEx ifadesi boş değilse:
- 5- Eşleşmeleri Saklamak İçin Boş Bir MatchCollection Oluşturulur: eslesmeler
- 6-Dosya İçeriğinde RegEx İfadesine Göre Eşleşmeler Aranır ve eslesmeler Koleksiyonuna Eklenir. Eğer Eşleşmeler Bulunursa:
- 7- Her Bir Eşleşmeyi eslesmeler Koleksiyonundan Al ve sonuclarForm'un SonuclarListBox'ına Eklenir
- 8- SonuclarForm Formu Gösterilir.Eğer Eşleşmeler Bulunmazsa:"Eşleşme Bulunamadı." Mesajı Gösterilir.Eğer RegEx İfadesi Boşsa:"Lütfen bir RegEx ifadesi girin." Uyarısı Gösterilir
- 9-RegEx İfadesini Al ve Eşleşmeleri Bulma İşlemi Tamamlanır.Eğer Dosya Yolu Boşsa veya Dosya Mevcut Değilse:"Lütfen geçerli bir dosya seçin." Uyarısı Gösterilir
- 10- Dosya Yolu ve Varlığı Kontrol Etme İşlemi Tamamlanır
11. Sonucu yazdır ve bitir.

```
private void rgxAraBtn_Click(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(dosyaYolu) && File.Exists(dosyaYolu))
    {
        string dosyaIcerigi = File.ReadAllText(dosyaYolu);

        string regexIfadesi = rgxAraTxtBx.Text;
        if (!string.IsNullOrEmpty(regexIfadesi))
        {
            MatchCollection eslesmeler = Regex.Matches(dosyaIcerigi,
            regexIfadesi);

            RegexSonucFrm sonuclarForm = eslesmeler.Count > 0
                ? new RegexSonucFrm()
                : null;
        }
    }
}
```

```

        sonuclarForm?.SonuclarListBox.Items.AddRange(eslesmeler
            .Cast<Match>()
            .Select(eslesme => eslesme.Value)
            .ToArray());

        sonuclarForm?.Show();
    }
    else
    {
        MessageBox.Show("Lütfen bir RegEx ifadesi girin.");
    }
}
else
{
    MessageBox.Show("Lütfen geçerli bir dosya seçin.");
}
RgxAraBtn Fonksiyonu Kodu

```

Açıklama:

Kullanıcı, bir dosya seçmeksizin "Ara" butonuna tıkladığında veya geçerli bir dosya seçilmediğinde bir uyarı mesajı gösterilir. Eğer kullanıcı bir dosya seçtiyse ve seçilen dosya mevcutsa, dosyanın içeriği okunur ve dosyalcerigi adlı bir metin değişkenine atanır.

Ardından, kullanıcının bir RegEx ifadesi girmesini bekler. Eğer kullanıcı boş bir ifade girmişse veya hiçbir ifade girmemişse, bir hata mesajı gösterilir ve kullanıcıya bir RegEx ifadesi girmesi gerektiği bildirilir.

Eğer kullanıcı geçerli bir RegEx ifadesi girmişse, bu ifade kullanılarak dosyalcerigi içinde eşleşen desenler aranır. Bu eşleşmeler eslesmeler adlı bir koleksiyon içinde saklanır.

Son olarak, eşleşen desenler yeni bir form olan RegexSonucFrm'de gösterilir. Her bir eşleşme, bu formun bir liste kutusuna (ListBox) eklenir ve kullanıcıya sonuçlar yeni bir pencerede görüntülenir.

2.2 Regex Sonuc Form Ekranı

Burada regex sonucu ekranda gösterilir.

```

public partial class RegexSonucFrm : Form
{
    public RegexSonucFrm()
    {
        InitializeComponent();
    }
    public ListBox SonuclarListBox
    {
        get { return sncListBox; }
    }
}

```

Regex ifadesi giriř ekranı

Açıklama:

RegexSonucFrm sınıfının yapıcı metodu (public RegexSonucFrm()) oluşturulmuřtur. Bu metot, sınıf bir örneęi oluşturulduęunda (nesne oluşturulduęunda) çalıştırılır ve formun başlatılması için gerekli işlemleri gerçekleştirir. InitializeComponent() metodu çağırılır.

