

**BULANIK MANTIK VE YAPAY SİNİR AĞLARI İLE
TÜRKÇE YAZIM DENETLEYİCİSİ**

**YÜKSEK LİSANS TEZİ
Müh. Simla DİLSİZ
504021546**

**Tezin Enstitüye Verildiği Tarih : 9 Mayıs 2005
Tezin Savunulduğu Tarih : 30 Mayıs 2005**

**Tez Danışmanı : Prof. Dr. Eşref ADALI
Diğer Jüri Üyeleri Doç. Dr. Sabih ATADAN (İ.T.Ü)
Prof. Dr. Oya KALIPSIZ (Y.T.Ü.)**

MAYIS 2005

ÖNSÖZ

Tez çalışmam sırasında benden desteğini hiçbir zaman esirgemeyen, bu çalışmada bana büyük bir altyapı ve bilgi birikimi sağlayan Doğal Dil İşleme grubunun kurulmasına önderlik eden, değerli hocam ve danışmanım Sayın Prof. Dr. Eşref ADALI'ya teşekkürlerimi sunarım.

Tecrübelerini benimle paylaşan, yüksek lisans eğitimim ve tez çalışmam sırasında yardımlarını ve değerli vakitlerini esirgemeyen Gülşen Eryiğit ve Cüneyd Tantığ'a, birlikte Doğal Dil İşleme alanında çeşitli çalışmalar ve bilgi paylaşımında bulunduğum Doğal Dil İşleme grubu hocalarıma ve arkadaşlarıma teşekkür ederim.

Tüm yaşantım boyunca her zaman sevinç ve üzüntülerimi benimle paylaşan, sonsuz destekleri ile bana güç veren sevgili aileme ve kardeşim Rauf'a herşey için çok teşekkür ederim.

Lisans, Yüksek Lisans ve mesleki hayatım boyunca hep yanımda olan, varlığıyla bana her zaman güven ve azim veren, birlikte yaptığımız her çalışmadan büyük zevk duyduğum ve tecrübe edindiğim sevgili arkadaşım Özlem Sema Eroğlu'na, yaşamım boyunca bana destek, yol gösterici ve can yoldaşı olan sevgili Ece Gündoğdu'ya teşekkürü bir borç bilirim.

Ve sevgili eşim Mehmet Erkul'a, her konuda olduğu gibi yüksek lisansa girişim ve tez çalışmam sırasında beni yüreklendirdiği, varlığıyla bana güven ve destek kaynağı olduğu, yoğun geçen tez çalışması sırasında hayatı bana kolaylaştırdığı için çok teşekkür ederim.

Mayıs, 2005

Simla Dilsiz

Sevgili Eşime

İÇİNDEKİLER

ÖNSÖZ.....	i
İÇİNDEKİLER	iv
KISALTMALAR	vi
TABLO LİSTESİ.....	vii
ŞEKİL LİSTESİ.....	viii
SEMBOL LİSTESİ	ix
ÖZET.....	x
SUMMARY	xi
1 . BÖLÜM GİRİŞ	1
2 . BÖLÜM DOĞAL DİL İŞLEME	4
2.1 Yapay Zeka	4
2.2 Doğal Dil İşlemeye Genel Bakış.....	4
2.3 Diller	5
2.4 Türkçe	7
3 . BÖLÜM BULANIK MANTIK VE YAPAY SİNİR AĞLARI İLE TÜRKÇE YAZIM DENETLEYİCİSİ TASARIMI ve GERÇEKLEŞTİRİMİ	8
3.1 Yazım Denetleyicilere Genel Bakış	8
3.2 Türkçe İçin Yazım Denetleyici	8
3.3 İstatistiksel Yöntemler	9
3.4 Bulanık Mantık	9
3.5 Yapay Sinir Ağları	10
3.5 Bulanık Mantık Ve Yapay Sinir Ağları İle Türkçe Yazım Denetleyicisi	12
3.5.1 Genel Yapı	13
3.5.2 Sözcüklerin Çözümlemesi.....	17
3.5.3 Bulanık Mantık İle İki Sözcüğün Karşılaştırılması.....	23
3.5.4 Öğrenme Aşaması	33
3.5.4.1 Öğrenme Öncesi Hazırlık.....	34
3.5.4.2 Öğrenme.....	38
3.5.4.3 Öğrenme Puanının Hesaplanması	41
3.5.5 Bulanık Mantık ve YSA ile Genel Benzerlik Puanlamasının Yapılması..	44
3.5.6 Kök ve Eklerin Karşılaştırılması, Bütünleşik Puanların Elde Edilmesi,	
Sonuç Önermelerin Çıkarımı	45
3.5.7 Deney Sonuçları	51

4 . BÖLÜM YAZILIMIN AÇIKLANMASI.....	57
4.1 Veri Yapıları.....	57
4.2 Yöntemler.....	60
4.3 Kullanılan Ortam.....	70
4.4 Uygulamanın Çalışması	70
5 . BÖLÜM KAYNAK TARAMASI	83
6 . BÖLÜM SONUÇ VE ÖNERİLER.....	92
KAYNAKLAR	94
EK-A.....	97
ÖZGEÇMİŞ.....	99

KISALTMALAR

YSA	: Yapay Sinir Ağları
KKK	: Kök Kaynak Kümesi
EKK	: Ek Kaynak Kümesi
BÇ	: Biçimbirimsel Çözümleyici
BMP	: Bulanık Mantık Puanı

TABLO LİSTESİ

Tablo 3.1: EKK’de “bilgilerimizin” sözcüğünün ekleri görünümü	18
Tablo 3.2: Kök Kaynak Kümesi görünümü	19
Tablo 3.3: Ek Kaynak Kümesi görünümü	19
Tablo 3.4: “albert einstein” ifadesinin aranması sırasında karşılaşılmış yazımlar....	24
Tablo 3.5: Bulanık Mantık Puanlaması örnekleri	31
Tablo 3.6: Düz BMP ve Ters BMP örnekleri	32
Tablo 3.7: Türkçe’de yazımı birbirine benzeyen harfler.....	44
Tablo 3.8: Öğrenmeden önce ve sonra elde edilmiş puanlar	44
Tablo 3.9 : Örnekteki karşılaştırmada ikinci ek için aday ek ikilileri	48
Tablo 3.10: Örnekteki karşılaştırmada birinci ek için aday ek ikilileri.....	49
Tablo 3.11: Morfolojik Analiz (Eryiğit) BÇ’si kullanılarak oluşan kaynak kümeler	51
Tablo 3.12: Morfolojik Analiz (Eryiğit) BÇ’si kullanılarak oluşan KKK	52
Tablo 3.13: Morfolojik Analiz (Eryiğit) BÇ’si kullanılarak oluşan EKK	52
Tablo 3.14: Zemberek BÇ’si kullanılarak oluşan kaynak kümeler.....	53
Tablo 3.15: Zemberek BÇ’si kullanılarak oluşan KKK ve EKK.....	53
Tablo 3.16: Morfolojik Analiz (Eryiğit) BÇ’si kullanılarak oluşan kaynak kümeler	55
Tablo 3.17: Zemberek BÇ’si kullanılarak oluşan kaynak kümeler.....	55
Tablo 3.18: BÇ (Of lazer) uygulaması kullanılarak oluşan kaynak kümeler.....	55
Tablo 3.19: Deney Sonuçları Özeti	56
Tablo A.1: Q Klavye Harf Koordinatları	97
Tablo A.2: F Klavye Harf Koordinatları	98

ŞEKİL LİSTESİ

Şekil 3.1: İnsan Sinir Hücresi	10
Şekil 3.2: İnsan sinirinin modellenmesi	11
Şekil 3.3: Eğitim işleminin genel yapısı	14
Şekil 3.4: Denetim işleminin genel yapısı	15
Şekil 3.5: Morfolojik Analiz (Eryiğit) uygulaması görünümü	21
Şekil 3.6: BÇ (Ofłazer) uygulamasının sonuç dosyası görünümü.....	23
Şekil 3.7: Harf ve Yer sınıfları örneđi	25
Şekil 3.8: Örnekte ilk sözcük için oluşmuş Harf ve Yer sınıfları.....	26
Şekil 3.9: Örnekte ikinci sözcük için oluşmuş Harf ve Yer sınıfları	27
Şekil 3.10: Örnek için yeri tutmayan harf olası durumları	28
Şekil 3.11: Örnek Yeri Tutmayanlar Puan Listesi görünümü	29
Şekil 3.12: Öğrenme sırasından tutulan yapının görünümü	35
Şekil 3.13: Örnekte oluşan Öğrenme yapısı görünümü.....	36
Şekil 3.14: Örnekte oluşan Öğrenme yapısı görünümü.....	37
Şekil 3.15: Öğrenme yapısının genel görünümü	38
Şekil 3.16: YSA Puan yapısının genel yapı içerisindeki yeri.....	42
Şekil 3.17: Q Klavye Görünümü	43
Şekil 3.18: F Klavye Görünümü.....	43
Şekil 4.1: Kök tablosu veri yapısı.....	57
Şekil 4.2: Ek tablosu veri yapısı	58
Şekil 4.3: Uygulamada kullanılan yöntem çeşitleri	60
Şekil 4.4: Uygulamanın Eğitim sekmesi görünümü.....	71
Şekil 4.5: Biçimbirimsel Çözümleyici grubu	71
Şekil 4.6: Eğitim İşlemleri grubu.....	72
Şekil 4.7: Kaynak Kümeleri grubu	73
Şekil 4.8: Kaynak Kümeleri Listesi.....	73
Şekil 4.9: Uygulamanın Denetim sekmesi görünümü	74
Şekil 4.10: Denetim sekmesi Klavye grubu	74
Şekil 4.11: Denetim sekmesi Denetim İşlemleri grubu	74
Şekil 4.12: Detaylı Denetim Sonucu Listesi.....	76
Şekil 4.13: Denetim Sonucu Listesi	76
Şekil 4.14: Uygulamanın İşlemler Sekmesi	77
Şekil 4.15: İşlemler sekmesi Bulanık Mantık Puanlama grubu.....	78
Şekil 4.16: İşlemler sekmesi Bulanık Mantık Puanlama grubu, tersten denetleme...	78
Şekil 4.17: İşlemler sekmesi Tekli Denetim grubu	79
Şekil 4.18: İşlemler sekmesi Dosya İşlemleri grubu	79
Şekil 4.19: İşlemler sekmesi Türkçe Kurallarının Uygulanması grubu	80
Şekil 4.20: İşlemler sekmesi Öğrenme Puanı grubu.....	81
Şekil 4.21: İşlemler sekmesi Türkçe Kurallarının Uygulanması grubu	81
Şekil 4.22: Uygulama veri dosyaları dizin yapısı	82

SEMBOL LİSTESİ

␣	:boşluk
K_o	:Kullanım Oranı
K_s	:Kullanım Sayısı
BMP	:Bulanık Mantık Puanı
H_s	:Harf Sayısı
TH_s	:Tutan Harf Sayısı
$YTmP$	:Yeri Tutmayan Harf Puanı
U	:Harflerin Yerleri Arası Uzaklık
OgP	:Öğrenme Puanı
KU	:Klavye Uzaklığı
KO	:Kullanım Oranı
GBP	:Genel Benzerlik Puanı

ÖZET

BULANIK MANTIK VE YAPAY SİNİR AĞLARI İLE TÜRKÇE YAZIM DENETLEYİCİSİ

Yapay Zeka insanın düşünme sistemini modellemeyi amaçlayan çalışmalar bütünüdür. Yapay Zekanın bir alt kolu olan Doğal Dil İşleme ise bir doğal dili çözümleme, anlama, yorumlama, üretme gibi işlevlerin bilgisayar ortamında ele alındığı bir mühendislik alanıdır.

Türkçe bağlantılı ve kurallı bir dildir. Köke belli kurallara göre eklenen ekler ile birçok yeni sözcük türetilmektedir. Bu özelliği ile dilde kullanılan sözcük sayısı yönetilemez düzeylere ulaşabilmektedir. Ayrıca dile sonradan eklenmiş kural dışı fakat günlük hayatta kullanımı yoğun olan sözcükler bulunmaktadır.

Bu tezde, Türkçe'nin bağlantılı dil yapısında olmasından yola çıkılıp kök ve ekler temel alınarak bir Türkçe yazım denetleme çalışması yapılmıştır. Sözlük yerine bir eğitim kümesinden elde edilmiş kaynak kümeler kullanılmış, kural tabanlı bir mantık yerine sadece temel birkaç ses uyumu kuralının ele alındığı, asıl yapının istatistiksel temeller üzerine oturtulduğu yeni bir yaklaşım getirilmeye çalışılmıştır. Çalışma iki ana kısımdan oluşmaktadır: Eğitim ve Denetim. Eğitim kısmında kök ve eklerine ayrılmış bir eğitim kümesinden kök ve ekler için iki ayrı kaynak kümesi oluşturulmuştur. Bu kaynak kümeler sözcüklerin kullanımına ait kullanım sıklıklarını da içermektedir. Denetim aşamasında denetimi yapılacak sözcükler de kök ve eklerine ayrılmış, hem kök hem de ekler kendilerine ait kaynak kümeler kullanılarak ayrı ayrı denetlenmiştir. Türkçe'de eklerin sıralanışı önemli olduğu için ekler, art arda gelen ek çiftleri şeklinde ele alınmıştır. Denetleme için iki sözcük (iki kök ya da iki ek çifti) karşılaştırılmış ve derecelendirilmiştir. Dereceleme, bulanık mantık ile geliştirilmiş bir benzerlik oranı, yapılan hataların öğrenilmesi ile elde edilmiş bir öğrenme derecesi ve kullanım sıklıklarından oluşan bütünleşik bir yapıya sahiptir. Her denetleme sonrası, sistem yapılan hataları öğrenmekte, bir sonraki denetimlerde bu öğrenilmiş bilgileri de dereceleme yapısı içinde kullanmaktadır. Öğrenme için Yapay Sinir Ağları kullanılmıştır. Daha sonra denetlenen kök ve ekler Türkçe'nin en temel birkaç ses uyumu kuralı gözetilerek birleştirilip, yorumlanmıştır. Hatalı sözcükler için, denetleme sonuçlarından elde edilen uygun önermeler yapılmıştır.

Anahtar Sözcükler: Doğal Dil İşleme, Yazım Denetleme, Bulanık Mantık, Yapay Sinir Ağları, öğrenme

SUMMARY

TURKISH SPELL CHECKER AND CORRECTION WITH FUZZY LOGIC AND ARTIFICIAL NEURAL NETWORKS

Turkish is an agglutinative and rule-based language. By suffixing morphemes to the roots, a huge variety of new words can be produced. With this property, Turkish language has a very rich, unmanagable vocabulary size. Besides being rule-based, there are lots of exceptional words transferred from other languages and widely used in daily life, violating these rules.

In this study, starting from the point of Turkish's being an agglutinative language, a root and suffix based spell checking and correction is implemented. Instead of a dictionary, a reference set – created from a training set – is used. Because of exceptions and violations in the language, a solely rule-based approach would not help. Thus, a statistical based structure is used with only some basic Turkish consonant and vowel harmony rules. This thesis consists of two main parts: Training and Checking&Correction. In the training part, from a big morphological analyzed train set, root and suffix reference sets are created. While doing this, occurrence rates are also calculated. In the checking and correction part, text to be checked is parsed into its roots and suffixes by a morphological analyzer. These roots and suffixes are checked against the reference sets produced in the training part. The order of suffixing is important in Turkish, thus suffixes are handled as consecutive suffix pairs. For checking and correction, two roots or two suffix pairs are compared and scored. Scoring is a combination of Fuzzy Logic based string match ratio, a learning score and the occurrence ratio. After the correction of a sample text, the system learns the errors made and uses this knowledge in the successive checking and correction operations. Learning is managed with Artificial Neural Networks. Finally, a set of candidate root and suffix corrections are produced, which are then combined with some basic Turkish consonant and vowel harmony rules. The resulting words are proposed.

Keywords: Spell Checking, Spell Correction, Fuzzy Logic, Artificial Neural Networks, Learning

1. BÖLÜM

GİRİŞ

Yapay Zeka, insanın düşünme yapısını modellemek ve bu sayede insan yetkinliklerinin bilgisayar tarafından da uygulanabilirliğini sağlamak amaçlı yapılan çalışmalar bütünüdür. Günümüzde yapay zeka alanında yapılan çalışmaların önemli bir kısmını Doğal Dil İşleme uygulamaları oluşturmaktadır. Yapay Zekanın bir alt kolu olan Doğal Dil İşleme, başta iletişim olmak üzere, insan ya da uygulama (optik okuyucular) kaynaklı hataların giderilmesi, doğal dillerin modellenerek daha iyi bir şekilde kavranması gibi konulardaki çalışmaları içerir. İletişim konuları farklı diller arasında iletişim, insan ile bilgisayarlar arasındaki iletişim gibi konuları kapsamaktadır.

Doğal Dil İşleme konuları ilgilenilen dilin özelliklerine göre özel uygulanış biçimleri içerebilmektedir. Farklı yapılardaki diller farklı modeller oluşturmakta, farklı yaklaşım biçimleri gerektirmektedir. Bu bağlamda Türkçe, bağlantılı bir dil yapısına sahip olması sebebiyle diğer yalın dillere göre özel durumlar içermekte, başarımı için klasik yaklaşımlar dışında uygulamalar gerektirmektedir. Bağlantılı diller, köklere çeşitli sayı ve sırada ekler getirilerek yeni sözcükler türetilbilmesine olanak sağlarlar. Bu özellikleri ile dilde kullanılan sözcük sayısı yönetilemeyecek düzeyde fazladır. Bu yüzden bağlantılı dillerde sözcükleri bütün olarak değil kök ve eklerine göre ele alma yaklaşımı yaygındır. Türkçe'nin kurallı bir dil olması sebebiyle Türkçe için yapılan çalışmalar genel olarak kural tabanlı çalışmalardır. Diğer tüm dillerde olduğu gibi uygulamaların büyük bir çoğunluğu sözlük tabanlıdır. Fakat Türkçe'nin yapısı gereği sözcük sözlükleri yerine kök ve ek sözlükleri kullanımı daha yaygındır.

Bu tez kapsamında Türkçe ile yazılmış metinlerdeki yazım hatalarının bulunup düzeltilmesi hedeflenmektedir. Türkçe'nin bağlantılı bir dil olması sebebiyle sözcükler kök ve ekler olmak üzere ayrı ayrı ele alınmıştır. Genel yaklaşımın aksine sözlük kullanılmadan, büyük eğitim kümeleri ile sistem eğitilmiş, kaynak kök ve ek kümeleri bu eğitim kümesinden elde edilmiştir. Böylece uygulama alanına özgü bir yapı gerçekleştirilmiştir. Türkçe'de kural dışı kullanımların giderek artması sebebiyle

kural tabanlı yaklaşımların bir noktadan sonra yetersiz kaldığı düşünülerek, kuralların temel alınmadan sadece iyileştirici anlamda basit düzeyde kullanıldığı bir uygulama geliştirilmiştir. Amaç sözlük ve kurallardan bağımsız, eğitim ve kullanıma duyarlı bir yapı ortaya çıkarmaktır.

Kök ve eklerin kaynak kümeler temel alınarak düzeltilmesi sırasında kullanılan katar karşılaştırma yöntemi yapay zekanın farklı bir dalı olan Bulanık Mantık yöntemi ile gerçekleştirilmiştir. Bulanık mantık yöntemi ile iki sözcüğün benzerlik oranı bulunmaktadır. Ayrıca yapılan hatalar sistem tarafından öğrenilmekte, sık yapılan hatalara karşı daha duyarlı davranılmaktadır. Öğrenme işlemi yine yapay zekanın bir alt dalı olan Yapay Sinir Ağları ile gerçekleştirilmiştir. Yapay Sinir Ağları öğrenmeyi dolayısıyla insan sinir yapısını modellemeye çalışan, ağırlıklar üzerinden birbirine bağlanmış hücre yapılarından oluşan sistemlerdir. Bu çalışmada, yapılan hatalar ağırlıklandırılmış, daha önce yapılmış hatalara karşı sistem akıllı hale getirilmiştir. Ayrıca karşılaştırma sırasında, kaynak kümelerde yer alan sözcükler kullanım sıklıklarına göre de değerlendirilmiştir. Böylece iki sözcüğün karşılaştırılmasında bulanık mantık, öğrenme ve kullanım sıklığı birleşiminden oluşmuş bir puanlama mantığı geliştirilmiştir.

Köklerden farklı olarak köklerin ele alınışında, bağlantılı dillerin yapısı gereği, sıralanış önem kazanmaktadır. Bu yüzden bu tezde ekler üzerinden yapılan işlemlerde ekler tek başlarına değil, arka arkaya gelmiş ek ikilileri şeklinde el alınmaktadırlar.

Uygulamada kök ve ekleri ayırmak için 3 farklı biçimbirimsel çözümleyici kullanılmıştır. Çözümleyicilerden bir tanesi uygulama içinden bütünleşik olarak kullanılabilir. Biçimbirimsel Çözümleyicinin verimliliği hata denetim işleminin verimliliğini doğrudan etkilemektedir. Bu yüzden farklı çözümleyiciler kullanılarak oluşturulmuş kaynak kümeleri ve denetlenecek metinlerin değişik kombinasyonları denenmiş, sonuçlar yorumlanmıştır.

Tez 6 bölümden oluşmaktadır. Giriş bölümünün ardından gelen “Doğal Dil İşleme” başlıklı ikinci bölümde, öncelikle doğal dil işlemenin içinde bulunduğu Yapay Zeka kavramı tanıtılmış daha sonra Doğal Dil İşleme, dünya dilleri ve sınıflandırılmaları hakkında bilgi verilmiştir. İkinci bölümde son olarak Türkçe ve yapısı anlatılmıştır. “Bulanık Mantık ve Yapay Sinir Ağları ile Türkçe Yazım Denetleyicisi” başlıklı

üçüncü bölümde yazım denetleyiciler ve Türkçe için kullanımları hakkında genel bilgi verilmiş, bulanık mantık, yapay sinir ağları ve istatistiksel yöntemler tanıtılmıştır. Bu ön bilgilerden sonra uygulamanın mantığı ve aşamaları anlatılmış, yapılan deneyler ve sonuçları ile başarımı üzerinde tartışılmıştır. “Yazılımın Açıklanması” başlıklı dördüncü bölümde yazılımın detayları hakkında bilgi verilmiştir. Kullanılan sınıf ve tabloların yapıları, yazılımın yöntemleri anlatılmış, yazılımın geliştirilmesi için kullanılan ortam hakkında bilgi verilmiştir. Bu bölümde son olarak uygulamanın çalışması, ekran görüntüleri ile anlatılmıştır. Beşinci bölümde Doğal Dil İşleme ve Doğal Dil İşleme uygulamalarında kullanılan yöntemler üzerine yapılmış çalışmalar hakkında genel bir tanıtım yapılmış, tez çalışması sırasında incelenmiş çalışmalar hakkında özet anlatımlar ile genel bir kaynak taraması sunulmuştur. Son bölümde bu çalışma ile elde edilen sonuçlar belirtilmiş, yorumlanmış ve önerilerde bulunulmuştur.

2. BÖLÜM

DOĞAL DİL İŞLEME

2.1 Yapay Zeka

Yapay Zeka, insanın düşünme yapısını anlamak ve bunu modellemek üzere yapılan çalışmalar bütünüdür. İnsan gibi düşünen ve davranan sistemlerin geliştirilmesi çalışmaları 1950'lerden beri sürmektedir. Varılmak istenen nokta, insandan ayırt edilemeyecek seviyede geliştirilmiş bir bilgisayar sistemidir. Bunun olabilirliği tartışılırken, yapay zeka çalışmaları günümüzde pek çok alanda yoğun şekilde devam etmektedir. İlgi alanı insan olduğu için mühendislik dışında, nöroloji ve psikoloji alanlarında da yapılan çalışmalar yapay zekaya destek sağlamaktadır.

Yapay Zekanın alt bileşenleri şu şekilde sıralanabilir:

- Yapay Sinir Ağları
- Bulanık Mantık
- Uzman Sistemler
- Genetik Algoritma ve Programlama
- Sezgisel Yaklaşımlar
- Doğal Dil İşleme
- Ses/Konuşma Tanıma
- Robotlar
- Bilgisayarlı Görme

2.2 Doğal Dil İşlemeye Genel Bakış

Doğal Dil İşleme, bir doğal dili çözümleme, anlama, yorumlama, üretme gibi işlevlerin bilgisayar ortamında ele alındığı bir mühendislik alanıdır. 1950 ve 1960 yıllarında, yapay zekanın bir alt kolu olarak gelişmeye başlamış, konunun önemi ve

sürekliliği sayesinde günümüzde bilgisayar bilimlerindeki temel konulardan biri olarak yerini almıştır.

Doğal Dil İşleme alanında hedef, bilgisayar ile iletişimi, bir makina ile değil de insan ile iletişim ediliyor izlenimi verecek noktaya getirebilmektir. Henüz bu noktaya ulaşılammış olsa da büyük bir yol katedilmiştir.

Doğal Dil İşleme çok değişik alanlarda uygulamalara sahiptir. Örneğin yazılı bir metindeki hataları bulmak ve düzeltmek, iki dil arası çevrim yapmak, yazılanları yorumlayıp sonuçlar üretmek, bilgisayar yardımı ile dil öğrenmek, doğal dilde cümle ve metin üretmek, ses ve konuşma tanıma/üretme gibi uygulamalar ilk akla gelenleridir.

Doğal Dil İşleme beş ana başlık altında incelenmektedir.

Sesbilim : Harflerin seslerini ve bunların dil içerisindeki kullanımlarını inceler.

Biçimbilim: Sözcük kurulumu ve çözümlemesi ile ilgilenir. Sözcüklerin altyapı taşlarını inceler. Örneğin Türkçe için sözcüğün kök ve eklerini inceler.