SonuclarListBox adında bir özellik (property) tanımlanmıřtır. Bu sayede snclListBox adında bir ListBox öęesine erişimi sağlar. Dıřarıdan bu formun içindeki snclListBox öęesine erişim sağlamak için kullanılabilir.

Regex Sonuc Form Ekranı

morfolojikAnalizBtn_Click() Fonksiyonu :

burada amaç, seçilen dosyanın morfolojik analizinin gerçekleştirilmesidir. Morfolojik analiz işlemi sırasında metinde yer alan kelimeler üzerinde doğrulama işlemleri gerçekleştirilir. Bu işlemler arasında yazım hatalarının düzeltilmesi, metindeki noktalama işaretlerinin kaldırılması, kelimelerin çekimi ve benzersiz kelimelerin belirlenmesi bulunmaktadır. Analiz sonucunda elde edilen kelimeler liste halinde gösterilir ve her kelimenin kökü ve morfemleri ayrıntılı bir şekilde sunulur. Bu süreç, metindeki kelimelerin dilbilgisel yapılarını anlama ve analiz etme amacını taşır, ve bu işlemler, metnin dilbilgisel yapısını daha derinlemesine incelemek için kullanılır. Bu şekilde, kullanıcı metnin içeriğinde yer alan kelimelerin köklerini ve morfem yapılarını daha ayrıntılı bir şekilde keşfetme imkanına sahip olur.

Sözde Kod:

1-MorfolojiAnalizGetir Butonuna bas

Eğer dosya seçilmemişse:

Hata mesajı göster: "Lütfen geçerli bir dosya seçin."

Eğer dosya seçilmişse adım 2 ye git.

2-Dosyadaki metni oku ve metinDeğişkeni'ne ata

3-Metindeki hatalı yazımları düzelt (spelling correction) ,Metindeki noktalama işaretlerini sil.

4-Tüm kelimeleri ayır ve uniqueKelimeler listesine ekle

5-Eşsiz kelime sayısını hesapla ve eskiEşsizKelimeSayisi'na ata

6-Yeni bir form oluştur (MorfolojikAnalizSonucForm)

7-Formda uniqueKelimeler listesini kullanarak morfolojik analiz yap:

Her bir kelime için:

Kök alma işlemi uygula (stemming veya lemmatization)

Kök alma sonrası kelimeyi morfem analizi yaparak ayrıştır

8-Analiz sonuçlarını formda göster.

9-Bitir.

```
private void morfolojikAnalizBtn_Click_1(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(dosyaYolu) && File.Exists(dosyaYolu))
    {
        string dosyaIcerigi = File.ReadAllText(dosyaYolu);
        string metinWithoutPunctuation = new string(dosyaIcerigi.Where(c =>
!char.IsPunctuation(c)).ToArray());
```

```

        MessageBox.Show("Noktalama İşaretleri Silinmiş Metin: " +
metinWithoutPunctuation);

        File.WriteAllText(dosyaYolu, metinWithoutPunctuation);
        MessageBox.Show("Noktalama işaretleri silme işlemi tamamlandı ve dosya
güncellendi.");
        // Morfolojik analiz sonuçlarını yeni formda göster
        MorfolojikAnalizFrm morfolojiSonucForm = new MorfolojikAnalizFrm();
        MorfolojikAnalizYap(dosyaIcerigi, morfolojiSonucForm);
        morfolojiSonucForm.Show();
    }
    else
    {
        MessageBox.Show("Lütfen geçerli bir dosya seçin.");
    }
}

```

Açıklaması:

İlk olarak, dosya yolunun boş olup olmadığını ve seçilen dosyanın var olup olmadığını kontrol eder. Eğer dosya yolunda bir değer varsa ve seçilen dosya mevcutsa, aşağıdaki işlemler gerçekleştirilir. Aksi halde bir hata mesajı görüntülenir.