Sözdizimbilim: Cümle oluşturmak için sözlüklerin nasıl sıralanması gerektiği ile ilgilenir. Bunlarla ilgili kural ve istisnaları inceler.

Anlambilim: Cümlenin gerçek dünyaki yerini belirtir. Cümlenin gerçekte ne amaçla kullanıldığı ile ilgilenir. Cümle geniş bir kapsamda, öğrenilmiş bilgiler eşliğinde incelenir.

Kullanımbilim: Dilin yerine göre değişimini, cümlenin kullanıldığı yere göre anlamını inceler. Sözcüklerin tek başına ya da farklı sözcüklerle beraber kullanıldığında ne ifade etmek istediği ile ilgilenir.

Yukarıda belirtilen başlıklar her dilin özelliğine ve kullanılan doğal dil işleme yöntemine göre farklı biçimde incelenir.

2.3 Diller

Dünyada ortalama 4000 dil konuşulmaktadır. Bunlar arasında en yaygın kullanıma sahip olanlar Çince, Hintçe, İspanyolca, İngilizce, Fransızca, Rusça ve Türkçedir.

Biçim açısından dünya dilleri dört ana başlık altında toplanmıştır:

Yalınlayan Diller: Bu dillerde sözcükler genelde tek heceden oluşurlar. Sözcükler cümle içerisinde biçimsel olarak değişmezler fakat değişik vurgu ve tonlamalarla, farklı anlamlar kazanırlar. Yalınlayan dillere örnek olarak Çince, Bask dili verilebilir.

Bağlantılı Diller: Bu dillerde sözcükler köklere getirilen eklerle oluşturulur. Çekim ve türetme sırasında kök değişmez, köke bağlanan ekler ile binlerce sözcük türetilebilir. Bağlantılı dillere örnek olarak başta Türkçe olmak üzere, Fince, Macarca, Moğolca verilebilir.

Kaynaştıran Diller: Bu dillerde köklere son ekler gelir. Bu bağlamda bağlantılı dillere benzerler. Genellikle eylemleri diğer öğelerle birleştirerek sözcükler oluştururlar. Bir cümlenin bir sözcük haline geldiği durumlar dahi oluşabilir. Kaynaştıran dillere örnek olarak Eskimoca ve Gürcüce verilebilir.

Bükümlü Diller: Bu dillerde çekim ve türetim kök değiştirilerek yapılır. Bükümlü diller de kendi aralarında kök bükümlü ve gövde bükümlü olmak üzere ikiye ayrılırlar. Kök bükümlü dillere örnek olarak Arapça, gövde bükümlü dillere örnek olarak da Almanca, Fransızca ve İngilizce verilebilir.

Kaynak ve Akrabalık İlişkileri açısından diller beş ana başlık altında toplanmıştır:

Hint-Avrupa Dil Ailesi: Bu aileye ait diller arasında Almanca, İngilizce, Fransızca, İtalyanca, Rusça, Ermenice ve Farsça sayılabilir.

Sami Dil Ailesi: Bu aileye ait dillere örnek olarak Arapça ve İbranice verilebilir.

Bantu Dil Ailesi: Orta ve Güney Afrika'da kullanılan diller, bu aileye örnek diller arasındadır.

Çin-Tibet Dil Ailesi: Bu aileye ait diller arasında Çin ve Tibet dilleri sayılabilir.

Ural-Altay Dil Ailesi: Bu ailedeki diller yapı bakımında birbirleri ile benzerlik gösterirler. Hepsi bağlantılı dillerdir. Hepsi belli bir derecede ünlü uyumuna sahiptirler. Sözdiziminde sözcük sıralaması aynıdır.

Bu dil ailesi, Ural ve Altay olmak üzere iki kola ayrılmıştır. Ural koluna ait diller arasında Türkçe ve Moğolca sayılabilir. Altay koluna ait diller arasında da Fince ve Macarca yer almaktadır.

2.4 Türkçe

*“Ülkesini, yüksek istiklalini korumasını bilen Türk milleti,
dilini de yabancı diller boyunduruğundan kurtarmalıdır.”*

Atatürk

Türkçe, biçim açısından bağlantılı dillere, kaynak ve akrabalık ilişkileri açısından da Ural-Altay Dil ailesinin Altay koluna girmektedir. Türkçe, ses ahengi açısından gelişmiş bir dildir. Sesi ahengi dildeki sesli sayısı ile doğru orantılıdır.

Yapısı bakımından Türkçe, kural tabanlı bir dildir. Sözcükler köklere getirilen eklerle oluşturulur. Çekim ve türetme sırasında kök değişmez, köke bağlanan ekler ile binlerce sözcük türetilir. Türk Dil Kurumu’nun güncel sözlüğünde 100,353 adet sözcük bulunmaktadır. Bu sözcüklerden türetilebilecek yeni sözcükler de düşünüldüğünde günlük konuşma ve yazışma sırasında kullanılabilecek sözcükler kümesi oldukça büyüktür.

Yabancı kelimeler ve dile sonradan eklenmiş kelimeler dışında Türkçe bir kurallar dizisi şeklinde yapılandırılabilir. Ne yazık ki dilimize sonradan girmiş yabancı sözcükler, bu kurallar bütünü ile hareket etme olanağını sınırlandırmaktadır. Bu yüzden Türkçe’yi yapılandırmak için kurallardan daha ileri düzeyde, kuralları ihlal ediyor olsa da günlük hayatta kullanılan konuşma dilini kapsayan çalışmalar yapılmalıdır.

3. BÖLÜM

BULANIK MANTIK VE YAPAY SİNİR AĞLARI İLE TÜRKÇE YAZIM DENETLEYİCİSİ TASARIMI VE GERÇEKLEŞTİRİMİ

3.1 Yazım Denetleyicilere Genel Bakış

Yazım Denetleme işlemi, Doğal Dil İşleme alanında çok önemli bir yere sahiptir. Günümüzde yazılı iletişim için gerek bilgisayar ortamından klavye ile girilmiş, gerekse kağıttan taranmış metinler kullanılmaktadır. Bu metinlerin doğruluğu iletişimin doğruluğu açısından büyük önem kazanmaktadır.

3.2 Türkçe İçin Yazım Denetleyici

Genel anlamda yazım denetleyiciler, yazılı metni sözcüklere ayırıp, önceden hazırlanmış bir sözlükte bu sözcükleri arama mantığı ile çalışmaktadırlar. İngilizce, Almanca gibi yalın dillerde türetilebilecek kelime sayısının yönetilebilecek düzeyde olması, sözlük tabanlı yazım denetleyicilerini uygun kılmaktadır. Fakat Türkçe gibi bağlantılı dillerde, bir köke çeşitli sıra ve sayıda ekler ekleyerek sınırsız sözcük türetmek mümkündür. Türetilebilecek tüm sözcüklerin bir sözlükte toplanması neredeyse imkansızdır. Bu yüzden Türkçe için özelleşmiş sözlüklerle (kök sözlüğü gibi) ya da tamamen sözlüksüz bir yazım denetleyicisi tasarlama yöntemi geliştirmelidir.

Türkçe kural tabanlı bir dildir. Fakat yabancı sözcüklerin dile eklenmesi ile kural dışı birçok kullanım bulunmaktadır. Türkçe için tasarlanacak bir yazım denetleyicisinin bu kural dışı sözcükler de dahil, günlük konuşma dilini de kapsayacak bir yapıda olması gerekmektedir. Bu bağlamda, farklı yaklaşımlar ele alınmalıdır. Bu çalışmada istatistiksel yaklaşımlar ve istatistiksel yaklaşımlar temeline dayanan bulanık mantık ile yapay sinir ağları ele alınmıştır.

3.3 İstatistiksel Yöntemler

Genel anlamda istatistik, verilerin toplanması, organize edilmesi, özetlenmesi, sunulması, bazı tekniklerle analiz edilmesi ve bu verilerden bir sonuca varılabilmesi için kullanılan metodlar topluluğudur. Kurallar ile elde edilemeyen analizleri yapabilmek için sıklıkla başvurulanan bir yöntemdir.

Doğal Dil İşleme açısından düşünüldüğünde, istatistiksel yöntemlerin yoğunlukla kullanıldığı konuların başında doğal dil girdi ve çıktıların işlenmesi, eğitim kümesinden öğrenme, doğal dil işleme sistemlerinin başarımını ölçme, doğal dil işleme ile elde edilmiş sonuçlar arasındaki belirsizlikleri (puanlama ile) çözme sayılabilir.

İstatistiksel yöntemlerin en büyük avantajı kurallardan bağımsız çalışması, yani kural tanımlama zorunluluğunun ortadan kalkmasıdır. Kurallar kendiliğinden oluşmakta, sistem kendi kurallarını dinamik olarak kendi oluşturmaktadır.

Bu tezde istatistiksel yöntemler, öğrenme aşamasında kullanılan yapay sinir ağlarına veri sağlama ve sözcükleri karşılaştırma sırasında bulanık mantık ile elde edilen benzerlik puanını iyileştirmek için kullanılmıştır. Ayrıca kök ve eklerin kullanım sayılarına göre eğitim sırasında kullanım oranları elde edilmekte ve bu oran iki sözcüğü karşılaştırma sırasında elde edilen bütünleşik puanın bir parçasını oluşturmaktadır.

3.4 Bulanık Mantık

“Ali çok sağlıklıdır. Sağlıklı insanlar uzun yaşarlar. Öyleyse, Ali çok uzun yaşayacaktır.” Bu tarz bir yaklaşımı klasik mantıklarla ifade etmek zordur. Çünkü klasik mantık (“doğru” ve “yanlış” kavramlarından oluşan iki-değerli mantık) sistemlerinin çoğu, bu tür cümleleri ilgi alanlarının dışında bırakırlar. Fakat bu tür cümleleri ve kıyaslamaları günlük yaşantımızda çok sıklıkla kullanırız.

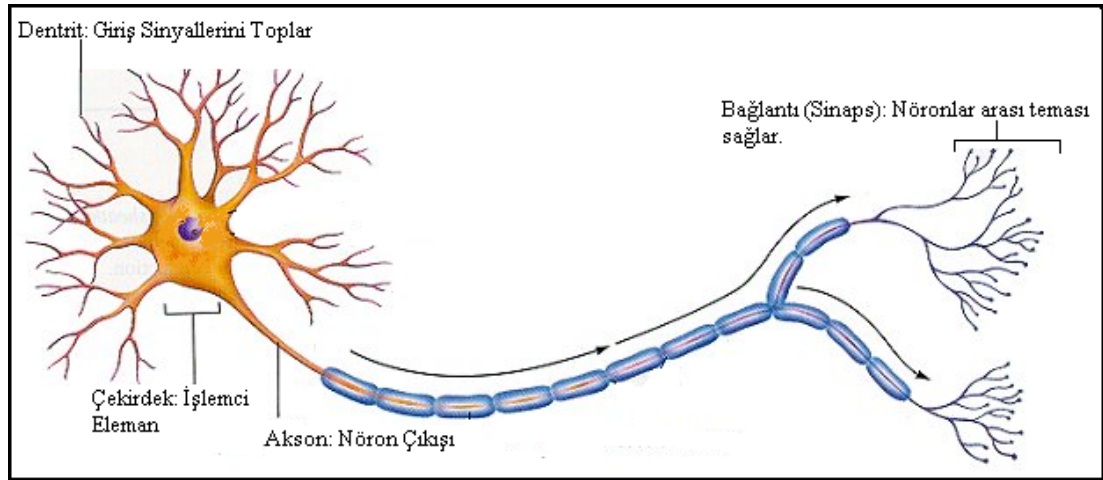
Bulanık mantık (Fuzzy Logic) kavramı ilk kez 1965 yılında California Berkeley Üniversitesinden Prof. Lotfi A. Zadeh'in bu konu üzerine ilk makalelerini yayınlamasıyla duyulmuştur. Klasik mantıklarda, bir önerme ya doğrudur ya da yanlıştır. Bunlar dışında bir durumun gerçekleşmesinin imkansız olduğu kabul edilir. Oysa gerçek hayatta bu kadar kesin yargılara varmak neredeyse imkansızdır.

Kavramlar doğru ve yanlış arasında birçok şekilde ifade edilebilecek haldedir. Bu bağlamda, bulanık mantık kavramı genel olarak insanın düşünme biçimini modellemek üzere ortaya çıkmıştır. Bulanık mantığın temelinde yatan, bir önermenin “doğru”, “yanlış”, “çok doğru”, “çok yanlış”, “çok çok doğru”, “çok çok yanlış”, “yaklaşık olarak doğru”, “yaklaşık olarak yanlış”, v.b. gibi olabileceğidir.

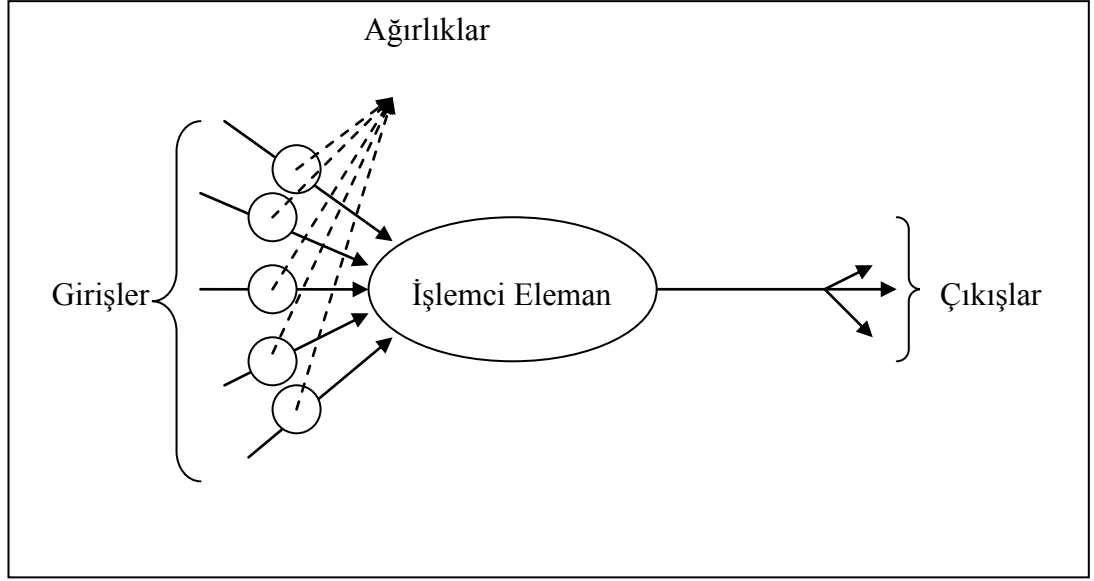
Bulanık sistemler, eğitilebilir devingen sistemlerdir. Bir fonksiyonu, çıktıların girdilere ne şekilde bağlı olduğunun matematiksel modelini bilmeksizin tahmin ederler. Sayısal, bazen dilsel örnek verilerden “deney yoluyla” öğrenirler. Uyarlanabilir bulanık sistemler, karmaşık süreçleri kontrol etmeyi, neredeyse bizler gibi öğrenebilirler. Bu bağlamda, bulanık mantığın en önemli ve güçlü tarafı, var olan bir uzman bilgisinin kullanılmasıdır.

3.5 Yapay Sinir Ağları

İnsan beyninin işlevini özellikle öğrenmeyi modellemek için üretilmiş sistemlerdir. Bu sistemler yapay sinir hücrelerinin ağırlıklar üzerinden birbirlerine bağlanması ile oluşmuşlardır. Şekil 3.1’de görülen insan sinir hücresine benzetilmiş, giriş ve çıkışları bulunan bir merkezi sistem şeklindedirler.



Şekil 3.1: İnsan Sinir Hücresi



Şekil 3.2: İnsan sinirinin modellenmesi

Şekil 3.2’de insan sinirinin nasıl modellendiği gösterilmektedir.

Yapay sinir ağları, hücrelerin bağlantı türlerine göre ileri beslemeli ve geri beslemeli olarak iki sınıfa ayrılabilir. İleri beslemeli ağlarda, bir hücre ancak kendinden sonraki hücrelere bağlı olabilir, fakat geri beslemeli ağlarda, en az bir hücrenin çıkışı kendinden önceki bir hücreye giriş olarak bağlanmıştır.

Öğrenme türüne göre bir sınıflandırma yapılırsa, denetimli ve denetimsiz öğrenen ağlar şeklinde ikiye ayrılabilir. Denetimli öğrenmede doğru çıktı bilinmekte ve YSA’nın ürettiği çıkış ile doğru çıktı arasındaki hataya göre sistem öğrenmektedir. Denetimsiz öğrenmede ise doğru çıktının ne olduğu bilinmemektedir. Ağın eğitimi için eğitici bir bilgiye gerek duyulmamakta, sadece giriş bilgileri yeterli olmaktadır.

Bu tezde yapay sinir ağları, öğrenme aşamasında kullanılmaktadır. Hata denetleme sırasında karşılaşılan hatalar yapay sinir ağları ile öğrenilerek daha sonra bu hata ile tekrar karşılaşma durumunda öğrenilen bilgiler ile hareket edilmektedir. Sık yapılan hatalar için dinamik olarak kurallar oluşturulmakta, bu kurallar çerçevesinde hata bulma ve düzeltme sırasında kullanılan puanlama işlemi öğrenilen bilgiler ile daha akıllıca yapılmaktadır.

3.5 Bulanık Mantık Ve Yapay Sinir Ağları İle Türkçe Yazım Denetleyicisi

Bu tezde, Bulanık Mantık ve Yapay Sinir Ağları ile Türkçe bir yazım denetleyicisi tasarlanmış, elde edilen sonuçlar yorumlanmaya çalışılmıştır.

Türkçe bağlantılı bir dildir. Bir köke ekler eklenerek sınırsız sayıda sözcük türetilebilir.

Örneğin; “Osmanlılaştıramayabileceklerimizdenmişsinizcesine” kelimesi “Osman” köküne 12 adet ek getirilerek türetilmiştir. Daha da türetilmeye açık gözükmektedir. Örnekteki sözcüğün kök ve eklerine ayrılmış (biçimbirimsel çözümlenmiş) hali “Osman+lı+laş+tır+ama+yabil+ecek+ler+imiz+den+miş+siniz+cesine” şeklindedir.

Genel anlamda Yazım Denetleyiciler bir sözlük yardımıyla sözcüğün doğruluğunu kontrol edip, hatalı ise yine aynı sözlükten önermeler yaparak çalışır. Fakat Türkçe gibi bağlantılı bir dilde, günlük hayatta kullanılan tüm sözcükleri kapsayan bir sözlük oluşturulması neredeyse imkansızdır. Bu yüzden bu tezde benimsenen yaklaşım, Türkçe için yapılacak bir yazım denetleyicisinde sözlük kavramından daha farklı olarak kaynak alınacak sözcük kümesi kavramı geliştirilmiştir. Bu küme belli bir konuda hazırlanmış büyük bir eğitim kümesi ile oluşturulmuştur. Kaynak küme sözcük bilgisi ile değil de sözcüklerin temel yapısı olan kökler ile oluşturulmaktadır. Sadece köklerden oluşan bir kaynak kümesi üzerinden karşılaştırmalar yapılmaktadır.

Türkçe yapısı bakımından kurallı bir dildir. Tanımlanmış kurallar doğrultusunda Türkçe sözcükler çözümlenebilir ve türetilebilir. Fakat ne yazık ki gerek dile girmiş olan birçok yabancı sözcük, gerek yıllar boyunca sözcüklerin kullanımında oluşmuş değişiklikler kural dışı kullanımların oluşmasına yol açmaktadır. Günlük hayatta kullanılan kural dışı sözcük sayısı yadsınamaz orandadır. Bu yüzden, kural tabanlı bir yaklaşım ile yapılacak bir yazım denetleyicisi, kural dışı sözcükleri ele almak açısından yetersiz kalacaktır. Bu bağlamda bu tezde, kurallar değil, kullanım ile ilgili istatistiksel veriler ve öğrenilen bilgiler kullanılmıştır. Büyük bir eğitim kümesi kullanılarak kaynak kümeler oluşturulmuş, bu kaynak kümelerle ait istatistiksel veriler türetilmiş ve bu veriler kullanılarak sonuçlar üretilmiştir. Bu bağlamda eğitim işleminin iki parçadan oluştuğu söylenebilir. Birinci kısım yukarıda bahsedilen kaynak küme ve istatistiklerinin oluşturulmasıdır. İkinci kısım ise denetleme işlemi

sırasında, denetlenmiş bilgiler ışığında elde edilmiş bilgiler ile hataların öğrenilmesi, dolayısıyla sistemin daha akıllı hale getirilmesidir.

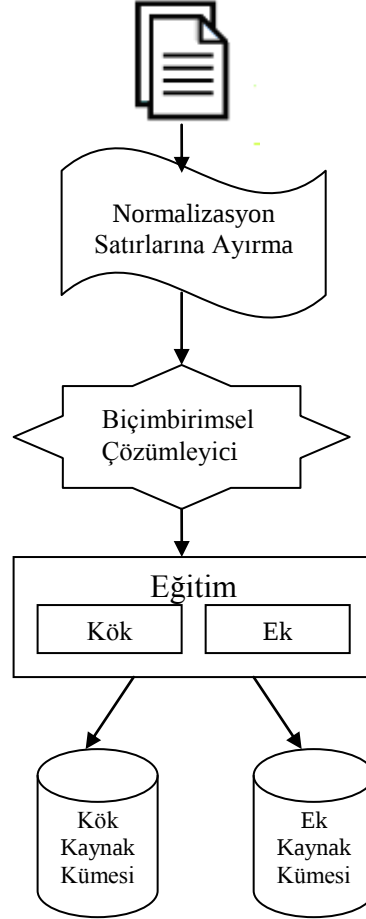
Türkçe sözcükler temel olarak kök ve eklerden oluşmaktadır. Kökler değişmeden kullanılırken, ekler belli bir sıralanışa göre dizilirler. Kural dışı sıralanışlar da mevcuttur. Bu tez, sözcükleri kök ve ekler olmak üzere iki kısımda inceleyip, her kısmı hem kendi içinde değerlendirip sonuçlar üretmek temeline dayanmaktadır.

Bu amaç doğrultusunda büyük bir eğitim kümesi alınmıştır. Bu kümedeki tüm sözcükler bir biçimbirimsel çözümleyici tarafından kök ve eklerine ayrılmıştır. Elde edilen kökler eğitim kümesinde geçme sayıları ve hesaplanan kullanım oranları ile birlikte bir veritabanında saklanmıştır. Bu veritabanındaki kökler daha sonra yazım denetleme aşamasında karşılaştırma amaçlı kullanılmıştır. Biçimbirimsel analiz sonucunda elde edilmiş olan ekler de daha sonra yazım denetleme aşamasında kullanılmak üzere yine bir veritabanında saklanmıştır. Türkçe’de eklerin sıralanışı genelde belli bir düzendedir. Bu yüzden eklerin sıralanışı ile ilgili istatistiksel bilgileri tutup, bu veriler üzerinden sonuç üretmek anlamlıdır. Bu istatistiksel bilgiler belli sıralanıştaki eklerin kullanım sayılarına göre hesaplanmış kullanım oranları şeklindedir. Ek sıralanışlarını ele almak için ekler, art arda gelmiş ek ikilileri şeklinde ele alınmaktadır.

3.5.1 Genel Yapı

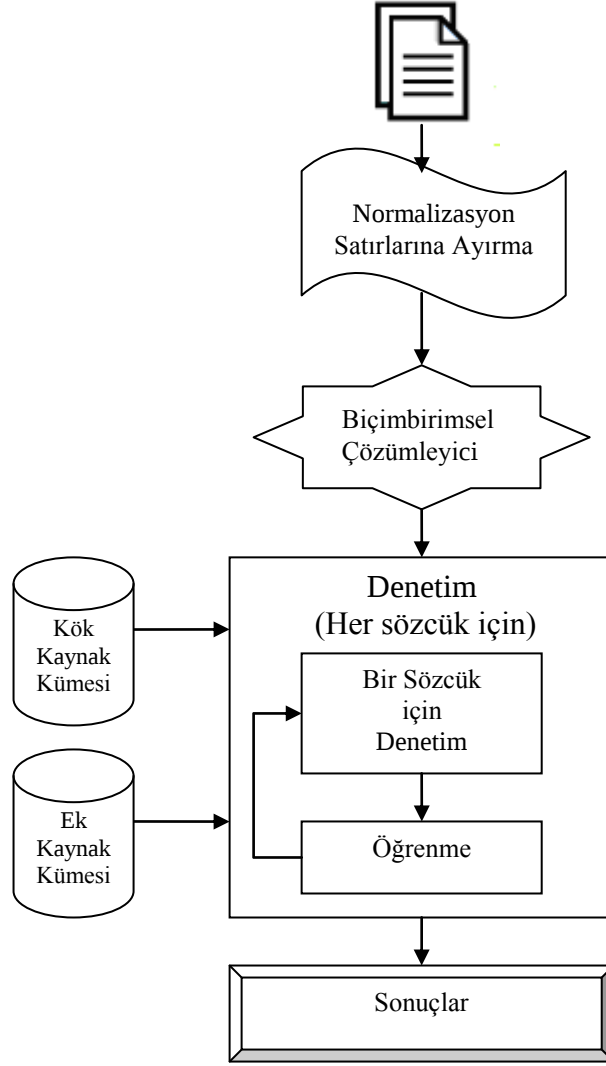
Uygulamanın yapısı, hata denetimi yapılacak metindeki tüm sözcükleri bir kaynak kümeyi temel alarak karşılaştırma yapma esasına dayanmaktadır. Öncelikle büyük bir eğitim dosyası işlenerek sisteme alınır. Şekil 3.3’de eğitim işleminin genel yapısı gösterilmektedir. Eğitim dosyası harf dışındaki karakterlerden arındırılıp, her satırda bir sözcük yer alacak şekilde sözcüklerine ayrılır. Daha sonra bu yapıdaki her sözcük biçimbirimsel olarak çözümlenerek kök ve eklerine ayrılır. Kökler bir kök tablosunda, eklerin ikili sıralanışları da ekler tablosunda yer alır. Her kök ve ek ikilisinin eğitim kümesi içerisinde geçme sayısı ve bu sayıların eğitim kümesindeki toplam sözcüklere oranından hesaplanan kullanım oranları da bulunup bu tablolara yazılır. Burada amaç, Türkçe için tüm sözlüğü kaynak olarak kullanmak yerine, sıkça kullanılan sözcükleri ya da belli bir konuda kullanılan sözcükleri kaynak almaktır. Bu yapı aynı zamanda konuya bağımlı bir yapıyı da beraberinde getirmektedir.