Dosya içeriği (dosyaIcerigi) dosyadan okunur ve metin üzerinde değişiklik yapılacak bir değişkene atanır. Metindeki noktalama işaretleri, kaldırılır ve işaretsiz metin (metinWithoutPunctuation) oluşturulur. Bu işlem, metin içeriğinin noktalama işaretlerinden arındırılmasını sağlar. Noktalama işaretleri kaldırılmış metin, dosyanın üzerine yazılır, böylece dosya güncellenmiş olur. Kullanıcıya bir bilgi penceresi gösterilir, bu pencerede noktalama işaretleri kaldırılma işleminin tamamlandığı bildirilir. Ardından, morfolojik analiz sonuçlarını göstermek için yeni bir form olan MorfolojikAnalizFrm oluşturulur ve bu form MorfolojikAnalizYap fonksiyonu aracılığıyla morfolojik analiz işlemlerine tabi tutulur. MorfolojikAnalizFrm formu, kullanıcıya gösterilir. Bu kod, seçilen dosyanın metin içeriğinden noktalama işaretlerini kaldırma işlemi ve ardından morfolojik analiz sonuçlarını gösterme işlemi gerçekleştirir.

Kullanıcının metin verisinin dilbilgisel yapısını incelemesi ve analiz etmesine olanak tanır.

The screenshot shows a Windows application window titled 'Form1'. On the left is a main form with the following elements: a label 'Seçilen Dosya: morfolojiTest.txt' next to a 'Dosya Ekle' button; a label 'Regex ifadesini giriniz :' next to an empty text input field; a green 'Ara' button; and a large green 'Morfolojik Analiz Getir' button. On the right is a smaller window titled 'RegEx' with a black background and the text 'RegEx' in large green letters. Below this, there is a word cloud and a text area containing a paragraph of Turkish text with punctuation removed. At the bottom right of this window is a 'Tamam' button.

Seçilen Dosya: morfolojiTest.txt Dosya Ekle

Regex ifadesini giriniz :

Ara

Morfolojik Analiz Getir

RegEx

Noktalama İşaretleri Silinmiş Metin: Bahçede harika bir gün geçirdik Kuşlar melodilerini söyledi çiçekler renk cümbüşü içinde açtı Doğanın güzellikleri karşısında büyüledik Akşam geldiğinde gökyüzü turuncudan mora döndü ve yıldızlar parlamaya başladı Bu anı ölümsüzleştirmek istedik ve birlikte fotoğraf çektirdik Doğanın mucizeleri karşısında hayran kaldık

Tamam

Metindeki noktalamaları sildiğine dair uyarı mesajı

This screenshot shows the same 'Form1' window as before, but with a small dialog box overlaid in the center. The dialog box has a title bar with a close button and contains the text 'Metin düzenleme işlemi tamamlandı ve dosya güncellendi.' At the bottom right of the dialog box is a 'Tamam' button.

Seçilen Dosya: morfolojiTest.txt Dosya Ekle

Regex ifadesini giriniz :

Ara

Morfolojik Analiz Getir

RegEx

Metin düzenleme işlemi tamamlandı ve dosya güncellendi.

Tamam

Uyarı mesajı ekranı

MorfolojikAnalizYap() Fonksiyonu:

Zemberek adlı Türkçe doğal dil işleme (NLP) kütüphanesini kullanarak metin verisi üzerinde morfolojik analiz işlemlerini uygular.temel amacı, verilen metin içeriğini analiz ederek içerisindeki kelimelerin köklerini belirlemek ve dilbilgisel özelliklerini tespit etmektir. Ayrıca, bu işlem sonucunda elde edilen verileri kullanarak, metindeki benzersiz (unique) kelime sayılarını karşılaştırmak ve metindeki hatalı yazımları düzeltmek yer almaktadır.