Eğitim işleminin bir kısmını da denetleme ardından yapılan öğrenme işlemi oluşturmaktadır. Öğrenme işlemine denetleme işlemi kapsamında ayrıca değinilmektedir.



Şekil 3.3: Eğitim işleminin genel yapısı

Kaynak kümeler oluşturulduktan sonra yazım denetleme kısmı hazır durumdadır. Yazım denetimi yapılacak metin için yine metin işleme işlemi yapılır. Yani metin harf dışındaki harflerden arındırıldıktan sonra, sözcüklere, sözcükler de biçimbirimsel çözümleyici ile kök ve eklerine ayrılır. Denetim süreci Şekil 3.4’de şematik olarak gösterilmektedir. Burada öncelikle puanlama devreye girmektedir. Puanlama, bulanık mantık puanlaması, öğrenme ve sözcüklerin kullanım oranlarından elde edilmiş bütünleşmiş bir puanlama yapısından oluşmaktadır. Denetleme işlemi sırasında öğrenilecek bilgiler bir öğrenme yapısı içerisinde saklanır. Denetleme işleminin ardından öğrenme işlemi devreye girmektedir. Öğrenilecek bilgiler ışığında sistemin ağırlıkları güncellenir yani sistem öğrenir. Böylece denetleme sırasında karşılaşılan hatalar sistem tarafından, bir sonraki denetlemede kullanılmak üzere öğrenilir.



Şekil 3.4: Denetim işleminin genel yapısı

Puanlama işlemleri sırasında iki sözcükteki harfler karşılaştırılmaktadır. Bu bağlamda üç çeşit kavram kullanılmaktadır. Tutan harfler, yeri tutmayan harfler ve tutmayan harfler. Tutan harfler, her iki sözcükte de aynı yerde bulunan harflerdir. Yeri tutmayan harfler, her iki sözcükte de yer alan fakat sözcüklerin farklı yerlerinde bulunan harflerdir. Tutmayan harfler ise iki sözcüğün aynı yerinde bulunan fakat farklı olan harflerdir.

Örneğin: “kitap” ve “kışpa” sözcükleri karşılaştırılırken

k ve t : tutan harfler

ş-t : tutmayan harf çifti

a ve p : yeri tutmayan harflerdir.

Tez metni boyunca “tutan harfler”, “tutanlar” ifadeleri, her iki sözcükte de aynı yerde yer alan harfleri, “yeri tutmayan harfler”, “yeri tutmayanlar” ifadeleri her iki sözcükte de yer alan fakat farklı yerlerde bulunan harfleri ve son olarak “tutmayan harfler”, “tutmayanlar” ifadeleri de iki sözcükte aynı yerde yer alan fakat birbirinden farklı olan harfleri ifade etmek için kullanılmaktadır. Bu ifadelerden tutan harfler, puanlama ile ilgili tüm kısımlarda, yeri tutmayan harfler, genelde bulanık mantık ile ilgili kısımlarda, tutmayan harfler ise öğrenme ile ilgili kısımlarda yoğun olarak kullanılmaktadır.

Bulanık mantık puanlaması, iki sözcüğün tek tek harflerinin karşılaştırılıp, her iki sözcükte yer alırken yeri tutan ve tutmayan harflere göre, sözcüklerin benzerlik olasılığının bulunması şeklindedir. Puanlama öğrenme yapısı ile zenginleştirilip daha akıllı hale getirilmiştir. Öğrenme puanlaması, hatalı harf ikilileri sayesinde elde edilmiş ağırlıklar üzerinden yapılır. Bu ağırlıklar, tutan harfler, tutmayan harfler, tutmayan harflerin karşılaşıma sayısı, klavyedeki uzaklıkları, yazım benzerlikleri kısımlarından oluşur. Her denetleme işlemi sonrasında, son denetlenmiş sözcükler ele alınıp, bu ağırlıklar güncellenir. Genel puanlamanın son ögesi, eğitim sırasında sözcüklerin kaynak kümedeki kullanım sıklıklarına göre hesaplanmış kullanım oranlarıdır.

Puanlama işlemi iki aşama için gerçekleştirilir. Birincisi köklerin karşılaştırılıp puanlanması, ikincisi ise eklerin karşılaştırılıp puanlanmasıdır. Yazım denetimi yapılacak metnin içindeki sözcükler sırası ile ele alınır. Her sözcük çözümlendikten sonra, elde edilen kökün, kaynak kümesinde varlığı araştırılır. Eğer yok ise denetlenen kök hatalı demektir ve kaynak kümesinde bu sözcüğe yakın tüm kökler ile karşılaştırılıp, puanlanır. Puanı belli bir seviyeden yüksek bulunan karşılaştırmaların en yüksek puanlısı hatalı köke karşılık gelen doğru kök önermesidir. Puanlama 0 ile 1 arasında bir değer alır.

Ekler için puanlama yapılması sırasında, denetlenecek sözcük çözümlemesinden elde edilen her ek ikilisi için işlem yapılır. Eğer bu ek ikilisi, ekler kaynak kümesinde yer alıyorsa, hatasız bir ikilidir. Yer almıyorsa, ek kaynak kümesinden, önce ilk eki denetlenen ek çiftinin ilk eki olan tüm kayıtlar (ek ikilileri) çekilir. Çekilen bu listedeki kayıtların ikinci ekleri, denetlenen ek çiftinin ikinci eki ile karşılaştırılıp, puanlanır. Aynı işlem denetlenen ek çiftinin ikinci eki için de yapılır. Bu sefer ek kaynak kümesinden, ikinci eki, denetlenen ek çiftinin ikinci eki olan kayıtlar çekilir

ve ilk ekler karşılaştırılarak puanlanır. Tüm bu ek ikilisi puanlama işlemlerinden elde edilen en yüksek puanı oluşturan ek çifti, hatalı ek ikilisi için yapılan önermedir. Bu işlemler, çözümlenen sözcükten elde edilen tüm ek ikilileri için tekrarlanır. Bu işlemin detayı bölüm 3.5.6’da açıklanmaktadır.

Tüm metin içinde yakalanmış hatalar ve bunlar için oluşturulmuş önermeler kullanıcıya sunulur.

3.5.2 Sözcüklerin Çözümlemesi

Biçimbilim, sözcükleri, sözcük türlerini, kökler ve eklerle ilgili her türlü konuyu, yani dilin yapı özelliklerini inceleyen dilbilgisi dalıdır. Türkçe bağlantılı bir dil olduğu için sözcük çözümlemesinde ve türetilmesinde kökler, ekler ve bunların birleşim, sıralanış özellikleri diğer dillere göre çok daha önemli ve anlamlıdır.

Türkçe’de kökler değişmez. Kök sonuna gelen ekler yeni sözcüklerin türetilmesini sağlar. Kökler değişmediği için, bir yazım hatası denetleyici açısından köklerin ayrı olarak incelenmesi mantıklıdır.

Sözcüklerin görevlerini belirtmeye ya da sözcük türetmeye yarayan anlam değiştirici ses ve ses birleşimlerine ek denir. Dünya dillerinde ön ek, iç ek ve son ek olmak üzere üç çeşit ek kullanılmaktadır. Türkçe’de yalnızca sonekler vardır. Kök sonuna eklenen ek sayısı ortalama 3 - 5 civarında olup, günlük kullanımda ek sayısının 12’ye kadar çıktığı durumlar mevcuttur. Eklerin sıralanışının incelenmesi bakımından, kök ardından gelen eklerin incelenmesi yeterli olacaktır. Yabancı dillerden dile katılmış önekli sözcükler ise önekleriyle beraber bir kök olarak varsayılacaktır.

Bu tezde, büyük bir eğitim kümesi sözcüklerine ayrılıp, her sözcük bir biçimbirimsel çözümleyici tarafından kök ve eklerine ayrılmaktadır. Biçimbirimsel çözümleme işleminden önce metin normalize edilir. Yani harfler dışındaki karakterlerden temizlenir, tüm metin küçük harfe dönüştürülür. Ayrıca Türk alfabesinde yer almayan fakat birçok metinde rastlanan â, î, û harfleri sırasıyla a, i, u harflerine dönüştürülür. Metin sözcüklerine ayrılıp, her sözcük bir satıra gelecek şekilde biçimlendirilir.

Biçimbirimsel çözümleyici tarafından elde edilmiş kökler eğitim kümesi içerisinde geçme sayıları ile birlikte bir veritabanında saklanmaktadır. Ekler de sıralanışlarına göre yine veritabanında yer almaktadır. Ekleri ele alırken ana mantık, eklerin

sıralanışı üzerine kurulmuştur. Çünkü Türkçe’de ekler sıralanışları ile köklere anlam kazandırmaktadırlar. Bu yüzden ek kaynak kümesinde ekler tek başlarına değil, art arda gelmiş ek ikilileri şeklinde tutulmaktadır. Türkçe’de kök sonuna gelen ortalama ek sayısının 3 - 5 civarında olduğu düşünülürse, eklerin ikili sıralanışları üzerinden çalışmak mantıklıdır. Bu bağlamda, kök sonuna art arda gelmiş her iki ek, veritabanında, eğitim kümesi içerisinde geçme sayıları ve kullanım oranları da hesaplanarak yer almaktadır.

Örneğin: “bilgilerimizin” sözcüğünün biçimbirimsel çözümlemesi “bilgi+ler+imiz+in” şeklindedir. “bilgi” kelimesi kök, “ler”, “imiz” ve “in” sesleri eklerdir. Tablo 3.1’de görüldüğü gibi veritabanında “ler-imiz” ikilisi bir kayıt, “imiz-in” ikilisi bir kayıt, “in-İ” ikilisi de diğer bir kayıt oluşturur. Bu ikililerin daha önce de kullanılmış olmasına göre kullanım sayısı hesaplanır. Aynı ikili için her yeni kullanımında yeni bir kayıt yaratılmaz. Sadece kullanım sayısı bir artırılır.

Tablo 3.1: EKK’de “bilgilerimizin” sözcüğünün ekleri görünümü

Ön Sesliler	Ek1	Ek2	Ek İkili Kullanım Sayısı	Ek İkili Kullanım Oranı
i	ler	imiz	2	0,0002
i	imiz	in	1	0,0001
ieüö	in		266	0,0206

Tüm eğitim sözcükleri okunup veritabanına yazıldıktan sonra, ek ikililerinin kullanım oranları bulunur. Kullanım oranı aşağıdaki formüle göre hesaplanır:

$$K_o = \frac{K_s}{\sum K_s} \quad (3.1)$$

Büyük ünlü uyumu, Türkçe’de eklerin sıralanışı bakımından büyük önem taşımaktadır. Bir sözcükte kalın ünlülerden sonra kalın, ince ünlülerden sonra ince ünlülerin gelmesi kuralıdır. Kök sonuna gelecek olan eklerde de (bazı kural dışı ekler hariç) büyük ünlü uyumu aranır. İstisnaların olması, istatistiksel ve öğrenme temelinde bir çalışmanın daha anlamlı olacağı düşüncesi doğrultusunda, kök ile ardından gelen ekler arasındaki büyük ünlü uyumu istatistiksel olarak elde edilmiştir. Her ek ikilisi bulunduğu zaman, bu ek ikilisinden önce gelebilecek sesli harfler de tutulur. Bu, ek ikilisinin bağlı olduğu kökün son sesli harfi alınarak sağlanır.

Tablo 3.2’de köklerin tutulduğu tablonun, Tablo 3.3’de de eklerin tutulduğu tablonun bir kısmının görüntüsü yer almaktadır.

Tablo 3.2: Kök Kaynak Kümesi görünümü

Kök	Kullanım Sayısı	Kullanım Oranı
antik	9	0,0008
yakındoğu	30	0,0028
hak	6	0,0006
bilgi	7	0,0007
çoğ	18	0,0017
çoğu	18	0,0017
son	16	0,0015
yıl	40	0,0038
arkeoloji	17	0,0016
al	99	0,0093
ala	42	0,0039
alan	73	0,0069
çalış	12	0,0011
ara	37	0,0035
aracı	1	0,0001
aracılığ	1	0,0001
el	20	0,0019

Tablo 3.3: Ek Kaynak Kümesi görünümü

Ön Sesliler	Ek1	Ek2	Ek İkili Kullanım Sayısı	Ek İkili Kullanım Oranı
ao	tik		34	0,0008
aio	ti	k	116	0,0028
aüeio	k		144	0,0035
ouai	u		371	0,009
aeoiüöü	i		1354	0,0327
ıauo	da	ki	115	0,0028
ıaiüeuoö	ki		254	0,0061
ıauo	nda	ki	61	0,0015
ıaou	n	da	187	0,0045
ioaerüü	n		2460	0,0594
ieoüiauö	in		1140	0,0275
i	miz	in	1	0
i	ler	imiz	2	0
i	imiz	in	1	0
ıauo	da		495	0,012

Bu tezde, biçimbirimsel çözümleme için hazır yazılımlar kullanılmıştır. Biçimbirimsel çözümleyicinin ürettiği sonuçların doğruluğu doğrudan doğruya yazım hatası denetleyici verimliliğini etkilemektedir. Bu yüzden farklı biçimbirimsel çözümleyiciler kullanılıp, sonuçlar karşılaştırılmıştır.

Bu çalışmada, kullanılan biçimbirimsel çözümleyiciler:

- Zemberek yazılımı [31]
- Gülşen Cebiroğlu'nun yüksek lisans tez çalışması; "Sözlüksüz Köke Ulaşma Yöntemi"[32]
- Kemal Oflazer'in biçimbirimsel çözümleyicisi [35]

Zemberek Yazılımı: Ahmet A. Akın tarafından 2004 yılında eski Tspell* projesinin devamı olarak başlatılmıştır. Proje, Türkçe diline ilişkin çeşitli bilgi işlem problemlerinin çözümlenmesi için açık kodlu, platform bağımsız bir doğal dil işleme kütüphanesidir. Kelime çözümleme için 20,000 kök ve 6,000 özel isimden oluşan bir sözlük kullanılmıştır.

Yazılım sözcüğü köklerine ve eklerine ayırıp, ekleri olduğu gibi değil, eklerin türleri şeklinde vermektedir.

Örnek:

{Icerik: bilgilerimizin Kok: bilgi tip:IS} Ekler:ISIM_YALIN + ISIM_COGUL + ISIM_SAHİPLİK_BİZ + ISIM_IN

Fakat bu tezde ihtiyaç olan yapı, kök ve eklerin asıl hallerinden oluşmuş bir biçimdedir. Bu amaçla, proje sahiplerinin [33] de fikir ve yardımlarıyla, açık kaynak kodlu yazılım üzerinde çeşitli değişiklikler yapılmış ve istenen biçimde sonuçlar üretilmesi sağlanmıştır.

Örnek: bilgi+ler+imiz+in

Bir sözcüğe ait birden fazla çözümleme var ise, tüm farklı çözümlemeler uygulama kapsamında ele alınmıştır.

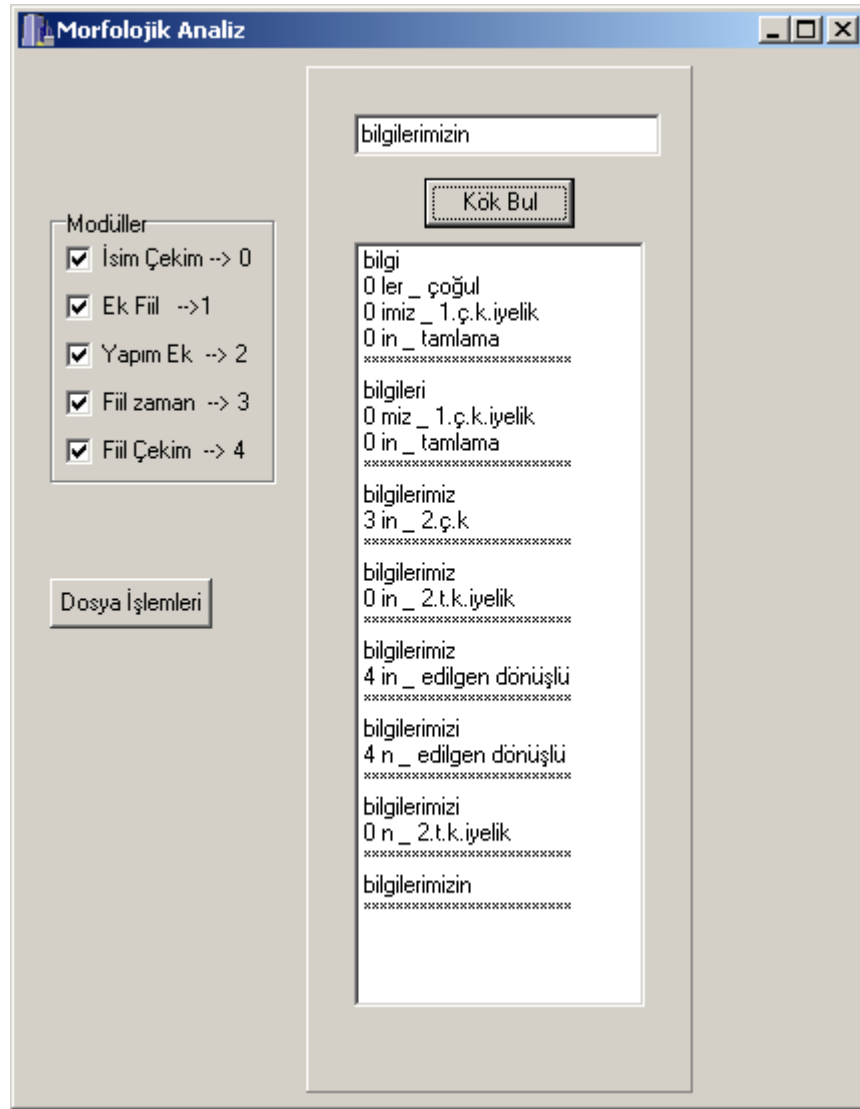
Sözlüksüz Köke Ulaşma Yöntemi: Bu çalışmada, bir sözcük köküne getirilebilecek ekler kümelerine ayrılmış ve her küme için, eklerin sözcüğe getiriliş sıraları

* Tspell: Türkçe dili ile ilgili yazılım uygulamalarında kullanılmak için gerekli araçları içeren bir kütüphane oluşturma amacı ile başlatılmış bir projedir.

kurallaştırılarak, sonlu durum makineleri oluşturulmuştur. Kümelere özgü oluşturulan makineler birleştirilerek ortaya çıkan ana makine yardımıyla yapılan biçimbirimsel analizde, bir sözcüğe getirilen eklerin sondan başlanarak başa doğru sözcükten çıkarılması ile, sözcük köküne ulaşılmıştır.

Tez boyunca ve uygulamada, bu çalışmadan “Morfolojik Analiz (Eryiğit)” ifadesi kullanılarak bahsedilmektedir.

Uygulama, sözcüğü eklerine ayırarak köküne ulaşma temelinde çalışırken, sonuçları Şekil 3.5’de görüldüğü gibi ek türleri şeklinde vermektedir. İhtiyacı karşılayacak biçime (eklerin türleri değil gerçek halleri biçimi) dönüştürme çalışması tez sahibi [34] ile beraber gerçekleştirilmiştir.



Şekil 3.5: Morfolojik Analiz (Eryiğit) uygulaması görünümü

Bir sözcüğe ait birden fazla çözümleme var ise, tüm farklı çözümlemeler uygulama kapsamında ele alınmıştır.

Morfolojik Analiz (Eryiğit) çalışması özellikle denetlenecek sözcüklerin biçimbirimsel çözümlemesi açısından önem kazanmaktadır. Çünkü diğer biçimbirimsel çözümleyiciler yazım hatası olan sözcükleri çözümlememektedir. Oysaki bu tezde ihtiyaç olan, hatalı olsa da sözcüklerin çözülmesidir. Hatalı sözcüklerin çözülmemiş halleri kullanılarak sonuca varılmaya çalışılmaktadır.

Biçimbirimsel Çözümleyici (Kemal Oflazer) : Kemal Oflazer'in çalışması, PC-KIMMO* programı kullanılarak, 23,000 kök sözcük içeren bir sözlük yardımıyla gerçekleştirilmiştir. Yirmi iki çeşit iki seviyeli biçimbirimsel kural tanımı yapılmış, eklerin geliş kuralları ise sonlu durum makineleri ile tanımlanmıştır.

Yazım Denetleyici için gerekli biçimbirimsel çözümleme için Kemal Oflazer ile iletişime geçilmiştir. Kullanılmak istenen eğitim kümeleri kendisine iletilmiş, biçimbirimsel çözümlemesi yapılmış dosyalar kendisi [35] tarafından gönderilmiştir.

Tez boyunca ve uygulamada bu çalışmadan “Biçimbirimsel Çözümleyici (Oflazer)” ifadesi kullanılarak bahsedilmektedir.

Elde edilmiş çözümleme sonuçları istenilen <kök+ek1+ek2+ek3....> biçimindedir.

Şekil 3.6'da sonuç dosyasının bir kısmı örneklenmiştir.

* PC-KIMMO: İki-seviyeli biçimbirimsel çözümleyici modeli. Bu model 1983 yılında Finli bilimadamı Kimmo Koskeniemi tarafından doktora tezi olarak ortaya çıkmıştır. Model, kelimenin yazımsal şekli için bir biçimsel şekil yaratma ve bu iki şekil arasında eşleştirme yapma temeline dayanır.

yakındoğu	yakındoğu
hakkındaki	hak+kı+nda+ki
hakkındaki	hak+kın+da+ki
bilgilerimizin	bilgi+ler+imiz+ın
çoğu	çoğ+u
çoğu	çoğu
son	son
yılda	yıl+da
arkeolojik	arkeoloji+k
alan	al+an
alan	ala+n
alan	alan
çalışmaları	çalış+ma+lar+ı
çalışmaları	çalış+ma+lar+ı
aracılığıyla	ara+cı+lığ+ı+yla
aracılığıyla	aracı+lığ+ı+yla
aracılığıyla	aracılığ+ı+yla
elde	el+de
elde	elde
edilmiştir	ed+il+miş+tir
yüzyılda	yüzyıl+da
arkeologları	arkeolog+lar+ı
arkeologları	arkeolog+lar+ı
ilgilendiren	ilgi+len+dir+en
ilgilendiren	ilgilen+dir+en

Şekil 3.6: BÇ (Oflazer) uygulamasının sonuç dosyası görünümü

Biçimbirimsel çözümleyicinin üretmiş olduğu sonuçlar yazım denetleyicinin verimini doğrudan etkilemektedir. Eğitim kümesi için yapılan çözümlemede yanlış üretilmiş sonuçlar gereksiz denetlemeler yapılmasına ve denetlemenin yanılmasına sebep olabilmektedir. Aynı şekilde yazım denetimi yapılacak olan metindeki sözcüklerin çözümlemesinde yanlış sonuçlar üretilmesi, hatasız olan bir kelimenin hatalı olarak algılanmasına ya da sonuç alınamayacak denetlemeler yapılmasına yol açabilmektedir. Bu yüzden yazım denetleyicinin verimi farklı biçimbirimsel çözümleyiciler kullanılarak test edilmiştir. Test sonuçları ile ilgili sayısal veriler Bölüm 3.5.7’de yer almaktadır.

3.5.3 Bulanık Mantık İle İki Sözcüğün Karşılaştırılması

Genel anlamda yazım hataları meydana geliş sebeplerine göre iki grupta incelenebilir:

- Yazımsal hatalar: Klavye kullanım hataları, yazım tanıma sırasında yapılan hatalar.

- Anlam hataları: Dilbilgisi hataları ve sözcüklerin yazımının yanlış bilinmesinden kaynaklanan hatalar.

Elektronik ortamlardaki metinlerde karşılaşılan yazımsal hataları şu şekilde sınıflandırabiliriz:

- İki harfin yerlerinin karışması : bilgisyaar
- Fazladan bir harf ekleme : bilgisayyar
- Harf silme, olması gereken bir harfin yazılmaması : bilgisaar
- Bir harf yerine farklı bir harf yazılması : bilgisayat

İnternet üzerindeki bir arama motorundan çıkarılmış istatistiklere göre [3] “albert einstein” ifadesinin belli bir zaman aralığında aranması sırasında karşılaşılmış yazımlar Tablo 3.4’de gösterilmektedir. Buradan da görüleceği gibi günlük hayatta, elektronik ortamlardaki yazımlar sırasında, yukarıda bahsedilen yazım hatalarına sıklıkla rastlanmaktadır.