Sözde Kod:

- 1- Zemberek ile Türkçe morfoloji analizi nesnesi oluştur
- 2- öncekiEsisizKelimeSayisi = Bir küme oluştur ve metindeki benzersiz kelimelerin sayısını hesapla
- 3- tokenizedMetin = Metin içeriğini boşluklar, satır sonları ve sekme karakterleri gibi ayırıcılarla parçalara ayır
- 4- sonraEsisizKelimeler = Bir küme oluştur // Sonraki eşsiz kelimeleri saklamak için
- 5- Her token için tokenizedMetin üzerinde döngü yap
- 6- Kelimenin morfolojik analizini yap
- 7- Kök belirleme işlemi
- 8- pos = Analizler içinden dilbilgisel özellikleri tespit et
- 9- Analiz sonuçlarını döndür.
- 10-Bitir

```
private void MorfolojikAnalizYap(string metin, MorfolojikAnalizFrm
morfolojikAnalizFrm)
{
    var morphology = TurkishMorphology.CreateWithDefaults();
    var öncekiEsisizKelimeSayisi = new HashSet<string>(metin.Split(new[] { ' ',
'\r', '\n', '\t' }, StringSplitOptions.RemoveEmptyEntries)).Count;
    var tokenizedMetin = metin.Split(new[] { ' ', '\r', '\n', '\t' },
StringSplitOptions.RemoveEmptyEntries);
    var sonraEsisizKelimeler = new HashSet<string>();

    foreach (var token in tokenizedMetin)
    {
        var analyses = morphology.Analyze(token);
        var enYayginKok = analyses
            .Select(analysis => analysis.GetLemmas()[0])
            .GroupBy(kok => kok)
            .OrderByDescending(grp => grp.Count())
            .Select(grp => grp.Key)
            .FirstOrDefault();

        var pos = analyses.FirstOrDefault()?.GetPos();

        if (enYayginKok != null && pos != null)
        {
```

```

        morfolojikAnalizFrm.MorfolojikAnalizSonuclariniGoster("Kelime: " +
token + ", Kök: " + enYayginKok + ", POS: " + pos);
        sonraEsisizKelimeler.Add(enYayginKok);
    }
}
var sonraEsisizKelimeSayisi = sonraEsisizKelimeler.Count;
if (morfolojikAnalizFrm != null)
{
    morfolojikAnalizFrm.SetEsisizKelimeSayilari(oncekiEsisizKelimeSayisi,
sonraEsisizKelimeSayisi);
}}
Morfolojik Anliz fonksiyonu kodu


```

Seçilen Dosya: morfolojiTest.txt
Dosya Ekle

Regex ifadesini giriniz :

Ara

Morfolojik Analiz Getir



Kelime: gün, Kök: gün, POS: Noun
Kelime: Kuşlar, Kök: kuş, POS: Noun
Kelime: melodilerini, Kök: melodi, POS: Noun
Kelime: çiçekler, Kök: çiçek, POS: Verb
Kelime: renk, Kök: renk, POS: Noun
Kelime: cümbüşü, Kök: cümbüş, POS: Noun
Kelime: içinde, Kök: iç, POS: Adv
Kelime: Doğanın, Kök: doğa, POS: Noun
Kelime: güzellikleri, Kök: güzel, POS: Noun
Kelime: karşısında, Kök: karşı, POS: Noun
Kelime: Akşam, Kök: akşam, POS: Noun
Kelime: gökyüzü, Kök: gökyüzü, POS: Noun

Kök Öncesi Eşsiz Kelime Sayısı: 37

Kök Sonrası Eşsiz Kelime Sayısı: 30

Açıklaması:

TurkishMorphology.CreateWithDefaults() ile varsayılan ayarlarla bir Türkçe morfoloji analiz nesnesi (morphology) oluşturulur.

Metin içeriği boşluklar, satır sonları ve sekme karakterleri gibi ayırıcılar kullanılarak parçalara ayrılır ve bu parçalar tokenizedMetin dizisine atanır.

tokenizedMetin üzerinde döngü ile her kelimenin morfolojik analizi yapılır.

Her kelimenin analizi sonucunda, kelimenin en yaygın kökü ve dilbilgisel özellikleri tespit edilir.