Tablo 3.4: “albert einstein” ifadesinin aranması sırasında karşılaşılmış yazımlar

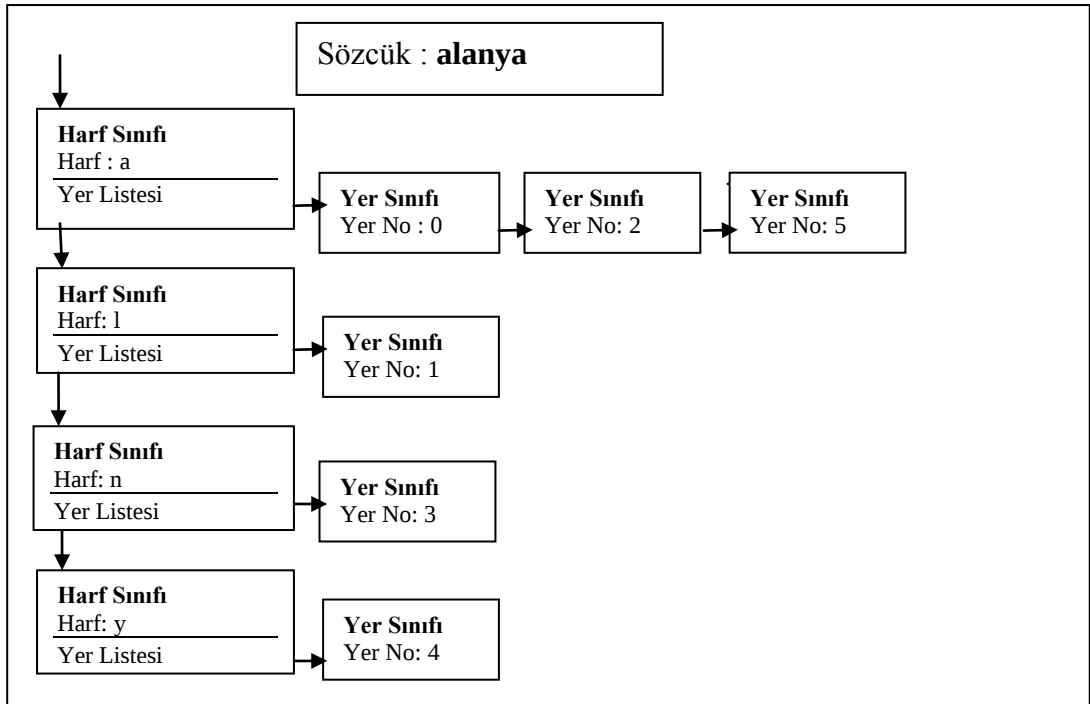
Yazım	Karşılaşılma Sayısı
albert einstein	4834
albert einstien	525
albert einstine	149
albert einsten	27
albert einsteins	25
albert einstain	11
albert einstin	10
albert eintein	9
albeart einstein	6
aolbert einstein	6
alber einstein	4
albert einseint	3
albert einsteirn	3
albert einsterin	3
albert eintien	3
alberto einstein	3
albrecht einstein	3
alvert einstein	3

Sözcük karşılaştırma yaklaşımlarından en yoğun kullanılanı en yakın uzaklık (minimum edit distance) yöntemidir. En yakın uzaklık değeri, bir sözcükten diğer bir sözcüğü elde etmek için yapılması gereken ekleme, silme, harflerin yerini değiştirme ve bir harf yerine başka bir harf yazma işlemleri sayısıdır. Bu tez çalışmasında en

yakın uzaklık yönteminden daha detaylı bir yaklaşım ele alınmak üzere bulanık mantık kavramı kullanılmaktadır.

Bulanık mantık ile sözcükleri karşılaştırmadaki temel mantık, sözcükler arası benzerlik oranını bulmaktır. Basit anlamda bu oran, iki sözcükteki tutan harflerin sayısının, sözcüğün harf sayısına oranıdır. Fakat bulanık mantık ile daha detaylı bir benzerlik oranı bulunmaya çalışılmaktadır. Tutan harfler yanında, yeri tutmayan harfler için de değerlendirme yapılmaktadır. İki harfin karışma olasılığı, bir harf yerine yanlış harf yazılma olasılığına göre daha fazladır. Özellikle hızlı yazılmış metinlerde bu tarz harf karışması durumlarıyla sıklıkla karşılaşılır. Bu bağlamda, yeri tutmayan harfler tamamen hatalı olarak kabul edilmemeli, benzerlik oranını olumlu yönde etkileyecek bir etkiye sahip olmalıdır.

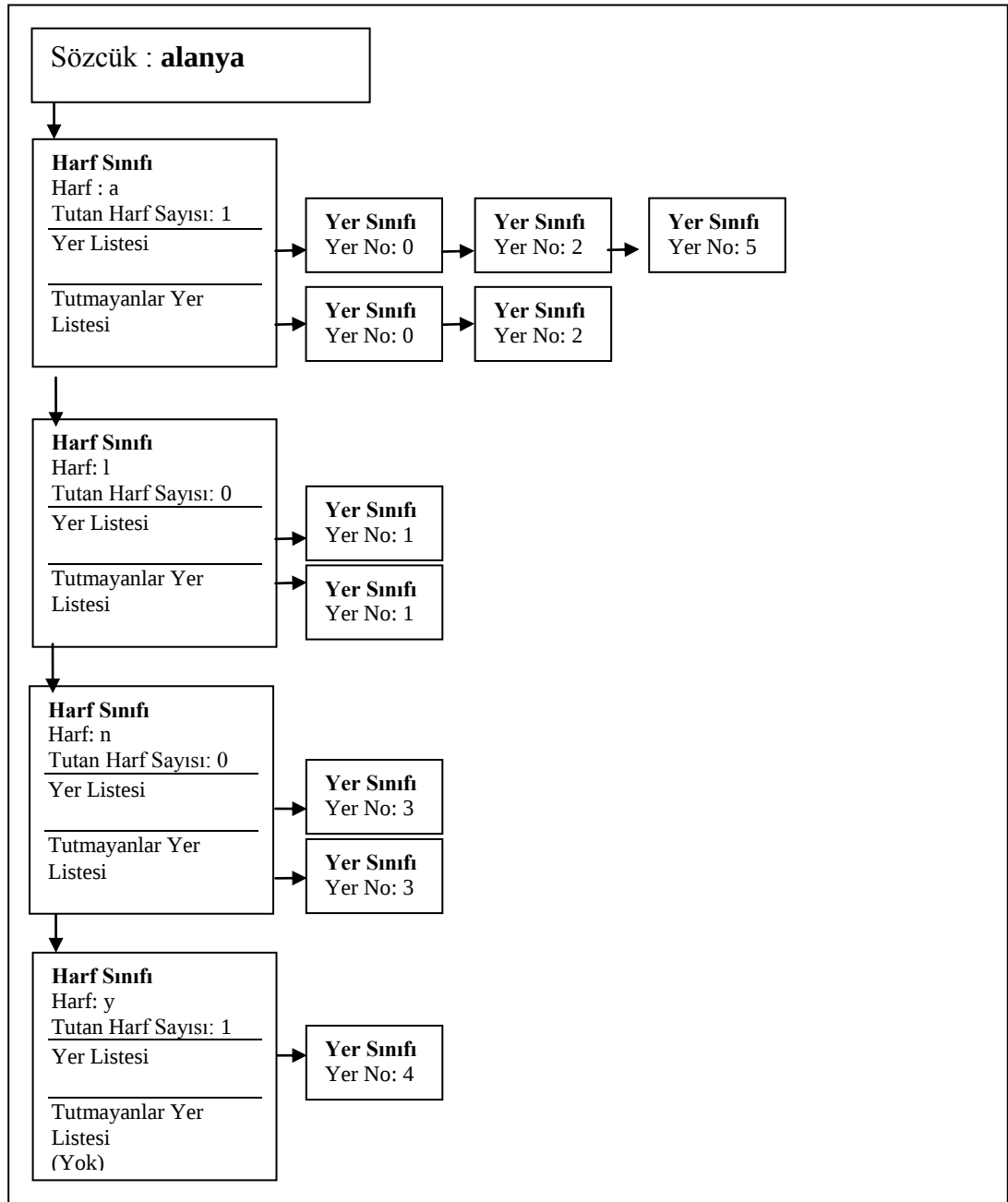
Bu yöntemi gerçekleyebilmek için öncelikle iki sözcükteki harflerin yerlerinin karşılaştırılması gerekmektedir. Bu yüzden ilk aşamada sözcükler içinde yer alan harfler için birer sınıf yaratılır. Bu sınıf içinde, harfin kelime içerisindeki yerleri belirlenir.



Şekil 3.7: Harf ve Yer sınıfları örneği

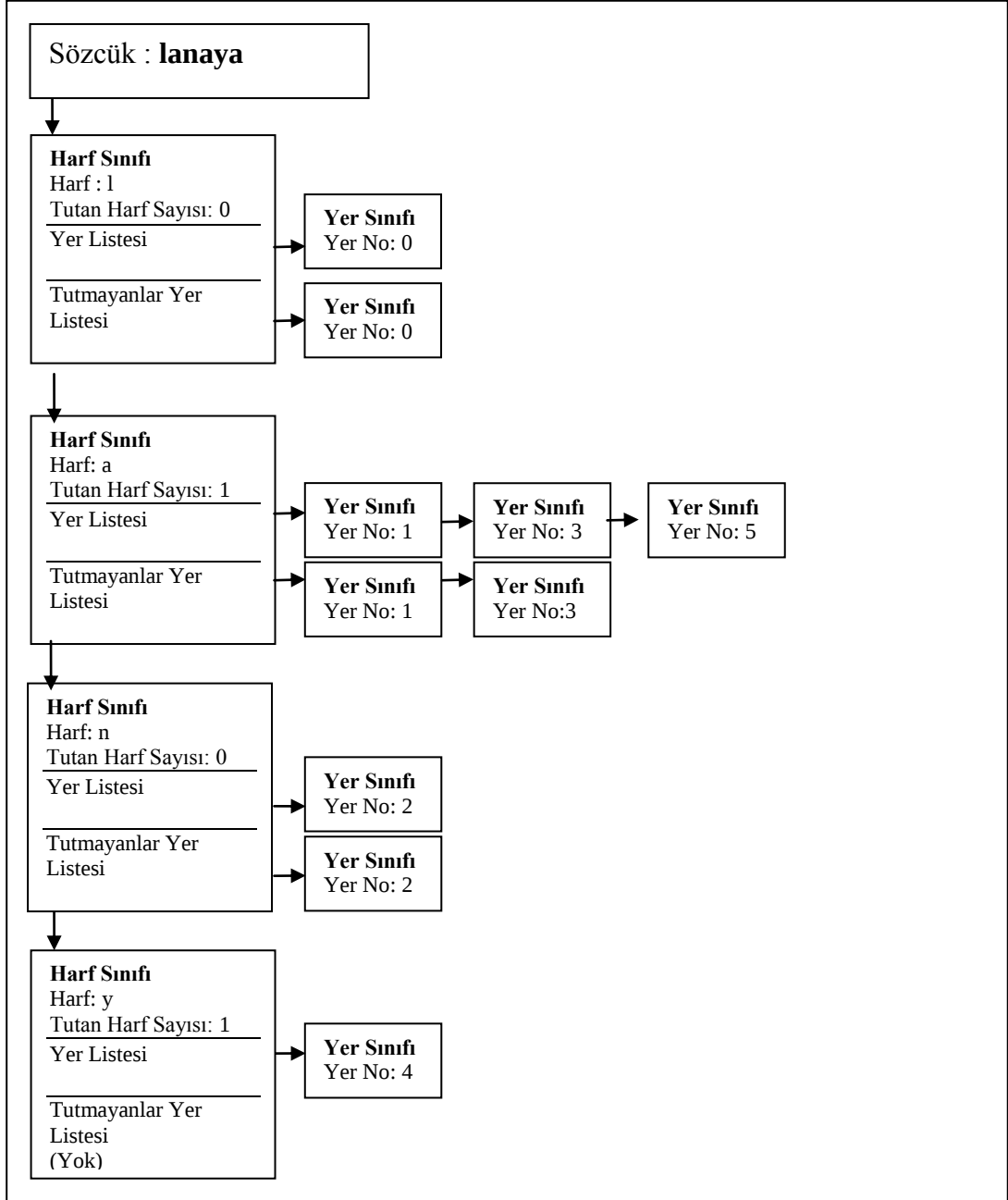
Şekil 3.7’de “alanya” sözcüğü için, harfler a, l, n ve y olarak belirlenmiş, her harf için bir sınıf oluşturulmuştur. Sözcükteki her harfin sözcük içerisinde bulunduğu

yerler belirlenip, o harfe ait yer sınıflarına atanmıştır. Karşılaştırılacak her iki sözcük için de bu işlemler yapıldıktan sonra aynı harfler için yer karşılaştırmaları yapılır. Her iki sözcükte de aynı yerde bulunan harf, tutan harfler sayısını bir arttırır. Elde edilen değer, harf sınıfında, “tutan harf sayısı” alanına atanır. İki sözcükte de yer alan fakat farklı yerlerde bulunan harfler, yani yeri tutmayan harfler için de bulundukları yerleri belirtecek yer sınıfları oluşturulur. Bu yer sınıfları ileride yeri tutmayan harfler arası uzaklıkların bulunmasında kullanılacaktır.



Şekil 3.8: Örnekte ilk sözcük için oluşmuş Harf ve Yer sınıfları

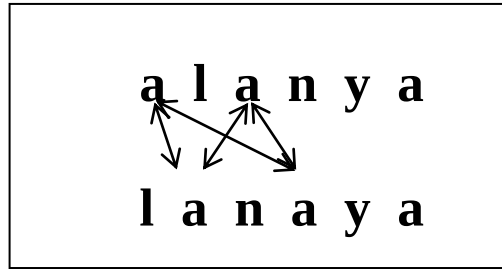
Şekil 3.8’de “alanya” ve Şekil 3.9’da “lanaya” sözcüklerinin karşılaştırılması sırasında oluşturulmuş yapıları yer almaktadır. Yeri Tutmayanlar yer listesi, tek tek harflerin her iki sözcükteki yerleri karşılaştırılıp, aynı yerde olmayan harflere rastlanınca oluşturulur.



Şekil 3.9: Örnekte ikinci sözcük için oluşmuş Harf ve Yer sınıfları

Yeri Tutmayan harfler için uzaklık değerleri bulunur. Bu değer, harfin iki kelimedeki bulunduğu yer numaraları arasındaki farktır. Yeri Tutmayan harfler için, harf sınıfı içinde ayrı bir Yeri Tutmayanlar Puan Listesi sınıfı oluşturulur. Bu sınıfı oluşturmak

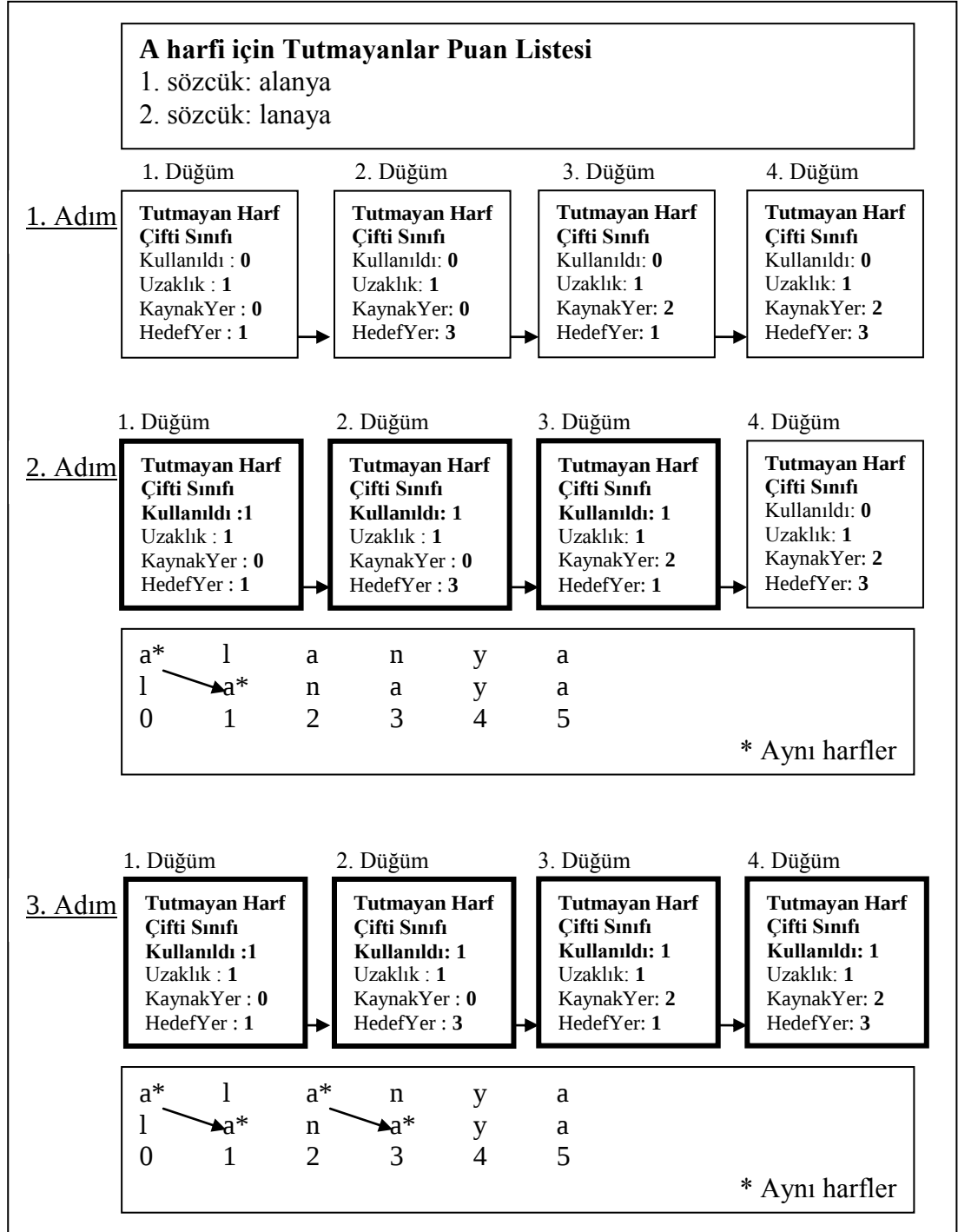
için, yeri tutmayanlar yer listesi sınıfından faydalanır. Bir harf için, her iki sözcükteki yeri tutmayanlar yer listesindeki yerleri alınıp, bunlar arasındaki fark bulunur. Bu fark uzaklığı verir. Yeri Tutmayanlar Puan Sınıfındaki uzaklık alanına bu değer atanır. Yeri Tutmayanlar Puan Listesi, uzaklığa göre azalan sırada oluşturulur. Her yeni Yeri Tutmayan Puan Sınıfı, listeye uygun sırada eklenir. Şekil 3.11’de “a” harfi için oluşturulmuş, Yeri Tutmayanlar Puan Listesi görülmektedir. Karşılaştırılacak “alanya” ve “lanaya” sözcüklerinde “a” harfi dört yerde tutmamaktadır. Şekil 3.10’da “a” için yeri tutmayan harf olası durumları belirtilmiştir.



Şekil 3.10: Örnek için yeri tutmayan harf olası durumları

1. alanya’nın 0. ve lanaya’nın 1. harfinde
2. alanya’nın 0. ve lanaya’nın 3. harfinde
3. alanya’nın 2. ve lanaya’nın 1. harfinde
4. alanya’nın 2. ve lanaya’nın 3. harfinde

Bu olası dört durum için dört adet Yeri Tutmayan Puan Sınıfı oluşturulmuştur. Her sınıfta, Kaynak Yer alanına, harfin 1. sözcükteki yeri, Hedef Yer alanına da harfin 2. sözcükteki yeri atanmıştır. Yeri Tutmayan Harf listesinden faydalanılarak oluşturulmuş, uzaklık değeri de Uzaklık alanına atanmıştır. Şekil 3.11, 1. Adımda atanan değerler görülmektedir. Puanlama sırasında bir harf için yeri tutmayan harf olasılıklarından sadece en uygunu için puan hesabı yapılmalıdır. Bu bağlamda, Yeri Tutmayan Puan Sınıfı içerisinde hesaplamalar sırasında ele alınmış bir harfin diğer olasılıkları için de fazladan işlem yapmamak için “Kullanıldı” isimli bir alan tutulmuştur. Bu alanla ilgili ayrıntılara puan hesaplamanın detayında değinilecektir. Bu yapı oluşturma işlemleri ve atamalar ile puan hesaplama ile ilgili altyapı hazır durumdadır.



Şekil 3.11: Örnek Yeri Tutmayanlar Puan Listesi görünümü

Yeri Tutmayan puanı hesaplanırken yeri tutmayan harf listesindeki her düğüm sırası ile incelenir. Şekil 3.11, 1. Adımda görüldüğü gibi uzaklığa göre sıralanmış düğümler için ilk düğüm en uygun yeri tutmayan harf olasılığını belirtmektedir. İlk düğüm için puan hesaplandıktan sonra, diğer olasılıkların artık kullanılmaması gerekmektedir. Bu yüzden puan hesabında kullanılan düğümün ve bu düğüm ile aynı

Hedef ve Kaynak yerlerine sahip düğümlerin “Kullanıldı” alanına 1 atanır. Bu, bu düğümlerin puan hesaplamada artık kullanılmayacak olduğunu belirtir. Şekil 3.11, 2. Adımda puan hesaplaması için en uygun düğüm olan 1. düğümün, 1. düğüm ile aynı Kaynak Yer’ine sahip olan 2. düğümün ve 1. düğüm ile aynı Hedef Yer’ine sahip olan 3. düğümün Kullanıldı alanlarına 1 atanmıştır. Yeri Tutmayan Puan Listesinde ilerlemeye devam edilince Kullanıldı alanı 0 olan, yani puanlama için henüz kullanılmamış sıradaki düğüm, 4. Düğüm olarak gelmektedir. Bu düğüm değerleri de kullanılarak puan hesabı yapıldıktan sonra Şekil 3.11, 3. Adımda görüldüğü gibi 4. Düğümün de Kullanıldı alanına 1 atanmaktadır.

Sözcükleri Bulanık Mantık ile karşılaştırırken kullanılacak puanlama formülü:

$$BMP = \frac{1}{H_s} (TH_s + \sum YTM_P) \quad (3.2)$$

$$\sum YTM_P = \sum \frac{1}{U + k} \quad (3.3)$$

- H_s : Karşılaştırılan sözcüklerden uzun olanının harf sayısı
- TH_s : İki sözcükteki tutan harf sayısı
- $\sum YTM_P$: Her harf için yeri tutmayan puanı (3.3) değerleri toplamı
- U : Yeri Tutmayan harflerin her iki sözcükteki yerleri arası uzaklığı. Bu uzaklık yeri tutmayan harf sınıfı içerisinde hesaplanmış olarak tutulmaktadır.
- k : Sabit değer. Değeri 0.2’dir.

Şekil 3.11’de gösterilen tüm bu işlemlerin matematiksel karşılıkları gösterilecek olunursa:

Karşılaştırılacak sözcükler: alanya - lanaya

$$\begin{aligned} \text{Bulanık Mantık Puanı : } BMP &= \frac{1}{H_s} (TH_s + \sum YTM_P) \\ &= \frac{1}{6} (2 + \sum YTM_P) \end{aligned}$$

$$\sum YTM_P = \sum \frac{1}{U + k} = 3,3333$$

$$\text{“a” için : } \frac{1}{1 + 0,2} = 0,8333 \text{ (Şekil 3.11’de 1. Düğümden elde edilmiş puan)}$$

$$\text{"l" için : } \frac{1}{1+0,2} = 0,8333$$

$$\text{"a" için : } \frac{1}{1+0,2} = 0,8333$$

$$\text{"n" için : } \frac{1}{1+0,2} = 0,8333$$

$$\rightarrow \sum TmP = 3,3333$$

Bulanık Mantık Puanı : $BMP = \frac{1}{6}(2 + 3,3333) = 0,8888$ olarak bulunur.

Tablo 3.5’de değişik örneklerin Bulanık Mantık Puanlaması görülmektedir.

Tablo 3.5: Bulanık Mantık Puanlaması örnekleri

1. Sözcük	2. Sözcük	Puan
merhaba	mehraba	0,952381
gül	gul	0,666667
iskandinavya	iksandinavya	0,972222
iskandinavya	iksandinayva	0,944444
sözcük	söczük	0,944445

Karşılaştırmalar sırasında elde edilen bulanık mantık puanı, belirlenmiş bir seviyeden yüksek değil ise işleme katılmaz. Bu seviye eşik değeridir.

Harf karıştırılması sonucu oluşan yazım hatalarında bulanık mantık puanlaması çok iyi bir verimlilikte sonuç vermektedir. Sıkça karşılaşılan hata türlerinden biri olan harf karıştırılması bulanık mantık ile çözümlenmektedir.

Harf eklenmesi ve eksik harf yazılması sonucu oluşan yazım hataları için de bulanık mantık puanlaması devreye girmektedir. Yalnız burada dikkat çeken bir konu, harf eklemesi ve eksik yazılması sırasında, sözcükteki harf yerlerinin kaymış olmasıdır. Yapılan bir araştırmaya göre [4] güncel hayatta yazım hataları, büyük bir olasılıkla kelimenin sonuna doğru yapılmaktadır. Bu sonuca dayanarak, harflere soldan sağa doğru yer numarası verilmektedir. Aynı zamanda harf eklenmesi ya da eksik harf hatalarından kaynaklanacak yer kayması, hatanın kelimenin sonuna yakın oluşuyla doğru orantılı olarak puanlamayı etkileyecektir. Yani hata ne kadar sonda olursa, yeri kayan harf sayısı o kadar az, dolayısıyla benzerlik olasılığı o kadar yüksek olacaktır. Yalnız burada yapılabilecek bir iyileştirme, kelimeye bir de sondan başa doğru yer numarası vermek, böylece tersten de bulanık mantık puanını bulmak şeklinde olabilmektedir. Bu amaçla, kullanıcının seçimi doğrultusunda, düz bulanık mantık

puanı bulma yanında bir de tersten bulanık mantık puanı bulunabilmekte, yüksek olanı, sonuç bulanık mantık puanı olarak kabul edilmektedir. Böylece hatanın sonda ya da başta yapılmış olması puanlamayı olumsuz yönde etkilememiş olacaktır.