Eğer kelimenin kökü ve dilbilgisel özellikleri geçerliyse, bu bilgiler bir sonuç formunda gösterilir ve kelimenin kökü, benzersiz kelime listesine eklenir.

Metindeki benzersiz kelime sayıları hesaplanır ve önceki ve sonraki benzersiz kelime sayıları karşılaştırılarak hatalı yazımlar düzeltilir.

Son olarak, işlem sonucu elde edilen veriler MorfolojikAnalizFrm adlı forma iletilir.

2.3 Morfolojik Analiz Form Ekranı

Morfolojik Analiz formunda ,Morfolojik analiz sonuçlarını göstermek ve kök alma işlemi öncesi ve sonrası eşsiz kelime sayılarını görüntülemek için gerekli label ve listbox oluşturulmuştur.

```
public MorfolojikAnalizFrm()
{
    InitializeComponent();

}
public void MorfolojikAnalizSonuclariniGoster(string analizSonuclari)
{
    listBox1.Items.Add(analizSonuclari);
}
public void SetEsisizKelimeSayilari(int oncekiEsisizKelimeSayisi, int
sonraEsisizKelimeSayisi)
{
    oncekiKelimeSayisiLbl.Text = "Kök Öncesi Eşsiz Kelime Sayısı: " +
oncekiEsisizKelimeSayisi.ToString();
    sonrakiKelimeSayisiLbl.Text = "Kök Sonrası Eşsiz Kelime Sayısı: " +
sonraEsisizKelimeSayisi.ToString();
}
```

}

Açıklaması:

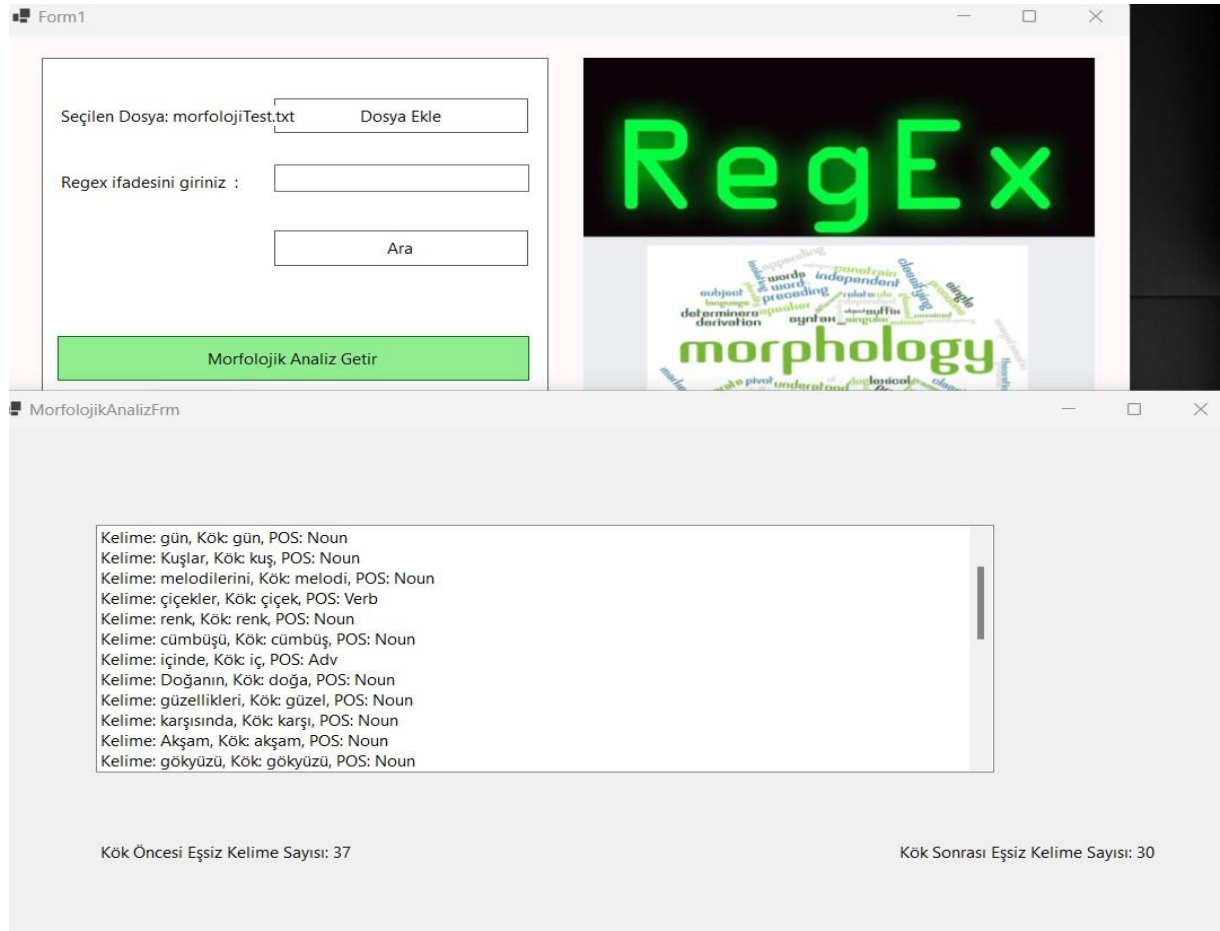
Constructor (MorfolojikAnalizFrm()): Bu metod, MorfolojikAnalizFrm sınıfının bir nesnesi oluşturulduğunda çalışır. Yapıcı metod olarak da bilinir ve formun bileşenlerini başlatır, yani formun tasarımını ve arayüz elemanlarını yükler.