Tablo 3.6’da eksik/fazla harf ve yerleri karışmış harf hatalarına değişik örnekler verilmiş olup, düz bulanık mantık puanı ve ters bulanık mantık puanları karşılaştırma amaçlı belirtilmiştir.

Tablo 3.6: Düz BMP ve Ters BMP örnekleri

Hata Türü	1. Sözcük	2. Sözcük	Düz BMP	Ters BMP
1 harf eksik	alanya	lanya	0,555556	0,833333
1 harf eksik	alanya	alaya	0,777778	0,611111
1 harf eksik	alanya	alany	0,833333	0,631313
1 harf fazla	alanya	alatnya	0,785714	0,666667
1 harf fazla	alanya	talanya	0,476191	0,857143
1 harf fazla	alanya	alanyat	0,857143	0,476191
harflerin yeri karışmış	alanya	alayna	0,944444	0,944444
harflerin yeri karışmış	alanya	alanay	0,944444	0,944445
harflerin yeri karışmış	alanya	laanya	0,944445	0,944444

Tablo 3.6’daki puan değerlerine göre yorum yapılırsa, kelimenin sonunda yapılan hatalarda, yer numaralama işlemi soldan sağa yapılan düz bulanık mantık puanı daha yüksek çıkmaktadır. Çünkü yapılan hatanın sonda olması yeri kayan harf sayısının daha az olması anlamına gelmektedir. Tersine sözcüğün baş tarafında yapılmış bir hatada ise, ters bulanık mantık puanı daha yüksek çıkmaktadır. Bu durumda da sağdan sola yapılmış yer numaralama işlemi sözcüğün başında yapılan hatalarda etkilenen, yeri kayan harf sayısını en aza indirmiştir.

Bulanık mantık, uzun kelimeler için daha sağlıklı sonuçlar üretmektedir. Çünkü uzun kelimelerde yapılacak hata sayısı kısa kelimelere göre daha azdır [4]. Örneğin ikiş harfinin yeri karışmış, “ali“ ile “ail” kelimeleri için hesaplanan bulanık mantık puanı 0.888889 iken “alanyalı” ile “alanyalı” kelimeleri için hesaplanan bulanık mantık puanı 0,958333’dir.

Kökler ele alınırken, hatanın sözcüğün ortasında olduğu durumlar için bir iyileştirme yapılmıştır. Bu iyileştirme kapsamında öncelikle hatanın sözcüğün ortasında olup olmadığı taranmıştır. Bu amaçla sözcük 3 eşit kısma ayrılmış, KKK’de hatalı sözcüğün ilk kısım ile başlayan, son kısmı ile biten sözcükler taranmıştır. Listelenen bu kökler ile hatalı sözcüğün orta kısımları karşılaştırılmış ve benzerlik oranına göre aday önerme bulunmaya çalışılmıştır. Buradaki benzerlik

oranı bahsedilen bulanık mantık puanlamasından daha basit bir orandır. İki sözcüğün alt kısmında da yer alan harflerin, uzun olan sözcüğün alt kısmına oranı şeklindedir. Bu oran belli bir seviyenin üstünde ise sözcük önerme olarak kabul edilebilir.

Örneğin; “psikloji” hatalı kökü için “psi” ile başlayıp “oji” ile biten kökler listelenmiştir. Listelenen kökler arasında “psikoloji”, “psikomitoloji”, “psikopatoloji” vb. gelmektedir. Hatalı sözcüğün orta kısmı olan “kl”, listelenen sözcüklerin orta kısımları olan “kol”, “komitol”, “kopatol” ile karşılaştırılır. En yüksek benzerlik oranı “kol” için çıkmakta ve önerilen kök, “psikoloji” olmaktadır.

3.5.4 Öğrenme Aşaması

Yazım hatalarının bir bölümü bir harf yerine yanlış bir harf yazılmasından kaynaklanır. Bu tür durumlar, bulanık mantık puanlaması ile ele alınmaz. Çünkü bahsedildiği gibi bulanık mantık puanlaması her iki kelimedede de yer alan (yerleri tutan ya da tutmayan) harfler üzerinden çalışır. Bir harf yerine farklı bir harf yazıldığı durumlarda ise hem bu tür hatalara sebep olacak durumlar değerlendirilmeli hem de özel bir sebep kaynağı olmayan fakat sıkça yapılan hatalara karşı sistemin duyarlı davranması sağlanmalıdır.

Harf karıştırmasına sebep olacak durumlar düşünüldüğünde, ilk akla gelen klavyede harflerin yakın yer alması ya da kağıttan taranmış bir metinde yer alan yazımı birbirine benzeyen harflerdir. Bu sebeplerden yola çıkarak, denetleme işlemi, klavye yakınlığı ve yazım benzerliğine göre daha akıllı hale getirilebilir.

Belli bir sebebe bağlı olmaksızın, kişiye özel hatalarla da sıkça karşılaşılmaktadır. Örneğin kişinin bir sözcüğün yazımını yanlış bilmesi ya da bazı harfleri alışkanlık halinde yanlış yazması bu tür hatalardandır. Hangi harflerin ne şekilde yanlış yazıldığından öte, sistemin bu tür sıkça yapılan hataları biliyor ve buna göre davranıyor olması, yazım denetleme işini çok daha akıllı hale getirebilmektedir. Bu durum sistemin yapılan hataları öğrenmesi ve öğrendiği bilgiler ışığında hareket etmesidir.

Bu amaçla insanın öğrenme işlevini modellemek üzere tasarlanmış Yapay Sinir Ağları kullanılmaktadır. Uygulamada kullanılan yapay sinir ağı, tek kademeli, denetimsiz (unsupervised) bir yapay sinir ağıdır. Her denetim sırasında karşılaşılan hatalar ile, öğrenilecek bilgiler yapısı oluşturulur. Denetleme bittikten sonra

hatalardan derlenmiş bu yapı kullanılarak sistemin ağırlıkları yeniden hesaplanıp, güncellenir. Böylece asıl öğrenme işlemi gerçekleşmiş olur.

Öğrenme yapısı 3 aşamada incelenebilir:

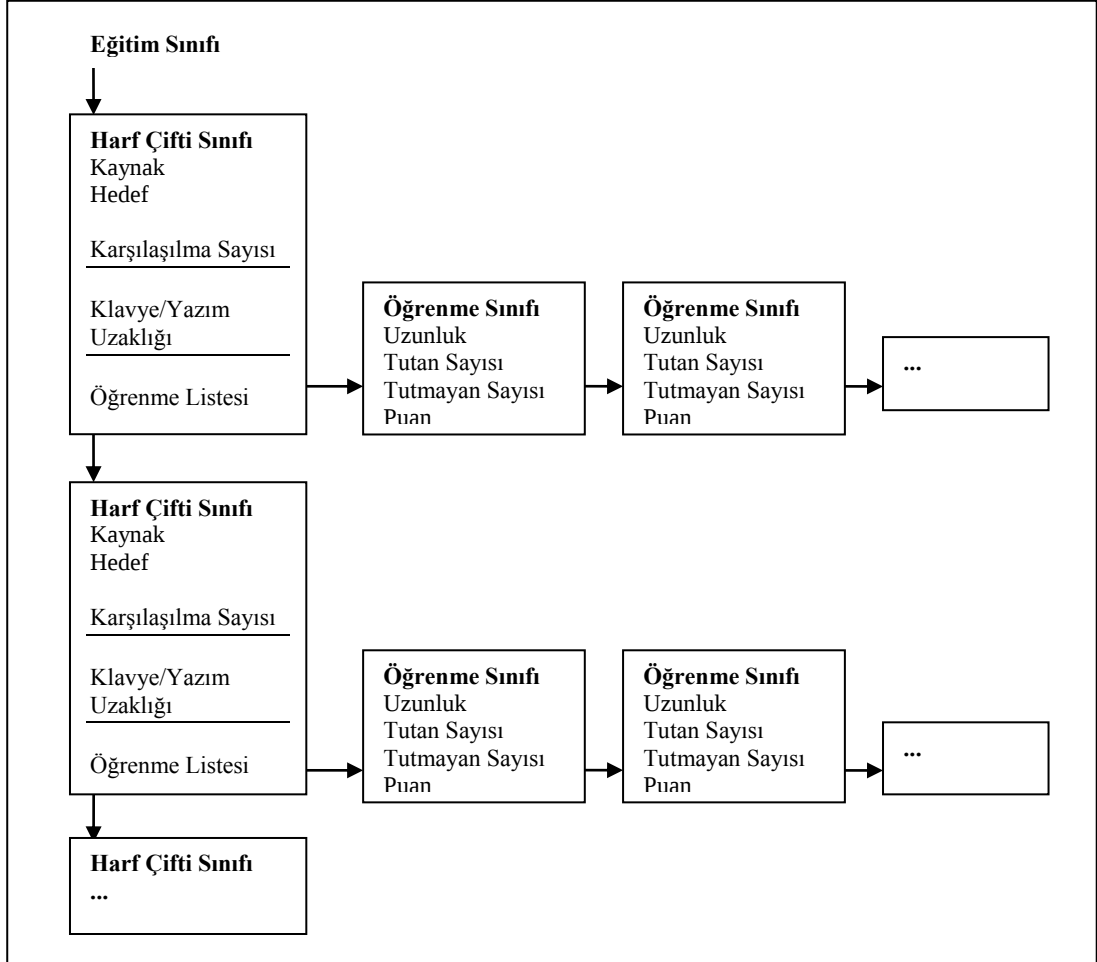
- Öğrenme Öncesi Hazırlık: Denetleme sırasında karşılaşılan hataların öğrenilmesi için öğrenilecek bilgilerin hazırlanması.
- Öğrenme: Hazırlanmış öğrenilecek hata bilgileri ile öğrenmenin gerçekleşmesi.
- Öğrenme Puanının Hesaplanması: Öğrenilmiş bilgiler kullanılarak öğrenme puanının hesaplanması.

3.5.4.1 Öğrenme Öncesi Hazırlık

Hatalı işlem, harflerin yeri karıştırılmış, eksik/fazla yazılmış olsa da harf anlamında bakıldığında, bir harf yerine farklı bir yazılmış olmasıdır. Sonuçta karşılaştırma açısından görünen, olması gereken bir harf yerinde farklı bir harfin yer alıyor olmasıdır. Bu bağlamda, öğrenme işlemi birbiri yerine yazılmış harfler üzerinden yapılmaktadır. Bu amaçla her karıştırılmış harf çifti için – denetleme işlemi sırasında – bir Harf Çifti sınıfı oluşturulmaktadır. Şekil 3.12’de de görüldüğü gibi bu sınıf içerisinde harf çiftini belirtmek üzere Kaynak ve Hedef Harf alanları, bu harf çiftinin şimdiye kadarki birbirinin yerine yazılma sayısını belirtmek üzere Karşılaşma Sayısı alanı, bu harf ikilisinin klavye ve yazım yakınlığını belirtme amaçlı Klavye/Yazım Uzaklığı alanı ve bu harf çifti için öğrenilmiş bilgiyi tutmak üzere Öğrenme Listesi bulunmaktadır. Her denetim sırasında yeni karşılaşılmış bir karışmış harf çifti için Harf Çifti sınıfı oluşturulmaktadır.

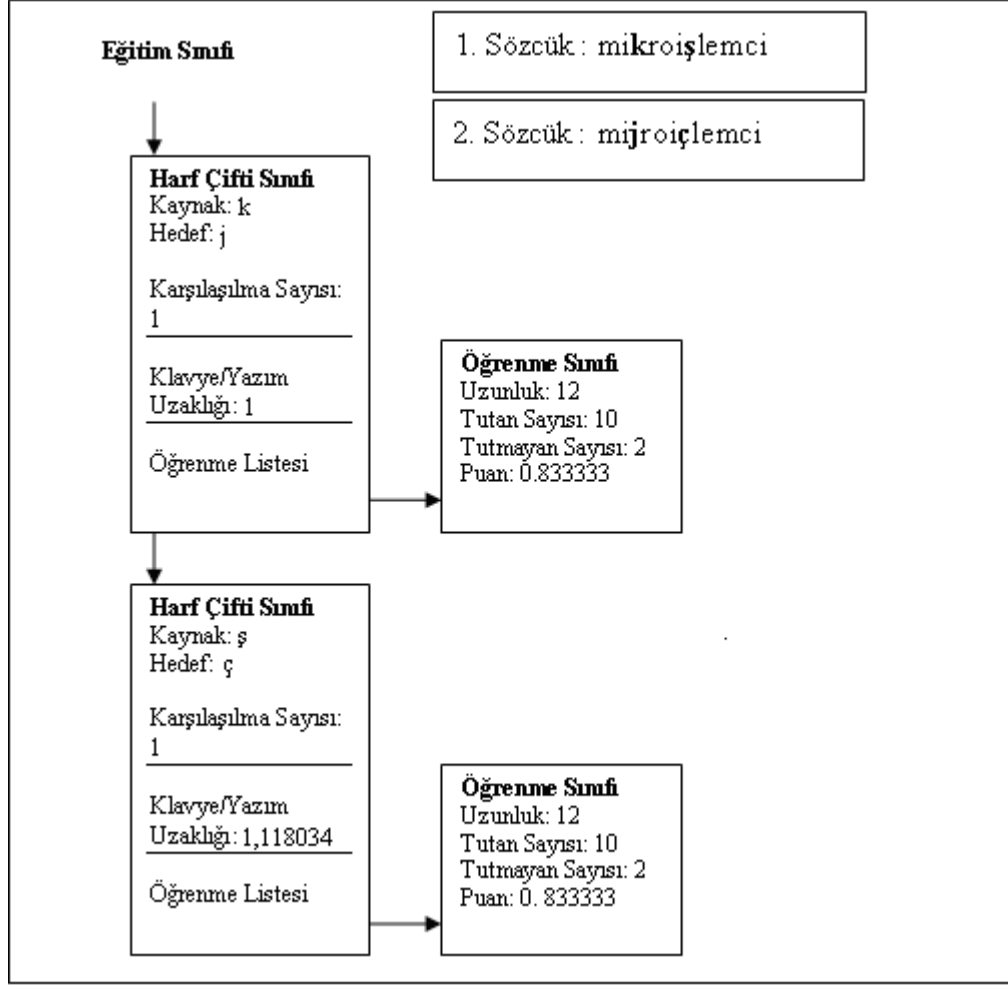
Her harf çifti için öğrenilecek bilgileri tutmak üzere, Öğrenme sınıfı kullanılmaktadır. Şekil 3.12’de görüldüğü gibi bu sınıfta, karşılaşılan ve öğrenilecek olan hatanın kaynağı iki sözcükle ilgili değerler tutulmaktadır. Karşılaştırılan iki sözcükten uzun olanının harf sayısı Uzunluk alanında, karşılaştırma sırasında hesaplanmış, tutan ve tutmayan harf sayıları, sırasıyla Tutan Sayısı ve Tutmayan Sayısı alanlarında tutulmaktadır. Ayrıca bu karşılaştırmadan elde edilmiş puan da bu sınıf içerisinde Puan alanında saklanmaktadır. Bir karıştırılmış harf çiftine her rastlandığında, yeni bir Öğrenme sınıfı oluşturulup, bu harf çiftine ait Öğrenme Listesine eklenmektedir. Öğrenme Listesinde Öğrenme sınıfları puana göre azalan

sırada tutulmaktadır. Denetleme ardından öğrenilecek bilgilerden kurulan öğrenme yapısı kullanılarak, sistemin ağırlıkları güncellenir, yani sistem öğrenir. Bu işlem sırasında, sistemin ağırlığını güncellemek için puanlara erişmek açısından sınıfların puan sıralı şekilde tutulması önemlidir.



Şekil 3.12: Öğrenme sırasından tutulan yapının görünümü

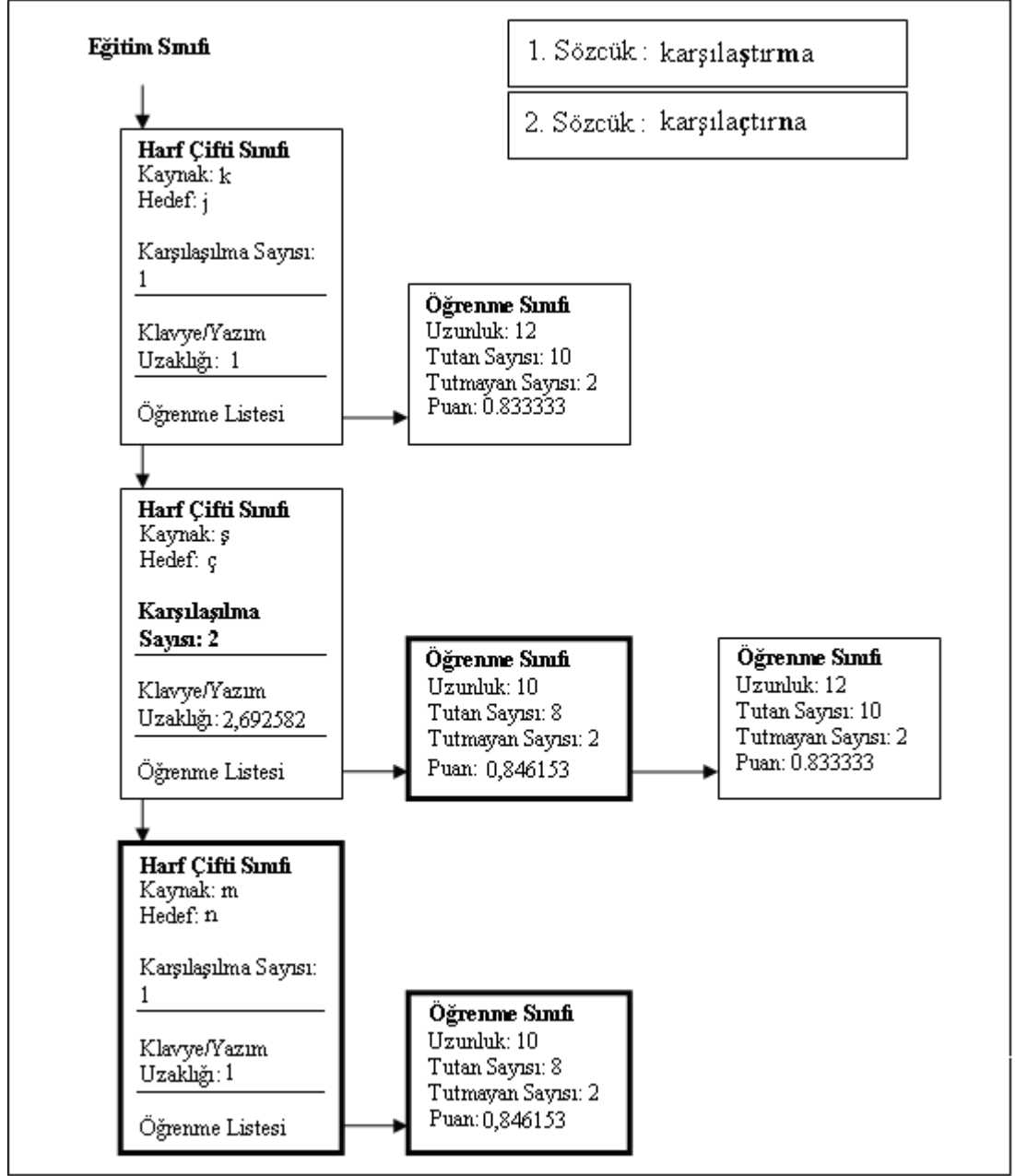
Şekil 3.13’de sistem tarafından bulunmuş ilk hatalı sözcük sonrası elde edilmiş yapı örneklenmektedir. Hatalı kelime “mijroiçlemci”, en yüksek puan ile önerilen kelime ise “mikroişlemci” önermesidir. Karıştırılmış harf çiftleri k-j ve ş-ç için iki tane Harf Çifti sınıfı oluşturulmuş, alanlara ilgili değerler atanmıştır. Bu karşılaştırmadan öğrenilen bilgileri tutmak amaçlı, bu Harf Çifti sınıflarına birer tane Öğrenme sınıfı eklenmiştir. Öğrenme sınıflarının alanlarına, bu karşılaştırmadan elde edilmiş, puan ve diğer sayısal değerler atanmıştır.



Şekil 3.13: Örnekte oluşan Öğrenme yapısı görünümü

Şekil 3.14’de örneklendiği üzere, bir sonraki denetlemede, karşılaşılan hatalı sözcük “karşılaştırna”, önerilen en yüksek puanlı sözcük ise “karşılaştırma”dır. Karıştırılmış harf çiftlerinden ş-ç için bir Harf Çifti sınıfı bulunduğu için, bu harf çifti için karşılaşılma sayısı bir arttırılıp, Öğrenme listesine bu karşılaştırmadan elde edilmiş değerlerden oluşmuş yeni bir Öğrenme sınıfı eklenmiştir. Yeni öğrenme sınıfı daha yüksek puana sahip olduğu için, Öğrenme Listesinde diğer sınıftan daha önde bir yere sahiptir. Diğer harf çifti olan m-n için yeni bir Harf Çifti sınıfı oluşturulmuş, Öğrenme Listesine bu karşılaştırmaya ait değerlere sahip bir Öğrenme sınıfı eklenmiştir. Yeni eklenen sınıflar Şekil 3.14’de koyu çerçeveli sınıf gösterimleri ile belirtilmiştir.

Bu şekilde her denetleme sonrası öğrenilecek yeni bilgiler hazırlanmaktadır.

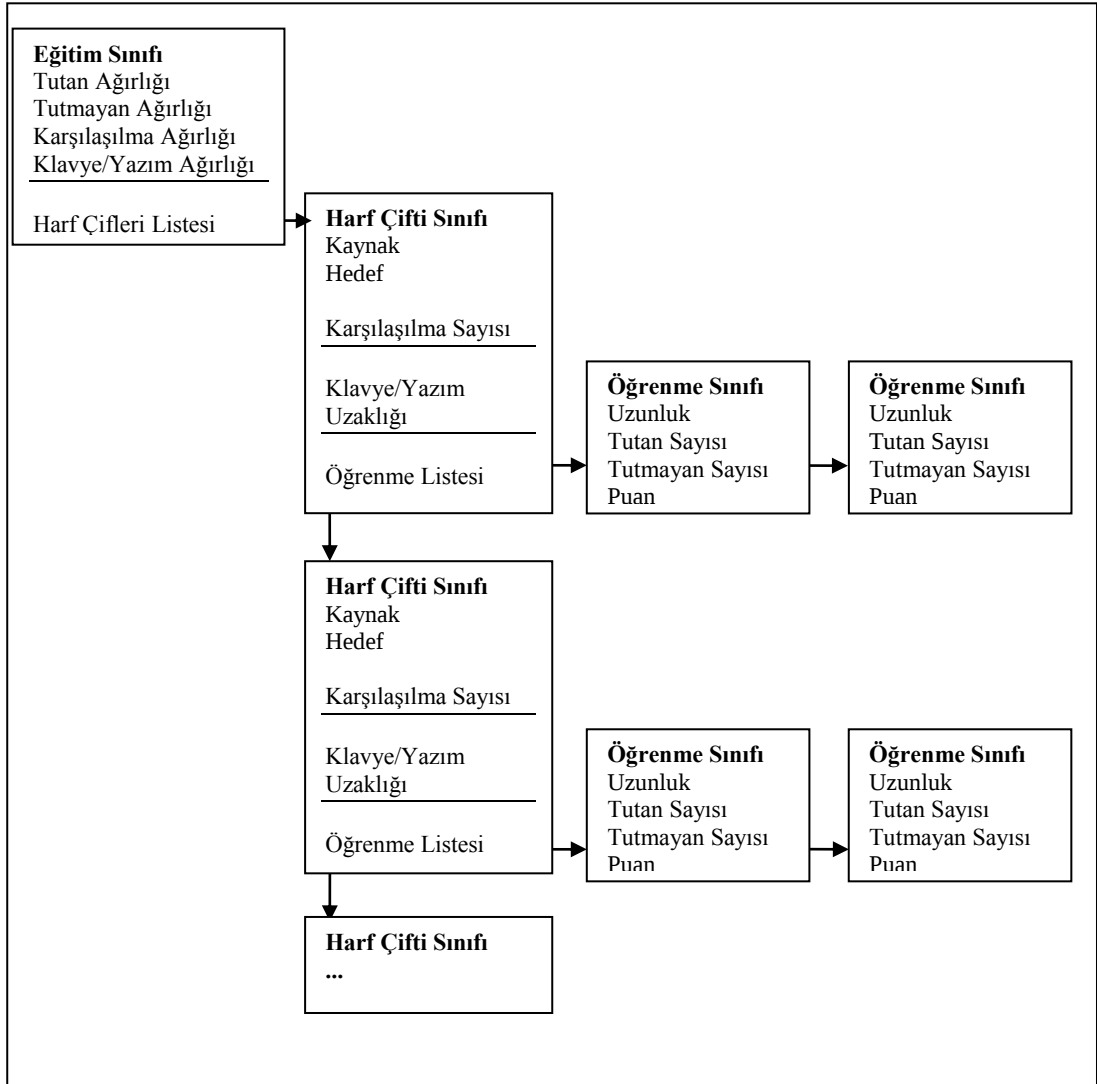


Şekil 3.14: Örnekte oluşan Öğrenme yapısı görünümü

Denetleme işlemi bittikten sonra öğrenilmek üzere hazırlanmış yapı kullanılarak sistemin ağırlıkları güncellenir. Elde edilen bu ağırlıklar daha sonra öğrenme puanını hesaplama aşamasında kullanılacaktır.