MorfolojikAnalizSonuclariniGoster(string analizSonuclari): Bu metod, parametre olarak aldığı analizSonuclari string değerini, listBox1 adlı ListBox bileşenine ekler. Yani, bu metod morfolojik analiz sonuçlarını kullanıcıya göstermek için ListBox içerisine eklemeyi sağlar.

SetEsisizKelimeSayilari(int oncekiEsisizKelimeSayisi, int sonraEsisizKelimeSayisi): Bu metod, iki tane integer parametre alır: oncekiEsisizKelimeSayisi ve sonraEsisizKelimeSayisi. Bu parametreler, kök öncesi ve kök sonrası eşsiz kelime sayılarını temsil eder. Bu metod, formun arayüzünde bulunan oncekiKelimeSayisiLbl ve sonrakiKelimeSayisiLbl adlı Label bileşenlerine,

parametre olarak aldığı değerleri metin olarak yerleştirir. Yani, bu metod kök öncesi ve sonrası eşsiz kelime sayılarını formun arayüzünde göstermeye yarar.

Bu metodlar, MorfolojikAnalizFrm sınıfının davranışını ve arayüzünü kontrol etmek için kullanılır.



Form1

Seçilen Dosya: morfolojiTest.txt Dosya Ekle

Regex ifadesini giriniz :

Ara

Morfolojik Analiz Getir

MorfolojikAnalizFrm

Kelime: gün, Kök: gün, POS: Noun
Kelime: Kuşlar, Kök: kuş, POS: Noun
Kelime: melodilerini, Kök: melodi, POS: Noun
Kelime: çiçekler, Kök: çiçek, POS: Verb
Kelime: renk, Kök: renk, POS: Noun
Kelime: cümbüşü, Kök: cümbüş, POS: Noun
Kelime: içinde, Kök: iç, POS: Adv
Kelime: Doğanın, Kök: doğa, POS: Noun
Kelime: güzellikleri, Kök: güzel, POS: Noun
Kelime: karşısında, Kök: karşı, POS: Noun
Kelime: Akşam, Kök: akşam, POS: Noun
Kelime: gökyüzü, Kök: gökyüzü, POS: Noun

Kök Öncesi Eşsiz Kelime Sayısı: 37

Kök Sonrası Eşsiz Kelime Sayısı: 30

Morfolojik Analiz Getir Buton sonuç ekranı