Ağırlıklar ve ağırlıkları güncellemek üzere denetleme sırasında oluşturulmuş öğrenilecek hatalar yapısı, sistemin geneline ait bir Eğitim sınıfı yapısında tutulmaktadır. Öğrenme sonrasında elde edilmiş bu ağırlıklar, Eğitim sınıfı içindeki karşılık gelen ağırlık alanlarında tutulur. Denetim sırasında öğrenilecek bilgileri

tutmak üzere yaratılmış Harf Çifti sınıfları da Eğitim sınıfının Harf Çiftleri listesini oluşturmaktadır. Böylece sisteme ait büyük resim Şekil 3.15’de görüldüğü gibidir.



Şekil 3.15: Öğrenme yapısının genel görünümü

3.5.4.2 Öğrenme

Öğrenme işlemi, denetim sırasında ortaya çıkan hatalar kullanılarak oluşturulmuş, öğrenilecekler bilgileri tutan yapı kullanılarak gerçekleştirilir.

Sistemin 4 çeşit ağırlığı bulunmaktadır.

1. Tutan Harflerin Ağırlığı
2. Tutmayan Harflerin Ağırlığı
3. Hatanın Karşılaşılma Ağırlığı

4. Klavye/Yazım Ağırlığı

Bu ağırlıklar sistemin o ana kadar olan yapısının da bir göstergesidir. Örneğin klavye/yazım ağırlığı, diğer ağırlıklara göre yüksek ise, sistemde o ana kadar yapılmış hataların büyük çoğunluğu, klavye/yazım hatasından kaynaklanıyor demektir. Böylece sistem bu tarz hatalarla daha yoğun karşılaşılabilceğinin bilincinde olmaktadır.

Sistem bu 4 ağırlık üzerinden öğrenme puanını elde eder. Her öğrenme puanı elde edilmesi sırasında, o anki karşılaştırma değerleri ve sistemin genel değerleri (ağırlıkları) kullanılır. Bu yapının matematiksel gösterimi aşağıdaki formül ile belirtilmiştir:

$$OgP = (W_{tutan} x1) + (W_{tutmayan} x2) + \sum (W_{geçme\ sayısı} x3) + \sum (W_{klavye/yazım} x4) + \varepsilon \quad (3.4)$$

Formülde yer alan değişkenlerle ilgili açıklamalar aşağıdadır. W ile başlayan değişkenler sistemin genel ağırlıkları, x ile başlayanlar ise o anki karşılaştırma ile ilgili değerleri içeren değişkenlerdir.

W_{tutan} : Tutan harf ağırlığı

$W_{tutmayan}$: Tutmayan harf ağırlığı

$W_{geçme\ sayısı}$: Hata ile karşılaşılma sayısından kaynaklanan ağırlığı

$W_{klavye/yazım}$: Harflerin klavyedeki uzaklıkları ve yazım benzerliği ile ilgili ağırlık

$x1$: tutan harf sayısı / sözcüklerden uzun olanının boyu

$x2$: tutmayan harf sayısı / sözcüklerden uzun olanının boyu

$x3$: tutmayan harf ikilisinin karşılaşılma sayısı / eğitim sınıfındaki toplam harf ikilisi sayısı (bir harf ikilisinin tüm karşılaşılma sayıları da bu toplama dahildir)

$x4$: harflerin klavyedeki uzaklıkları ya da benzerlikleri / klavyedeki en büyük uzaklık (Klavyede köşeden köşeye olan uzaklık)

ε : hata payı (1/10000)

Formülün tutan harf ve tutmayan harflerle ilgili alt kısımları, karşılaştırma yapılan iki sözcüğün geneline ait değerler iken diğer kısımlar karşılaştırılan sözcük içinde

rastlanmış hatalı harf çiftleri ile ilgilidir. Bu yüzden formülde $\sum$ işaretinin geçtiği kısımlarda, bu sözcüklerdeki her hatalı harf çifti için karşılaşılma sayısı ve klavye uzaklığı ile ilgili alt puanlar bulunmakta ve sonuç puana, bunların toplamaları şeklinde yansıtılmaktadır. Fakat sistemin ağırlıkları bulunurken sözcük anlamında değil, hatalı harf çifti anlamında hareket edilmektedir.

Öğrenme Puanı Formülündeki değişkenlerin öğrenme puanını hesaplama sırasındaki kullanımlarına Bölüm 3.5.4.3’de değinilecektir. Öğrenme işlemi sırasında sistemin yeni ağırlıkları bulunurken öğrenme puanı hesaplama formülü (3.4) kullanılmaktadır. Öğrenme Puanı formülüne, Eğitim sınıfından (denetim sırasında elde edilen öğrenilecekler listesi) seçilmiş bir Öğrenme düğümüne ait puan ve değerler konularak, ağırlıklar elde edilmeye çalışılır. Her denetleme işlemi sonucunda sistemde karşılaşılan hata oranı ve türü değişmektedir. Dolayısıyla sistemin ağırlıkları da değiştirilmelidir. Karşılaşılan toplam hata sayısı sistemin ağırlıklarını belirlemede en büyük rolü oynamaktadır. Yeni denetlemeler sonucu elde edilen değerler ile sistemin genel ağırlıkları tekrar hesaplanır.

Öğrenme formülündeki 4 adet ağırlık değerini bulmak için, dört bilinmeyenli denklemi çözmek gerekmektedir. Bu problemi kolaylaştırmak için denklemleri daha sadeleştirip, kısım kısım çözmeye yoluna gidilmiştir. Bu bağlamda, denklemlerin iki bilinmeyenli iki denklem haline getirilmesine çalışılmıştır. Bu amaçla, Harf Çiftlerinin Öğrenme listelerindeki Öğrenme sınıflarından uygun 2 tanesi alınmıştır.

$$OgP1 = (W_{tutan} x1^1) + (W_{tutmayan} x2^1) + (W_{geçme\ sayı\ ısı} x3^1) + (W_{klavye/yazım} x4^1) + \varepsilon$$

$$OgP2 = (W_{tutan} x1^2) + (W_{tutmayan} x2^2) + (W_{geçme\ sayı\ ısı} x3^2) + (W_{klavye/yazım} x4^2) + \varepsilon$$

(3.5)

Yukarıdaki denklemleri sadece $W_{geçme\ sayı\ ısı}$ ve $W_{klavye/yazım}$ kalacak şekilde sadeleştirmek için, Öğrenme sınıflarından Tutan Harf ve Tutmayan Harf sayısı aynı olan, iki sınıf seçilir. Böylece her iki denklemdeki W_{tutan} , $x1$, $W_{tutmayan}$ ve $x2$ değerleri aynı olur ve birbirini götürür. Elde edilen denklem aşağıdaki gibidir:

$$OgP1 = (W_{geçme\ sayı\ ısı} x3^1) + (W_{klavye/yazım} x4^1) + \varepsilon$$

$$OgP2 = (W_{geçme\ sayı\ ısı} x3^2) + (W_{klavye/yazım} x4^2) + \varepsilon$$

(3.6)

Benzer şekilde (3.6)'daki denklemleri sadece W_{tutan} ve $W_{tutmayan}$ kalacak şekilde sadeleştirmek için, Öğrenme sınıflarından aynı harf çiftine ait iki sınıf seçilir. Böylece her iki denklemdeki $W_{geçme\ sayısı}$, x_3 , $W_{klavye/yazım}$ ve x_4 değerleri aynı olur ve birbirini götürür. Elde edilen denklem aşağıdaki gibidir.

$$\begin{aligned} OgP1 &= (W_{tutan} x1^1) + (W_{tutmayan} x2^1) + \varepsilon \\ OgP2 &= (W_{tutan} x1^2) + (W_{tutmayan} x2^2) + \varepsilon \end{aligned} \quad (3.7)$$

Bu denklemler çözülerek sistemin yeni ağırlık değerleri bulunur ve güncellenir.

3.5.4.3 Öğrenme Puanının Hesaplanması

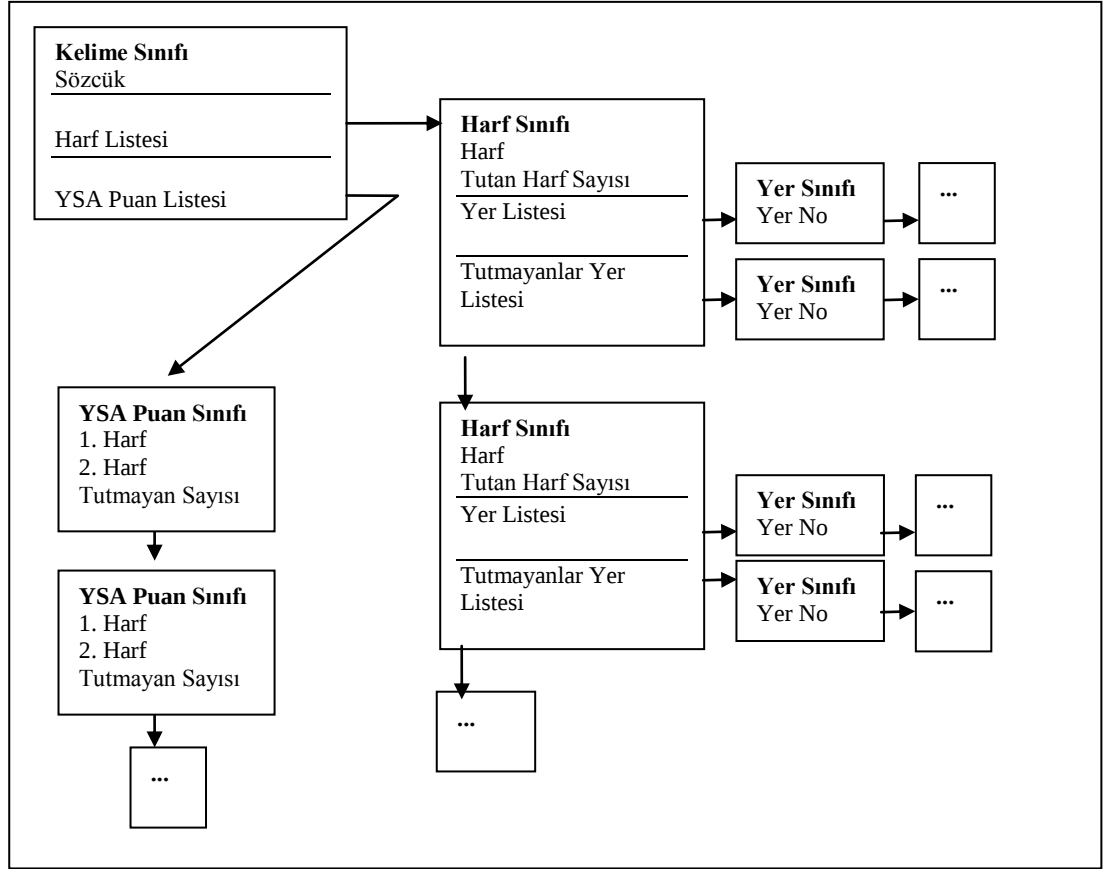
Öğrenme puanını hesaplamak için (3.4)'teki formül kullanılmaktadır. Formül tekrar aşağıda belirtilmiştir:

$$OgP = (W_{tutan} x1) + (W_{tutmayan} x2) + \sum (W_{geçme\ sayısı} x3) + \sum (W_{klavye/yazım} x4) + \varepsilon$$

Karşılaştırma işlemi sırasında karşılaştırılan iki sözcükten ve sistemin genel değişkenlerinden faydalanılarak, formüldeki değerler atanır ve Öğrenme Puanı elde edilir.

Öğrenme puanı elde edileceği zaman ilgili değerlerin doğrudan çekilebilmesi için bu değerleri Kelime sınıfı üzerinde tutan bir liste bulunmaktadır. Bu liste, YSA Puan sınıflarından oluşan YSA Puan listesidir. Bu listeyi oluşturmak için karşılaştırılan iki sözcük harf harf işlenir, her “tutmayan harf çiftine” rastlandığında ilk defa karşılaşıyorsa yeni bir YSA puan sınıfı oluşturulur. YSA Paun sınıfında, 1. Harf, 2. Harf ve Tutmayanlar Sayısı alanları bulunmaktadır. Varolan bir harf çiftine yeniden rastlandığında ise, ilgili YSA sınıfının Tutmayanlar Sayısı 1 arttırılır.

YSA Puan yapısı elde edildikten sonra, Öğrenme Puanı elde edilirken bu yapıdan çekilen değerler ile, Öğrenme Puan formülündeki (3.4) x_2 (tutmayan harf sayısı / sözcüklerden uzun olanın boyu) değişkenine ait değer doğrudan hesaplanabilir. Tutmayan harflerin dışında kalanlar da tutan harfler oranını yani x_1 değişkeninin değerini vermektedir. Bu doğrudan erişimi sağlayan yapıyı oluşturmanın amacı, öğrenme puanı hesaplama aşamasında, ilgili değerleri farklı yapılardan dolaşarak almak yerine, doğrudan erişilebilecek bir yapıdan almaktır. Şekil 3.16'da kelime sınıfına ait genel yapı gösterilmektedir.



Şekil 3.16: YSA Puan yapısının genel yapı içerisindeki yeri

Formülde, tutmayan harf çiftlerinin geçme sayılarından elde edilen kısım olan x_3 değişkeninin değerini bulmak için, Eğitim sınıfına bağlı Harf Çiftleri arasından bu harf çiftini içeren düğüm bulunur. Bu düğümden bu harf çiftinin karşılaşılma sayısı elde edilir. Eğitim sınıfı bünyesinde yer alan hatalı tüm harf çiftlerinin sayısını içeren Toplam Karşılaşılma Sayısı alanı da kullanılarak x_3 değeri elde edilir.

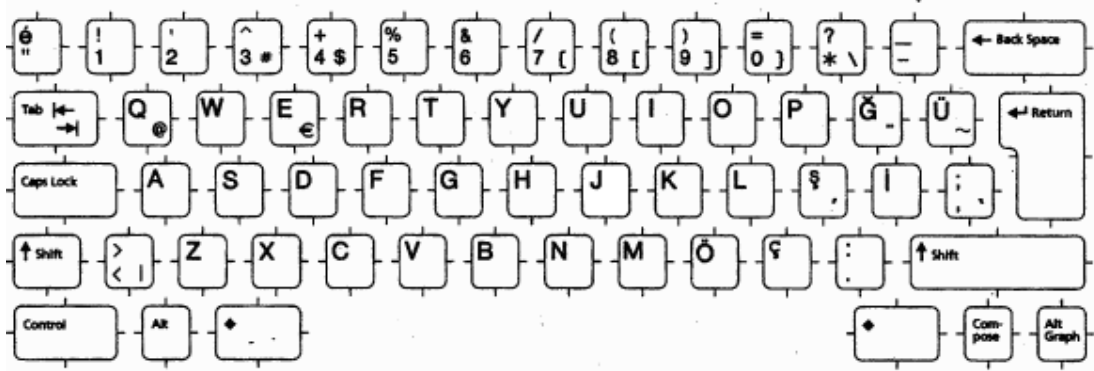
Formüldeki son kısım olan klavye/yazım ile ilgili değerlerin bulunması için yine eğitim sınıfına bağlı harf çiftleri arasından bu harf çiftini içeren düğüm bulunur. Bu düğüm içindeki klavye uzaklık değeri kullanılarak x_4 değişkeninin değeri elde edilir.

Klavyede Harfler Arası Uzaklıkların Bulunması

İki harfin klavyedeki uzaklıklarını bulabilmek amacıyla, klavyedeki her harf için klavyenin ortasındaki harfi merkez kabul edip, X ve Y koordinatları belirlenmiştir. İki harf arası uzaklık bu koordinat noktaları arası uzaklıktır. İki nokta arası uzaklık formülü ile uzaklık değeri hesaplanabilmektedir.

$$KU = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \quad (3.8)$$

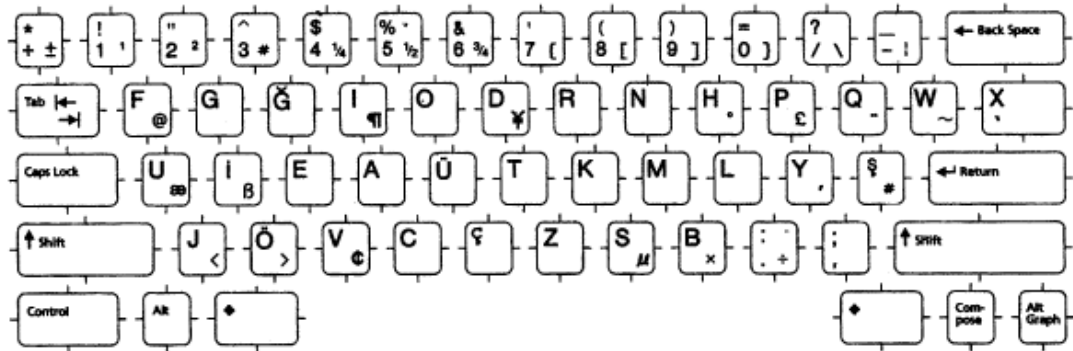
Şekil 3.17’de Q tipi klavye, Şekil 3.18’de F tipi klavye için harflerin dizilişlerinin görünümü verilmektedir. F klavye için “K” harfi, Q klavye için “J” harfi merkez kabul edilmiştir. Ek-A’da her iki klavye için de harflere ait belirlenmiş koordinat değerleri verilmektedir.



Şekil 3.17: Q Klavye Görünümü

Örneğin Şekil 3.17’de belirtilen Q klavye üzerinde Y ve M tuşları arasındaki uzaklık hesaplanırsa, Y harfinin koordinatları Y(-1, 1) ile M harfinin koordinatları M(1,-1) arası uzaklık, (3.8)’den bulunabilmektedir.

$$KU = \sqrt{((-1)-1)^2 + (1-(-1))^2} = 2\sqrt{2} = 2,828427$$



Şekil 3.18: F Klavye Görünümü

Elle yazılmış metinlerde yazımı birbirine benzeyen, harfler bazen yanlış, bazen okunaksız yazılmakta, bazen yıpranma sonucu değişmekte, bazen de tarayıcı tarafından yanlış algılanmakta ve elektronik ortama aktarıldığında hatalı olmaktadır.

Bu tür hatalar için de iyileştirmeler yapılmalıdır. Tablo 3.7’de yer alan harfler yazımı birbirine benzeyen harflerdir. Bu harfler nokta ya da çentik farkları sebebiyle birbirine benzemektedir.

Tablo 3.7: Türkçe’de yazımı birbirine benzeyen harfler

1. Harf	2. Harf
ı	1
ğ	g
ü	u
ş	s
ç	c
ö	o

Yazımı benzeyen harfler için klavyede yanyana yer aldıkları varsayımı yapıp, bu şekilde klavye ile ortak bir mantıkta ele alınmaları sağlanmaktadır.

Sonuç olarak, öğrenme sonucu sistem çok daha akıllı bir hale gelmektedir. Tablo 3.8’de öğrenme öncesi ve öğrenme sonrası puanlar belirtilmekte, öğrenme sonrası sistemini öğrendiği bilgiler doğrultusunda, daha mantıklı sonuçlar ürettiği görülmektedir.

Tablo 3.8: Öğrenmeden önce ve sonra elde edilmiş puanlar

1. Sözcük	2. Sözcük	Bulanık Mantık Puanı	Öğrenmeli Puan
merhaba	mehraba	0,952381	0,962767
gül	gul	0,666667	0,677052
iskandinavya	iksandinavya	0,972222	0,982608
iskandinavya	iksandinayva	0,944444	0,953565
sözcük	söczük	0,944445	0,954830

3.5.5 Bulanık Mantık ve YSA ile Genel Benzerlik Puanlamasının Yapılması

İki sözcük arası yapılan benzerlik puanlaması için önce bulanık mantık puanlaması kullanılmış, daha sonra bu puanlama YSA ile daha akıllı bir hale getirilmiştir. Ayrıca sözcüklerin kullanım oranları da ele alınmıştır. Böylece elde edilen Genel Benzerlik Puanı aşağıdaki şekilde formülleştirilmiştir.

$$GBP = BMP + OgP + KO \quad (3.9)$$

3.5.6 Kök ve Eklerin Karşılaştırılması, Bütünleşik Puanların Elde Edilmesi, Sonuç Önergelerin Çıkarımı

Denetleme işleminden önce yapılmış olan kaynak kümeleri oluşturma işlemi sayesinde sistem denetlenmeye hazır hale getirilmiştir. Kaynak kümeler, eldeki eğitim dosyasının biçimbirimsel çözümlemesi sonucu elde edilmiş kök ve eklerin, Kök Kaynak Kümesine ve Ek Kaynak Kümesine yazılması ile elde edilmiştir. Bu kaynak kümeler, hatalı sözcüklere önerme yapabilmek için danışman şeklinde kullanılmaktadır. Bir sözcüğün hatalı olup olmadığına, bu kaynak kümelerinde yer alıp almadığına göre karar verilir. Hatalı olduğu belirlenen bir sözcük için de yine bu kaynak kümelerine başvurarak doğru sözcük önermeleri yapılır. Kaynak kümelerine başvurma işlemi, doğrudan sözcüğün kaynak kümesinde yer alıp almalığına bakmak şeklinde değildir. Sözcük kök ve ekleri bakımından ayrı ayrı ele alınır. Türkçe’de belli sayıda kök sözcüğün bulunduğu, fakat onlarca ekin yanyana gelerek yeni sözcükler türetebileceği düşünülürse, kök ve ekler açısından ayrı ayrı denetleme yapıp, daha sonra elde edilen denetleme sonuçlarını birlikte yorumlanın daha mantıklı olduğu düşünülmektedir.

Bu amaçla, denetimi yapılacak sözcük için öncelikle biçimbirimsel çözümleme işlemi devreye girer. Biçimbirimsel çözümleme işleminden önce, Bölüm 3.5.2’de bahsedildiği gibi, bir normalizasyon işlemi yapılmaktadır. Biçimbirimsel çözümleme sonucu sözcük, kök ve eklerine ayrılır.

Elde edilen kök için, öncelikle kökün Kök Kaynak Kümesinde varlığı kontrol edilir. Eğer kök KKK’de var ise hatasız bir köktür, yok ise bu kök için doğru sözcük önermesi yapılmalıdır.

Doğru sözcük önermesi yapılması için, KKK’dan seçilen kökler listesi ile denetlenen kök arasındaki Genel Benzerlik Puanı (3.9, Bölüm 3.5.5) bulunur. En yüksek puanı elde eden kök, aday önermedir.

Genel Benzerlik Puanı hesaplamasının bir alt grubu olan Bulanık Mantık Puanlaması sırasında kullanılan eşik değeri, kök ve ekler için farklılık göstermektedir. Kökler daha uzun sözcüklerden oluştuğu için, eklere göre daha yüksek bir eşik değeri ile sınanırlar. Uygulama içerisinde bu değerler kök karşılaştırmaları için 0,749999 ek karşılaştırmaları için 0,499999’dur.

KKK'den tüm kökleri seçmek yerine hatalı sözcük yerine gelebilecek kökler seçilir. Bunun için, içinde hatalı kökün harflerinden en az biri olan kökler listelenir.

Gerçek önerme, en son önerilen aday kök ve aday ekler beraber yorumlanarak elde edilecektir.

Elde edilen köklerin denetlemesini yapmak için ekler ek ikilileri şeklinde ele alınır. Bu yaklaşımın temel sebebi, Türkçe'de eklerin sıralanışlarının sahip olduğu önemdir. Eklerin sıralanış biçimi, köklerin kazandığı anlam, dolayısıyla doğruluğu açısından yön göstericidir. Bu amaçla, ekler art arada gelmiş ek ikilileri şeklinde denetlenir. Çözümlemiş sözcüğe ait ekler sırası ile ikili gruplar halinde incelenir. Her ek ikilisinin EKK'de varlığı kontrol edilir. Eğer bu ikili, EKK'de yer alıyorsa hatasız bir ek ikilisidir. Ek ikilisi EKK'de yer almıyorsa doğru ek önermesi yapılmalıdır. Bu amaçla EKK'den uygun ek ikilisi bulma işlemi başlatılır. Bu işlem kapsamında ek ikilisi için önce 1. ek doğru kabul edilerek, daha sonra da 2. ek doğru kabul edilerek karşılaştırmalar yapılır. Sonuç olarak her iki durumdan elde edilmiş puanlar arasında en yüksek değere sahip ek ikilisi, hatalı ek ikilisi yerine aday olarak önerilir. Burada kesin bir önerme yapabilmek için birtakım kuralların sağlanıp sağlanmadığı kontrolü yapılmalıdır. Eklerin kökler ardına gelebilmesi için birtakım kuralların sağlanıyor olması gerekmektedir. Bu kuralların başında, Büyük Ünlü Uyumu* gelmektedir. Önerilen kök ve eklerin bir bütün halinde düşünüldüğünde Büyük Ünlü Uyumunu sağlıyor olması gerekmektedir. Bu kuralın dinamik olarak başarımı için, ek ikilileri bilgileri yanında, bu ek ikililerinden önce gelmiş kökün, son ünlü harfi veritabanında “ön sesliler“ isimli bir veri yapısında saklanmaktadır. Böylece kuralın dinamik tanımı, “bu ek ikilisinden önce gelebilecek köklerin sahip olması gerektiği son ünlü harf, ek ikilisine ait ön sesliler listesinde yer almalıdır” biçiminde oluşmuş olmaktadır. Ayrıca dinamik tanım sayesinde, bu kuralın istisnaları olan bazı ekler için de özel tanımlar yapılmasına gerek kalmamaktadır. Bu kurala uymayan önermeler geçerli sayılmamaktadır.

Büyük Ünlü Uyumu dışında bazı basit Türkçe ses kuralları da iyileştirme amaçlı geliştirilmiştir. Bu iyileştirmeler aşağıda alt başlıklar altında açıklanmaktadır:

İki sesli harfin yanyana gelmesi kontrolü: Kök ve ekler ayrı ayrı ele alınıp, elde

*Büyük Ünlü Uyumu: Türkçe bir sözcükte kalın ünlülerden sonra kalın, ince ünlülerden sonra inci ünlülerin gelmesi kuralıdır.[5]

edilen kök ve ek önermeleri birleştirilirken, birleştirme sırasında iki ünlü harfin yanyana gelip gelmediği kontrol edilmelidir. Çünkü Türkçe’de iki sesli harf yanyana gelemez. Bu kuralın istisnaları yabancı sözcükler ve bileşik sözcüklerdir. Fakat bu istisnalar kök sözcükleri etkileyen kullanımlardır. Köklerden sözcük türetilmesi sırasında bu kuralı bozan istisnalar yoktur. Bu kural, “kökün son harfi sesli ise, köke gelen ekin ilk harfi sesli olamaz” ve “bir ekin son harfi sesli ise, bundan sonra gelecek ekin ilk harfi sesli olamaz” şeklinde yorumlanırsa, istisnasız doğru bir kural elde edilmiş olur. Bu yüzden bir kök önerisi ardından yapılan önermelerde ve bir ek ikilisinde hatalı olan bir ek için önerme yapılması durumunda bu kontrol ele alınmaktadır.

Üç sessiz harfin yanyana gelmesi kontrolü: Türkçe’de en fazla iki sessiz harf yanyana gelebilir. Kök ve ek önermeleri yapıldıktan sonra elde edilen kök ve ekler birleşiminden oluşan sözcükte böyle iki sessiz harfi yanyana barındıran kök ya da ekler birleştirmişse, ikiden fazla sessiz harfin yanyana gelme durumu oluşabilir. Bu durumu engellemek için, sonuç önermede üç sessiz harfin yanyana gelip gelmediği kontrolü yapılır. Böyle bir önermeye rastlanırsa bu önerme iptal edilir.

Kaynaştırma Harfi Kontrolü: Morfoloji Analiz (Eryiğit) uygulaması bazı durumlarda kökü sözcükte yer alan kaynaştırma harfi ile beraber oluşturmaktadır. Bu tip durumlarda önerme olarak kökün kendi (kaynaştırma harfi olmadan) geldiği için kaynaştırma harfinin ayrıca ele alınması gerekmektedir. Burada yapılan iyileştirme hatalı sözcüğün son harfinin kaynaştırma harfi olduğu varsayıp (incelendiğinde bu şekilde denk gelme olasılığı çok yüksek çıkmıştır) kök ve eki birleştirirken araya bu kaynaştırma harfini eklemek şeklindedir. Bu iyileştirme kurallar için değil, kullanılan BÇ’ye özel yapılmıştır.

Eklerin denetlenmesi yaklaşımı bir örnek üzerinde incelenirse:

Denetlenecek sözcük: “gelememişlerdi”

Biçimbirimsel Çözümlemesi: gel+e+me+mış+ler+di

Kök: gel

Ek ikilileri: e-me, me-mış, mış-ler, ler-di, di-İ

Öncelikle kök sözcük KKK’de taranır. KKK’de böyle bir kök yer aldığı için “gel” hatasız bir köktür.

Ek ikilileri sırası ile EKK’de taranır.

- e-me : EKK’de yer alıyor. Hatasız ek ikilisi.
- me-mış: EKK’de yer almıyor. Hatalı ek ikilisi.

Bu aşamada, me-mış ek ikilisi için puanlama işlemi devreye girer. İlk olarak 1. eki doğru kabul edip, EKK’den “me” ile başlayan tüm ek ikilileri çekilir. Elde edilen liste Tablo 3.9’de gösterilmektedir. Bu listedeki tüm 2. ekler ile, denetlenen ek ikilisi “me-mış”ın 2. eki olan “mış” sözcüğü karşılaştırılır. Bu aşamada en yüksek puan 0,666667 ile mış – mış karşılaştırmasından elde edilmiştir.

Tablo 3.9 : Örnekteki karşılaştırmada ikinci ek için aday ek ikilileri

Ön Sesliler	Ek1	Ek2	Ek İkili Kullanım Sayıları	Ek İkili Kullanım Oranı
ieüöau	me	si	87	0,0021
ieöü	me	ler	82	0,002
ieüö	me	leri	9	0,0002
iöeü	me		166	0,004
ieüö	me	miş	14	0,0003
öe	me	ye	8	0,0002
e	me	dik	2	0
e	me	di	4	0,0001
ieı	me	n	23	0,0006
ieü	me	me	4	0,0001
ie	me	nin	12	0,0003
e	me	dir	2	0
e	me	yecek	2	0
eü	me	r	4	0,0001
i	me	de	1	0
ü	me	yle	2	0
e	me	yi	3	0,0001
e	me	yen	2	0
e	me	diğ	4	0,0001
i	me	mekte	12	0,0003
i	me	mek	12	0,0003

Daha sonra, 2. eki doğru kabul edip, EKK’den “mış” ile biten tüm ek ikilileri çekilir. Elde edilen liste Tablo 3.10’de gösterilmektedir. Bu listedeki tüm 1. ekler ile, denetlenen ek ikilisi “me-mış”ın 1. eki olan “me” sözcüğü karşılaştırılır. Bu aşamada en yüksek puan 0,5 ile me – ma karşılaştırmasından elde edilmiştir.

Tablo 3.10: Örnekteki karşılaştırmada birinci ek için aday ek ikilileri

Ön Sesliler	Ek1	Ek2	Ek İkili Kullanım Sayısı	Ek İkili Kullanım Oranı
ıa	l	miş	105	0,0025
au	ıl	miş	134	0,0032
ao	lan	miş	24	0,0006
aoı	an	miş	21	0,0005
ao	n	miş	51	0,0012
ıu	laş	miş	4	0,0001
ıau	ş	miş	5	0,0001
a	t	miş	2	0
au	ma	miş	25	0,0006
ao	a	miş	9	0,0002
a	ın	miş	14	0,0003
a	tan	miş	1	0
o	yan	miş	2	0
ı	la	miş	3	0,0001

Her iki aşamada elde edilmiş puanlar karşılaştırılıp yüksek olanı sağlayan ek ikilisi, önerilmeye aday kabul edilir. Örnekte 0,666667 ile 2. ek olan “miş” eki 0,5 puanlı 1. ek olan “ma” ekinden daha yüksek puanlıdır. Böylece hatalı “me-miş” ek ikilisine yapılacak aday önerme “me-miş” ikilisidir. Burada “aday” sözcüğünün kullanılmasındaki amaç, henüz Büyük Ünlü Uyumu ve diğer iyileştirme kuralını sağlayacak kontrollerin yapılmamış olmasıdır. Gerçek aday önerme, bu kurallar kontrolünden geçildiğinde yapılabilmektedir. Bu kontrollerden geçtikten sonra hatalı ek, aday ek ile güncellenir. Böylece denetimin kalan kısmı ile bütünlük bozulmamış olur. Sıradaki denetimler hatalı ekin düzeltilmiş hali üzerinden yapılamaya devam edilir.

Sıradaki diğer ek ikililerinin denetimine devam edilecek olunursa:

- mış-ler : mış daha önce ele alındığı için tekrar denetlemeye girmemektedir. Çünkü ilk ek için gerekli düzenleme yapılmış, ikinci ek de bir sonraki ek ikilisinde ele alınacaktır.
- ler-di : EKK’de yer alıyor. Hatasız ek ikilisi.
- di-ṭ : EKK’de yer alıyor. Hatasız ek ikilisi.

Tüm denetlemeler sonucunda elde edilen önermenin çözümlenmiş hali, “gel+e+me+miş+ler+di” şeklindedir. Aşağıda anlatılan çözümlenmiş sözcüğün

bütünleşik haline getirilmesi işleminin ardından, sonuç önermesi “gelememişlerdi” şeklini almıştır.

BÇler sözcükleri kök ve eklerine ayırırken bazı ses düşmeleri, ulamaları gibi kökte ve eklerde oluşan ses değişimlerini kaldırıp, kökün ve eklerin saf hallerini sonuç olarak sunmaktadırlar. Denetleme ve önerme işleminde de kökün ve eklerin saf hali kullanılmaktadır. Oysa Türkçe’de köklerden sonra gelen bazı ekler kökte ya da ekte ses değişimlerine sebep olmaktadır. Çözümlemiş haldeki önermeyi bütün bir sözcük haline getirirken ses değişimi ile ilgili kurallar uygulanmıştır. Bu kurallar aşağıda alt başlıklar altında açıklanmaktadır:

Ad durum ekleri ve ek eylemde ünsüz benzeşmesi: Ad durum ekleri -da, -de, -dan, -den ve ek eylem -dır, -dir, -dur, -dür, sözcük sonlarında ünsüz benzeşmesine uğrar. Sert ünsüzle (ç, f, h, k, p, s, ş, t) biten sözcüklerden sonra, bu eklerin başındaki d, t’ye dönüşür [5]. Çözümlemiş kök ve eklerin birleştirme sırasında bu kural uygulanmaktadır. Örneğin; “ağaç+da” sözcüğü birleştirilince “ağaçta” şeklini almaktadır.

Sert ünsüzlerin yumuşaması: Türkçe sözcüklerin sonunda bulunan p, ç, t, k ünsüzleri, sözcük ünlü ile başlayan bir ek aldığı zaman, iki ünlü arasında yumuşayarak b, c, d, ğ’ye dönüşür [5]. Çözümlemiş kök ve eklerin birleştirme sırasında bu kural uygulanmaktadır. Örneğin; “kitap+ı” sözcüğü birleştirilince “kitabı” şeklini almaktadır. Yalnız tek heceli sözcükler bu kural dışındadır. Örneğin; “top-u” sözcüğü değişmeden “topu” şeklini alır. Bu yüzden uygulama içerisinde bu kural, tek heceli kökler için uygulanmamaktadır.

Diğer bir sert ünsüzlerin yumuşaması kuralı da, sözcük sonunda n ünsüzünden sonra gelen ç, k sesleri bir ünlü ile karşılaşınca yumuşayarak c, g seslerine dönüşür [5]. Örneğin “genç+ın” sözcüğü “gencin” şeklini alır. Bu kural da uygulama kapsamında gerçekleştirilmektedir.

Son aşamada hatalı bir sözcük için elde edilen önermeler, puanlarına göre yüksekten alçağa doğru sıralanır ve kullanıcıya önerilir. Farklı çözümlemiş şekillerine sahip fakat bütünlemiş hali aynı olan sözcükler olabilmektedir.

3.5.7 Deney Sonuçları

Uygulama, belli bir alandaki büyük bir metnin, eğitim kümesi olarak kullanıldıktan sonra, yine aynı konudaki farklı bir metnin denetlenmesi şeklinde test edilmiştir. Farklı biçimbirimsel çözümleyiciler kullanılarak aynı metinden farklı Kaynak Kümeleri oluşturulmuştur. Yine eldeki test metni de farklı biçimbirimsel çözümleyiciler ile çözümlendikten sonra ayrı ayrı test edilmiştir. Kaynak ve test kümelerinin farklı kombinasyonları denenmiştir. Örneğin X BÇ'si ile oluşturulmuş bir kaynak küme kaynak alınarak, Y BÇ'si ile çözümlenmiş bir metin, Z BÇ'si ile oluşturulmuş bir kaynak küme kaynak alınarak, Z BÇ'si ile çözümlenmiş bir metin vb. test edilmiştir. Böylece biçimbirimsel çözümleyicinin sistemin verimliliği açısından önemi tartışılmıştır.

Kullanılan test konularından biri edebiyat konusunda yapılmış tez çalışmalarından derlenmiş bir kümedir. Bilkent Üniversitesi, Türk Edebiyatı Bölümünde Yüksek Lisans Tezi olarak hazırlanmış 47 adet tezdin 46 tanesi eğitim kümesi olarak kullanılmak üzere derlenmiş, 1 adet tezin bir kısmı da test edilmek üzere hazırlanmıştır. Test edilecek tez yeniden hızlı bir şekilde yazılmış, yazım sırasında yapılan hatalar düzeltilmemiştir.

Eğitim kümesi 749,207 adet sözcükten oluşmaktadır. Tezlerden oluşturulmuş küme hem Morfolojik Analiz (Eryiğit) uygulaması ile hem de Zemberek uygulaması ile çözümlenerek iki farklı eğitim kümesi elde edilmiştir.

Morfolojik Analiz (Eryiğit) uygulaması çözümlenmiş metin dosyası çözümlenmiş 2,697,614 adet sözcük içermektedir. Bir sözcüğe ait birden fazla çözümleme olabileceği için, toplam sözcük sayısından oldukça fazla sayıda çözümlenmiş sözcük bulunmaktadır. Çözümlenmiş dosya eğitim ve kaynak kümeleri oluşturmak amaçlı çalıştırıldıktan sonra, kaynak kümelerinin içerdiği eleman sayıları Tablo 3.11'de belirtilmektedir.

Tablo 3.11: Morfolojik Analiz (Eryiğit) BÇ'si kullanılarak oluşan kaynak kümeler

Kök Kaynak Kümesi (Tezler)		Ek Kaynak Kümesi (Tezler)	
Toplam Sözcük Sayısı	Farklı Sözcük Sayısı	Toplam Ek İkili Sayısı	Farklı Ek İkili Sayısı
2,697,614	92,071	4,319,816	4,854

Kök ve Ek Kaynak Kümeleri oluşturulduktan sonra her küme elemanı için kullanım oranları hesaplanılmaktadır. Kök Kaynak Kümesinden elde edilmiş kullanım oranlarının bir kısmı büyükten küçüğe doğru sıralanmış şekilde Tablo 3.12’de verilmektedir.

Tablo 3.12: Morfolojik Analiz (Eryiğit) BÇ’si kullanılarak oluşan KKK

Kök	Kullanım Sayısı	Kullanım Oranı
ol	30712	0,0114
bi	26076	0,0097
iç	21135	0,0078
ve	20328	0,0075
bu	19743	0,0073
an	14294	0,0053
kat	13311	0,0049
içi	12003	0,0044
ge	10514	0,0039
on	10366	0,0038
de	9966	0,0037
gör	9866	0,0037
et	9454	0,0035
olduk	9048	0,0034

Ek Kaynak Kümesinden elde edilmiş kullanım oranlarının bir kısmı büyükten küçüğe sıralanmış şekilde Tablo 3.13’de verilmektedir.

Tablo 3.13: Morfolojik Analiz (Eryiğit) BÇ’si kullanılarak oluşan EKK

Ön Sesliler	Ek1	Ek2	Ek İkili Kullanım Sayısı	Ek İkili Kullanım Oranı
ieraiüuoö	n		262942	0,0609
eoüüauöı	i		187924	0,0435
auoeüiö	ı		136923	0,0317
ieüöaouı	e		137046	0,0317
eiüauoöı	in		131080	0,0303
auoıeüö	a		120380	0,0279
arouüei	in		113999	0,0264
iaüüeuö	r		95987	0,0222
ıauoüei	da		49690	0,0115
uoelaiüiö	u		47428	0,011
iüööaouı	de		43586	0,0101
eüöıaiou	dir		40392	0,0094
ıauoeüi	dır		36489	0,0084
eoaruiü	an		32955	0,0076
auoeiü	lar	ı	31138	0,0072

Diğer bir biçimbirimsel çözümleyici olan Zemberek uygulamasını kullanılarak oluşturulmuş olan Kök ve Ek Kaynak Kümelerinin içerdiği eleman sayıları Tablo 3.14’de belirtilmiştir. Zemberek uygulaması ile çözümlenmiş eğitim metni 1,148,526 adet sözcük çözümlemesi içermektedir (Toplam sözcük sayısı daha önce de belirtildiği gibi 749,207 adettir).

Tablo 3.14: Zemberek BÇ’si kullanılarak oluşan kaynak kümeler

Kök Kaynak Kümesi (Tezler)		Ek Kaynak Kümesi (Tezler)	
Toplam Sözcük Sayısı	Farklı Sözcük Sayısı	Toplam Ek İkili Sayısı	Farklı Ek İkili Sayısı
1,148,526	23,237	1,237,087	2,588

Kök ve Ek Kaynak Kümelerinden elde edilmiş kullanım oranlarının bir kısmı büyükten küçüğe sıralanmış şekilde Tablo 3.15’de verilmektedir.

Tablo 3.15: Zemberek BÇ’si kullanılarak oluşan KKK ve EKK

Kök	Kullanım Sayısı	Kullanım Oranı	Ön Sesliler	Ek1	Ek2	Ek İkili Kullanım Sayısı	Ek İkili Kullanım Oranı
bir	23503	0,0205	eiüöaouı	i		45799	0,037
ol	22036	0,0192	auo	ı		45081	0,0364
ve	19539	0,017	ieöaüuoı	e		32625	0,0264
bu	16475	0,0143	üieööauı	in		28030	0,0227
de	15246	0,0133	üieöauo	de		27063	0,0219
iç	11325	0,0099	auo	da		26152	0,0211
in	9292	0,0081	eaüuoioö	n		24867	0,0201
anla	8395	0,0073	oauı	a		24356	0,0197
kadın	8164	0,0071	aiou	in		22861	0,0185
o	7919	0,0069	oau	an		15114	0,0122
da	7241	0,0063	üieöoauı	ler	in	13926	0,0113
et	6633	0,0058	ıauo	lar	in	13489	0,0109
on	6287	0,0055	ıauo	dan		12579	0,0102
kadı	5848	0,0051	uo	u		12613	0,0102
için	5771	0,005	eauüi	r		11127	0,009
gör	5139	0,0045	aiou	in	da	11148	0,009
al	5043	0,0044	aoıöieiü	lar		10936	0,0088
yer	4878	0,0042	ieüöoau	ler		10664	0,0086
üzeri	4789	0,0042					

Farklı B.Ç.ler kullanılarak kaynak kümeler elde edildikten sonra test metni farklı kaynak küme ve B.Ç. çözümlemesi kombinasyonları ile test edilmiştir.

Hem Kaynak Kümesini oluşturan çözümleyici olarak hem de test metnini çözümlemek için Morfolojik Analiz (Eryiğit) uygulaması kullanıldığı takdirde hatalı metinde hata bulma oranı %85,29 olmaktadır. Bulunan hataların düzeltilme oranı ise %82,75 olmaktadır. Hata bulma ve düzeltme anlamında elde edilen genel oran %70,58'dir. Hatalı biçimbirimsel çözümlmeler dolayısıyla oluşmuş yanlış alarm (doğru iken hatalı hale getirilen sözcük) oranı ise %0,29 olmaktadır. Alternatifli çözümlmeler yanılmaya sebep olmaktadır. Örneğin “sahip” yerine hatalı yazılan “sakup” sözcüğü için yapılan çözümleme “sak+ip” şeklindedir. Kaynak kümede “sak” kökü ve “ip-B” ek çifti bulunduğu için, sistem bu hatayı bulamamaktadır. “sak” kökü eğitim sırasında örneğin “sak+lı+dır” sözcüğü ile kaynak kümeye dahil olmaktadır. Bu sebeple hata bulma oranında düşme olabilmektedir.

Hem Kaynak Kümesini oluşturan çözümleyici olarak hem de test metnini çözümlemek için Zemberek uygulaması kullanıldığı takdirde hatalı metinde hata bulma oranı %61,76 olmaktadır. Bulunan hataların düzeltilme oranı ise %95,23 olmaktadır. Hata bulma ve düzeltme anlamında elde edilen genel oran %58,82'dir. Hatalı biçimbirimsel çözümlmeler dolayısıyla oluşmuş yanlış alarm oranı ise %0'dır. Zemberek yazılımının hatalı sözcükler için her zaman bir çözümleme üretmemesi, hata bulma oranını olumsuz yönde etkilemektedir. Hata bulunduğu takdirde düzeltilme oranı ise yüksektir.

Kaynak kümesinin BÇ'si olarak Zemberek uygulaması, denetim metninin BÇ'si olarak da Morfolojik Analiz (Eryiğit) uygulaması seçildiği takdirde, farklı yapıda çözümleme sebebiyle yanlış oranı yükselmekte, çok verimli sonuçlar alınamaktadır.

Farklı bir deney kümesi olarak da Prof. Dr. Eşref Adalı'nın Mikroişlemciler, Mikrobilgisayarlar kitabı eğitim kümesi, Mikroişlemci Sistemleri dersinin geçmiş yılları sınav soruları ve bazı örnek soruları da denetlenecek küme olarak seçilmiştir.

Eğitim kümesi 76,897 adet sözcükten oluşmaktadır. Tezlerden oluşturulmuş küme üç BÇ uygulaması ile de çözümlenerek üç farklı eğitim kümesi elde edilmiştir.

Morfolojik Analiz (Eryiğit) uygulaması çözümlenmiş metin dosyası çözümlenmiş 257,462 adet sözcük içermektedir. Bir sözcüğe ait birden fazla çözümleme olabileceği için, toplam sözcük sayısından oldukça fazla sayıda çözümlenmiş sözcük bulunmaktadır. Elde edilen kök ve ek sayıları Tablo 3.16'de belirtilmektedir.

Tablo 3.16: Morfolojik Analiz (Eryiğit) BÇ'si kullanılarak oluşan kaynak kümeler

Kök Kaynak Kümesi (Mikroişlemciler)		Ek Kaynak Kümesi (Mikroişlemciler)	
Toplam Sözcük Sayısı	Farklı Sözcük Sayısı	Toplam Ek İkili Sayısı	Farklı Ek İkili Sayısı
257,462	11,724	448,776	2,193

Zemberek uygulamasını kullanılarak oluşturulmuş olan Kök ve Ek Kaynak Kümelerinin içerdiği eleman sayıları Tablo 3.17'de belirtilmiştir. Zemberek uygulaması ile çözümlenmiş eğitim metni 118,816 adet sözcük çözümlemesi içermektedir.

Tablo 3.17: Zemberek BÇ'si kullanılarak oluşan kaynak kümeler

Kök Kaynak Kümesi (Mikroişlemciler)		Ek Kaynak Kümesi (Mikroişlemciler)	
Toplam Sözcük Sayısı	Farklı Sözcük Sayısı	Toplam Ek İkili Sayısı	Farklı Ek İkili Sayısı
118,816	3,096	133,892	1,219

Son olarak Biçimbirimsel Çözümleyici (Oflazer) uygulaması ile oluşturulmuş Kök ve Ek Kaynak Kümelerinin içerdiği eleman sayıları Tablo 3.18'de belirtilmiştir. Elde edilen eğitim metni 116,447 adet sözcük çözümlemesi içermektedir.

Tablo 3.18: BÇ (Oflazer) uygulaması kullanılarak oluşan kaynak kümeler

Kök Kaynak Kümesi (Mikroişlemciler)		Ek Kaynak Kümesi (Mikroişlemciler)	
Toplam Sözcük Sayısı	Farklı Sözcük Sayısı	Toplam Ek İkili Sayısı	Farklı Ek İkili Sayısı
116,447	3,411	132,886	1,429

Denetlenecek sınav soruları, Morfolojik Analiz (Eryiğit) uygulaması ile çözümlenip, yine aynı uygulama çözümlemesi ile oluşmuş kaynak kümeleri ile denetlenirse hata bulmada %94,73'lük bir başarımla sağlanmaktadır. Bu hatalar %78,94 oranında düzeltilmiştir. Genel olarak hata bulma ve düzeltme oranı %83,33. Yanlış alarm oranı ise %1,13 değerindedir.

Hem kaynak küme hem de denetlenecek dosya çözümlemesinde Zemberek uygulaması kullanıldığında hata bulmada %45'lik bir oran elde edilmiştir. Bu hataların %100'ü düzeltilmiştir. Yanlış alarm oranı 0'a yakın bir değerdedir. Genel hata bulma ve düzeltme oranı %45'de kalmıştır. Bunun sebebi kaynak kümesinin daha küçük olmasıdır. Çünkü Zemberek uygulaması çok alternatifli sonuç

üretmemekte, kaynak küme boyu sınırlı kalmaktadır. Oysa Morfolojik Analiz (Eryiğit) çok alternatifli sonuçlar ürettiği için, kaynak kümesi çok daha büyüktür. Bu yüzden başarımı daha yüksektir.

Son olarak Biçimbirimsel Çözümleyici (Oflazer) uygulaması kullanılarak çözümlenip oluşturulmuş kaynak kümeleri kullanılıp, denetleme için çözümleme işi Morfolojik Analiz (Eryiğit) ile yapılırsa başarımlar düşük olmaktadır. Bunun sebebi de farklı BÇlerin kullanılmış olmasıdır. (Biçimbirimsel Çözümleyici (Oflazer) uygulamasının kendisi kullanılmadığı için denetim için yararlanılamış, sadece kaynak küme oluşturmunda, Oflazer tarafından çözümlenip gönderilmiş eğitim dosyaları, eğitim amaçlı kullanılmıştır.)

Elde edilen sonuçlardan bir kısmı karşılaştırma amaçlı Tablo 3.19’de özetlenmiştir. Yüksek başarımın hem eğitim hem denetim sırasında aynı BÇ’nin kullanıldığı durumlarda elde edildiği gözlemlenmiştir. Tablo 3.19’ de özeti yer alan deneylerde, kolonlarda belirtilen BÇler bir deney için hem eğitim, hem de denetimde kullanılmıştır.

Sonuçlardan, BÇ’nin başarımının sistem başarımını doğrudan etkilediği görülmektedir. Hem doğru hem hatalı sözcükler için çok alternatifli sonuçlar üreten Morfolojik Analiz (Eryiğit) uygulaması ile hata bulma ve düzeltmede daha iyi bir başarımlar sağlanmıştır. Fakat çok alternatiflilik yanlış alarm düzeyini yükseltmektedir.

Tablo 3.19: Deney Sonuçları Özeti

	Tezler deney kümesi		Mikroişlemciler deney kümesi	
	Morf. Analiz (Eryiğit)	Zemberek	Morf. Analiz (Eryiğit)	Zemberek
Hata Bulma Oranı	%85,29	%61,76	%94,73	%45
Bulunan Hataların Düzeltme Oranı	%82,75	%95,23	%78,94	%100
Genel Hata Bulma ve Düzeltme Oranı	%70,58	%58,82	%83,33	%45
Yanlış alarm oranı	%0,29	%0	%1,13	%0

4. BÖLÜM

YAZILIMIN AÇIKLANMASI

4.1 Veri Yapıları

Kök ve ek için oluşturulan kaynak kümeleri bir veritabanı üzerinde saklanmaktadır. Kökler için bir tablo, ekler için ayrı bir tablo kullanılmaktadır.

Kökler tablosu sıra numarası alanı (sayısal), kök alanı (metin), kök kullanım sayısı (sayısal) ve kök kullanım oranı (sayısal) olmak üzere dört alandan oluşmaktadır. Kök alanı elde edilmiş kök sözcüğü, kullanım sayısı bu sözcüğün eğitim kümesi içinde geçme sayısını ve kullanım oranı kullanım sayısının eğitim kümesinde yer alan toplam sözcük sayısında oranını tutmaktadır. Şekil 4.1’de tablo yapısı görülmektedir.

	Alan İsmi	Veri Türü
no		AutoNumber
kok		Text
kullanim_sayisi		Number
kullanim_orani		Number

Şekil 4.1: Kök tablosu veri yapısı

Ekler tablosu sıra numarası alanı (sayısal), ön sesliler alanı (metin), ek1 alanı (metin), ek2 alanı (metin), ek ikilisi kullanım sayısı (sayısal) ve ek ikilisi kullanım oranı (sayısal) olmak üzere altı alandan oluşmaktadır. Ek1 ve Ek2 alanları eğitim kümesinden elde edilen ek ikililerini, ön sesliler alanı, eğitim kümesinde ek ikilisinden önce yer almış sesli harfleri, ek ikilisi kullanım sayısı da ek ikilisinin eğitim kümesi içinde geçme sayısını tutmaktadır. Ek ikilisi kullanım oranı ek ikilisi kullanım sayısının eğitim kümesindeki tüm ek ikilileri sayısına oranıdır. Şekil 4.2’de tablo yapısı görülmektedir.

	Alan İsmi	Veri Türü
no		AutoNumber
on_sesliler		Text
ek1		Text
ek2		Text
ek_ikilisi_kullanım_sayisi		Number
ek_ikilisi_kullanım_orani		Number

Şekil 4.2: Ek tablosu veri yapısı

Uygulama içerisinde oluşturulmuş veri yapıları aşağıda anlatılmıştır.

Yer Numarası Sınıfı: Harflerin yer numaralarını tutan yapıdır.

Sınıf İsmi: snfYer	
Alan Tanımı	Açıklama
public int m_YerNo	Harfin sözcük içerisindeki yer numarası
public snfYer m_Sonraki	Bir sonraki Yer sınıfı

Harf Sınıfı: Harfleri ve içerdiği bilgileri içeren yapıdır.

Sınıf İsmi: snfHarf	
Alan Tanımı	Açıklama
public char m_HarfAdi	Harfin adı
public snfYer m_Yer	Harfin sözcükte yer aldığı yer numaralarını tutan Yer sınıfı listesi
public snfYer m_TutmayanYer	Harfin karşılaştırılan sözcükte yer aldığı fakat ait olduğu sözcükteki yer numarası ile aynı olmayan yer numaralarını tutan Yer sınıfı listesi
public int m_TutanSayisi	Harfin karşılaştırılan sözcükte yer aldığı ve ait olduğu sözcükteki yer numarası ile aynı olan yerlerin sayısı
public snfYeriTutmayanPuani m_TPL	Yeri Tutmayan Puan Listesi
public snfHarf m_Sonraki	Bir sonraki Harf sınıfı

Harf Çifti Sınıfı: İki sözcükte aynı yerde yer alan ama aynı olmayan harfleri tutan yapıdır.

Sınıf İsmi: snfHarfCifti	
Alan Tanımı	Açıklama
public char m_Kaynak	Hatalı Harf
public char m_Hedef	Asıl Harf (olması gereken)
public int m_Karsilasilma	Bu iki harfin karıştırılması sayısı
public int m_KlavyeUzakligi	Bu iki harfin klavyedeki uzaklıkları
public snfOgrenme m_Ogrenme	Öğrenme Sınıfı Listesi
public snfHarfCifti m_Sonraki	Bir sonraki Harf Çifti sınıfı

Kelime Sınıfı: Sözcük yapısını ve içerdiği bilgiyi tutan yapıdır.

Sınıf İsmi: snfKelime	
Alan Tanımı	Açıklama
public String m_KelimeIcerik	Kelimenin İçeriği
public snfHarf m_HarfListesi	Kelimenin İçerdiği Harfler Listesi
public snfYSAPuani m_YSAPL	Yapay Sinir Ağları Puan Listesi
public snfHarfCifti m_Sonraki	Bir sonraki Harf Çifti sınıfı

Klavye Sınıfı: F ve Q tipi klavyede tuşların koordinatlarını tutan yapıdır.

Sınıf İsmi: snfKlavye	
Alan Tanımı	Açıklama
ArrayList X_Koord_Q	Harflerin Q Klavyedeki X koordinatları
ArrayList Y_Koord_Q	Harflerin Q Klavyedeki Y koordinatları
ArrayList X_Koord_F	Harflerin F Klavyedeki X koordinatları
ArrayList Y_Koord_F	Harflerin F Klavyedeki Y koordinatları

Yeri Tutmayan Harf Çifti Sınıfı: İki kelimedede de yer alan fakat yerleri aynı olmayan harf çiftlerini tutan yapıdır.

Sınıf İsmi: snfYeriTutmayanHarfCifti	
Alan Tanımı	Açıklama
public int m_Kaynak	Hatalı Harf
public int m_Hedef	Asıl Harf (olması gereken)
public int m_Kullanildi	Yeri Tutmayan Harf Çiftinin puanlamada kullanılıp kullanılmadığı bilgisi
public int m_Uzaklik	İki harf arası yer numarası farkı
public snfYeriTutmayanHarfCifti m_Sonraki	Bir sonraki snfYeriTutmayanHarfCifti sınıfı

Öğrenme Sınıfı: Harf Çiftlerine bağlı, bu harf çiftlerinin yer aldığı puanlamalardan elde edilen değerleri tutan yapıdır.

Sınıf İsmi: snfOgrenme	
Alan Tanımı	Açıklama
public int m_Uzunluk	Harf Çiftinin yer aldığı karşılaştırmadaki kelimelerden uzun olanının boyu
public int m_TutanSayisi	Harf Çiftinin yer aldığı karşılaştırmadaki tutan harf sayısı
public int m_TutmayanSayisi	Harf Çiftinin yer aldığı karşılaştırmadaki tutmayan harf sayısı
public float m_Puan	Harf Çiftinin yer aldığı karşılaştırmadaki puan
public snfOgrenme m_Sonraki	Bir sonraki snfOgrenme sınıfı

YSA Sınıfı: Yapay Sinir Ağları kullanılarak öğrenme puanı elde etme aşamasında kullanılan, birbiri yerine hatalı olarak yazılmış harf çiftlerini ve karşılaştırma sayılarını tutan yapıdır.

Sınıf İsmi: snfYSApuani	
Alan Tanımı	Açıklama
public char m_Harf1	Birbiri yerine hatalı olarak yazılmış harflerden ilki
public char m_Harf2	Birbiri yerine hatalı olarak yazılmış harflerden ikincisi
public int m_TutmayanlarSys	Hatalı harf çiftine karşılaşılma sayısı
public snfYSApuani m_Sonraki	Bir sonraki snfYSApuani sınıfı

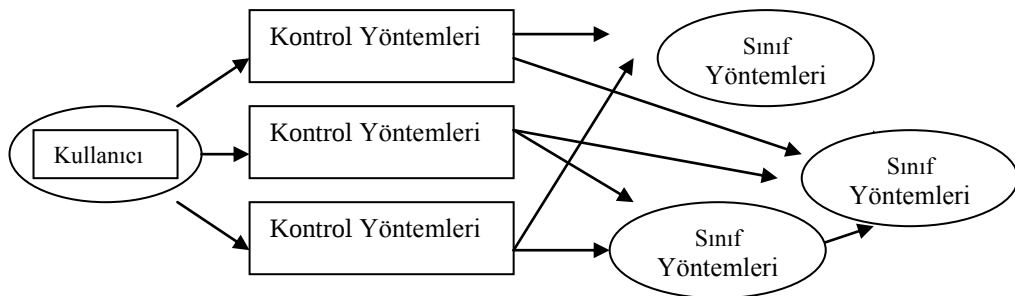
Eğitim Sınıfı: Eğitim ile elde edilmiş bilgilerin tutulduğu yapıdır.

Sınıf İsmi: snfEgitim	
Alan Tanımı	Açıklama
public snfHarfCifti m_HarfCiftiListesi	Eğitim ile elde edilmiş hatalı harf çiftleri listesi
public float m_TutanAgirligi	Eğitim ile tutan harflerden elde edilmiş olan sistemin tutan harf ağırlığı
public float m_TutmayanAgirligi	Eğitim ile farklı harflerden elde edilmiş olan sistemin tutmayan harf ağırlığı
public float m_KlavveAgirligi	Eğitim ile klavye uzaklıklarından elde edilmiş olan sistemin klavye ağırlığı
public float m_KarsilasilmaAgirligi	Eğitim ile hatalı harf çiftlerine karşılaşılma sayılarından elde edilmiş olan sistemin karşılaşılma ağırlığı
public int m_ToplamKarsilasilma	Eğitim boyunca hatalı harf çiftine toplam karşılaşılma sayısı

Dizgi İşlemleri Sınıfı: Uygulama sırasında yapılması gereken dizgi işlemlerini içeren sınıftır.

4.2 Yöntemler

Uygulamada yöntemler, arayüz kontrollerinin (tuşlar, metin kutuları, radyo düğmeleri, onay kutuları, sekme sayfaları) tetiklediği yöntemler, genel amaçlı yöntemler ve sınıflara ait yöntemler şeklinde üç grupta incelenebilir. Şekil 4.3’de yöntemler ve birbirleri ile olan bağlantıları gösterilmektedir.



Şekil 4.3: Uygulamada kullanılan yöntem çeşitleri

Arayüz kontrollerinin tetiklediği yöntemler, arayüzden bir kontrol aracılığıyla yapılan bir işlemin (tuşa basma, sekme sayfaları arasında gezme vb.) sonucunda çalışmaktadır. Bu yöntemler, arayüzdeki diğer kontrol nesnelerinin (metin kutuları, onay kutuları vb.) değerlerini ve sınıf yöntemlerini kullanarak işlem yapmaktadır. Arayüz yöntemleri aşağıda listelenmektedir.

Yöntem	private void btnEgitim_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnEgitim (Eğitim sekmesindeki “Eğitime Başla” tuşu)
Çağırdığı Yöntemler	DosyaIsmiAta, KaynakKumeTemizle, KaynakKumeBilgileri
Açıklama	Eğitim İşlemini gerçekleştirmektedir. Kullanıcı tarafından seçilen eğitim dosyası okunup, her satırı işlenip, veritabanına yazılmaktadır.

Yöntem	private void btnGetir_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnGetir (Eğitim sekmesindeki “Kaynak Kümeleri Getir” tuşu)
Çağırdığı Yöntemler	KaynakKumeBilgileri
Açıklama	Veritabanından Kaynak Kümelere ait bilgileri getirmektedir.

Yöntem	private void btnSil_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnSil (Eğitim sekmesindeki “Kaynak Kümeleri Temizle” tuşu)
Çağırdığı Yöntemler	KaynakKumeTemizle, KaynakKumeBilgileri
Açıklama	Veritabanından Kaynak Kümelere ait bilgileri silmektedir.

Yöntem	private void rbtZemberekEgitim_CheckedChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	rbtZemberekEgitim (Eğitim sekmesindeki, Biçimbirimsel Çözümleyici grubundaki “Zemberek” radyo düğmesi)
Çağırdığı Yöntemler	
Açıklama	Zemberek Biçimbirimsel Çözümleyicisi seçilince, sadece çözümlenmiş dosyalar üzerinde işlem yapılmaktadır. Bu yüzden Dosya Türü seçim grubunu pasifleştirmektedir.

Yöntem	private void rbtMorfolojikEgitim_CheckedChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	rbtZemberekEgitim (Eğitim sekmesindeki, Biçimbirimsel Çözümleyici grubundaki “Morfolojik Analiz (Eryiğit)” radyo düğmesi)
Çağırdığı Yöntemler	
Açıklama	Morfolojik Analiz Biçimbirimsel Çözümleyicisi seçilince, hem çözümlenmiş hem de asıl dosyalar üzerinden işlem yapılabilmektedir. Bu yüzden Dosya Türü seçim grubunu aktifleştirmektedir.

Yöntem	private void rbtBicimbirimseleEgitim_CheckedChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	rbtBicimbirimseleEgitim (Eğitim sekmesindeki, Biçimbirimsele Çözümleyici grubundaki “Biçimbirimsele Çözümleyici (Of lazer)” radyo düğmesi)
Çağırdığı Yöntemler	
Açıklama	Biçimbirimsele Çözümleyici (Of lazer) seçilince, sadece çözümlenmiş dosyalar üzerinde işlem yapılmaktadır. Bu yüzden Dosya Türü seçim grubunu pasifleştirmektedir.

Yöntem	private void btnDenetim_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnDenetim (Denetim sekmesindeki “Yazımı Denetle” tuşu)
Çağırdığı Yöntemler	DosyaIsmiAta, KatariDuzenle, Denetle, KelimeyiOlustur, DosyayiSatirlaraDonustur, MorfolojikAnaliz
Açıklama	Denetim İşlemini gerçekleştirmektedir. Kullanıcı tarafından seçilen denetim dosyası okunup, her satırı işlenip, veritabanına yazılmaktadır. BÇ çeşidi olarak Morfolojik Analiz (Eryiğit) programı ve dosya türü olarak “asıl” seçilmişse, dosya için önce satırlara ayırma ve Morfolojik Analiz (Eryiğit) programı ile biçimbirimsele analiz işlemi yapılmaktadır.

Yöntem	private void btnSonuclariTemizle_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnSonuclariTemizle (Denetim sekmesindeki “Temizle” tuşu)
Çağırdığı Yöntemler	
Açıklama	Denetim sonucu elde edilmiş bilgiler ekrandan temizlenmektedir.

Yöntem	private void chkDetay_CheckedChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	chkDetay (Denetim sekmesindeki “Detay Göster” onay kutusu)
Çağırdığı Yöntemler	
Açıklama	Seçime göre detaylı ya da normal sonuç listesi görüntülenir.

Yöntem	private void rbtZemberek_CheckedChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	rbtZemberek (Denetim sekmesindeki, Biçimbirimsele Çözümleyici grubundaki “Zemberek” radyo düğmesi)
Çağırdığı Yöntemler	
Açıklama	Zemberek Biçimbirimsele Çözümleyicisi seçilince, sadece çözümlenmiş dosyalar üzerinde işlem yapılmaktadır. Bu yüzden Dosya Türü seçim grubunu pasifleştirmektedir.

Yöntem	private void rbtMorfolojik_CheckedChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	rbtZemberek (Denetim sekmesindeki, Biçimbirimsel Çözümleyici grubundaki “Morfolojik Analiz (Eryiğit)” radyo düğmesi)
Çağırdığı Yöntemler	
Açıklama	Morfolojik Analiz Biçimbirimsel Çözümleyicisi seçilince, hem çözümlenmiş hem de asıl dosyalar üzerinden işlem yapılabilmektedir. Bu yüzden Dosya Türü seçim grubunu aktifleştirmektedir.

Yöntem	private void rbtBicimbirimsel_CheckedChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	rbtBicimbirimsel (Denetim sekmesindeki, Biçimbirimsel Çözümleyici grubundaki “Biçimbirimsel Çözümleyici (Of lazer)” radyo düğmesi)
Çağırdığı Yöntemler	
Açıklama	Biçimbirimsel Çözümleyici (Of lazer) seçilince, sadece çözümlenmiş dosyalar üzerinde işlem yapılmaktadır. Bu yüzden Dosya Türü seçim grubunu pasifleştirmektedir.

Yöntem	private void btnSatir_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnSatir (İşlemler sekmesindeki “Dosyayı Satırlara Ayır” tuşu)
Çağırdığı Yöntemler	DosyaIsmiAta, DosyayıSatirlaraDonustur
Açıklama	Kullanıcı tarafından seçilen bir dosyayı okuyup, dosyayı satırlara ayıran yöntem çağırılarak yeni bir dosya oluşturur. Yeni dosyanın ismi asıl dosya isminin başına “satirli_” sözü getirilerek oluşturulmaktadır.

Yöntem	private void btnCozumle_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnCozumle (İşlemler sekmesindeki “Dosyayı Biçimbirimsel Çözümle” tuşu)
Çağırdığı Yöntemler	DosyaIsmiAta, DosyayıSatirlaraDonustur, MorfolojikAnaliz
Açıklama	Kullanıcı tarafından seçilen bir dosyayı okuyup, seçime göre önce dosya satırlara ayrılmaktadır. Daha sonra Morfolojik Analiz (Eryiğit) uygulamasını çalıştırılan yöntem çağırılarak dosyadaki tüm sözcükler kök ve eklerine ayrılmaktadır. Yeni dosyanın ismi asıl dosya isminin başına “cozumlenmis_” sözü getirilerek oluşturulmaktadır.

Yöntem	private void btnButunlestir_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnButunlestir (İşlemler sekmesindeki, Türkçe Kurallarının Uygulanması grubundaki “Bütünleştir” tuşu)
Çağırdığı Yöntemler	KelimeyiOlustur
Açıklama	Ekrandan girilen çözümlenmiş sözcüğü Kelime Oluşturucu metoda gönderip, sonucu ekrana yazmaktadır.

Yöntem	private void btnTekliDenetle_Click()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnTekliDenetle (İşlemler sekmesindeki, Tekli Denetim grubundaki “Denetim” tuşu)
Çağırdığı Yöntemler	Denetle, KelimeyiOlustur
Açıklama	Tek bir sözcük için denetim işlemi yapmaktadır.

Yöntem	private void btnOgrenmeHesapla_Click ()
Parametreleri	object sender, System.EventArgs e
Çağırdığı Yöntemler	btnOgrenmeHesapla
Çağırdığı Yöntemler	KelimeyiAta, YapiyiOlustur, TutmayanYeriniGuncelle, OgrenmeliPuaniHesapla, UzaklikBul, Egıt
Açıklama	Ekrandan girilen iki sözcük için Öğrenme Puanını hesaplar.

Yöntem	private void btnBulanikPuanHesapla_Click ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnBulanikPuanHesapla (İşlemler sekmesindeki, Bulanık Mantık Puanlama grubundaki “Hesapla” tuşu)
Çağırdığı Yöntemler	KelimeyiAta, YapiyiOlustur, TutmayanYeriniGuncelle, BulanikMantikPuanHesapla
Açıklama	Ekrandan girilen iki sözcük için Bulanık Mantık Puanını hesaplar.

Yöntem	private void chkTerstenTekliDenetimBulanik_CheckedChanged()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	chkTerstenTekliDenetimBulanik (İşlemler sekmesindeki, Bulanık Mantık Puanlama grubundaki “Testen de Denetle” onay kutusu)
Çağırdığı Yöntemler	
Açıklama	Bulanık Mantık Puanlamasının Tersten de denetlenmesi seçimi içerir.

Yöntem	private void btnCikis_Click ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	btnCikis (“Çıkış” tuşu)
Çağırdığı Yöntemler	
Açıklama	Uygulamayı kapatmaktadır.

Yöntem	private void tabYazimDenetleyici_SelectedIndexChanged ()
Parametreleri	object sender, System.EventArgs e
Tetikleyen Kontrol	tabYazimDenetleyici (Ekrandaki Sekme sayfaları)
Çağırdığı Yöntemler	
Açıklama	Bir sekme sayfası seçilince, o sekmeye özel işlemler yapılmaktadır. Bu işlemler durum çubuğunun görünümü, ve Kaynak Küme Listesinin seçimi işlemleridir. Durum çubuğu sadece Denetim sekmesinde görüntülenmektedir. Herhangi bir sekmede Kaynak Küme Listesinden bir madde seçilmişse, diğer sekmelerdeki Kaynak Küme Listelerine (eğer hiçbir seçim yapılmamışsa) de bu seçim atanır.

Genel amaçlı yöntemler, kullanıcı arayüzündeki kontroller tarafından tetiklenen yöntemler tarafından çağırılan, kendine özgü işlevleri olan yöntemlerdir. Bu

yöntemler dosya işlemleri, veritabanı işlemleri ve diğer işlemler şeklinde üç grupta incelebilmektedir.

Yöntem	private void DosyaIsmiAta ()
Parametreleri	string tur
Çağırıldığı Yöntemler	btnEgitim_Click, btnDenetim_Click, btnSatir_Click, btnCozumle_Click
Çağırıldığı Yöntemler	
Açıklama	Kullanıcıya dosya seçimi yaptırmak üzere, dosya seçim ekranı açılıp, seçilen dosya ismi dizini ile alınmaktadır.

Yöntem	private void DosyayiSatirlaraDonustur ()
Parametreleri	string dosyaAdiDizinli,string dosyaAdi
Çağırıldığı Yöntemler	btnDenetim_Click, btnSatir_Click, btnCozumle_Click
Çağırıldığı Yöntemler	KatariDuzenle
Açıklama	Parametre olarak gelen dosya okunup, dosya içindeki harf dışındaki karakterleri elenip, her sözcük bir satıra gelecek şekilde, yeni bir dosya oluşturulur. Yeni dosyanın ismi asıl dosya isminin başına “satirli_” sözü getirilerek oluşturulmaktadır.

Yöntem	private void MorfolojikAnaliz ()
Parametreleri	string dosyaAdiDizinli,string dosyaAdi, ref string yeniDosyaAdi
Çağırıldığı Yöntemler	btnDenetim_Click, btnCozumle_Click
Çağırıldığı Yöntemler	
Açıklama	Morfolojik Analiz (Eryiğit) uygulaması çalıştırılarak, çözümlenmiş yeni dosya elde edilmektedir.

Yöntem	public int KaynakKumeGetir
Parametreleri	string sorguCumlesi
Çağırıldığı Yöntemler	KaynakKumeBilgileri
Çağırıldığı Yöntemler	
Açıklama	Parametre olarak gelen seçim sorgusu veritabanı üzerinde çalıştırılır. Sonuç olarak, dönen değer döndürülür.

Yöntem	public int KaynakKumeGuncelle
Parametreleri	string sorguCumlesi
Çağırıldığı Yöntemler	KaynakKumeBilgileri
Çağırıldığı Yöntemler	
Açıklama	Parametre olarak gelen kayıt güncelleme sorgusu veritabanı üzerinde çalıştırılır.

Yöntem	public int KaynakKumeListele
Parametreleri	string sorguCumlesi, DataGrid dg
Çağırıldığı Yöntemler	KaynakKumeBilgileri
Çağırıldığı Yöntemler	
Açıklama	Parametre olarak gelen listeleme sorgusu veritabanı üzerinde çalıştırılır. Parametre olarak gelen veri listesine atanır.

Yöntem	public int KaynakKumeTemizle
Parametreleri	
Çağırıldığı Yöntemler	btnEgitim_Click, btnSil_Click
Çağırdığı Yöntemler	
Açıklama	Parametre olarak gelen silme sorgusu veritabanı üzerinde çalıştırılır.

Yöntem	private void KaynakKumeBilgileri
Parametreleri	bool guncelle
Çağırıldığı Yöntemler	btnEgitim_Click, btnGetir_Click , btnSil_Click
Çağırdığı Yöntemler	KaynakKumeGetir, KaynakKumeGuncelle, KaynakKumeListele
Açıklama	Kaynak kümeler listelenir, kaynak kümelere ait sayısal değerler sorgulanır, ekranda ilgili yerlere bu değerler atanır. Parametre olarak gelen seçime göre, kaynak kümelerinin kullanım oranı alanlarını güncellemek üzere sorgu çalıştırılır.

Yöntem	private string KelimeyiOlustur
Parametreleri	string kelime
Çağırıldığı Yöntemler	btnDenetim_Click, btnButunlestir_Click, btnTekliDenetle_Click
Çağırdığı Yöntemler	
Açıklama	Türkçe'nin ses kurallarına göre, çözümlenmiş sözcükten, kurallı sözcük oluşturulur. Örneğin kitap-ı sözcüğünden, kitabı sözcüğünü oluşturur.

Sınıflara ait yöntemler, o sınıfa ait değişkenler ve parametre olarak gelen değişkenler üzerinden işlem yapmaktadır.

Eğitim sınıfına ait yöntemler aşağıda açıklanmaktadır.

Yöntem	public int Egit()
Parametreleri	
Çağırıldığı Yöntemler	btnDenetim_Click
Çağırdığı Yöntemler	
Açıklama	Eğitim sonuçlarına göre sistemin ağırlıkları elde edilir.

Yöntem	public DataRow Denetle
Parametreleri	string s, DataTable tblKok,DataTable tblEk,char klavye_tipi,bool terstenDenetle, ArrayList lstIncelenen,DataRow drYeni
Çağırıldığı Yöntemler	btnDenetim_Click, btnTekliDenetle_Click
Çağırdığı Yöntemler	KokEkAyristir, PuaniGetir, EkPuaniniBul, UzaklikBul
Açıklama	Denetleme işleminin yapıldığı yöntemdir. Parametre olarak gelen kelime için kök kaynak kümesinde en yüksek puanı veren kelime seçilir.

Yöntem	float PuaniGetir
Parametreleri	string klm1, string klm2,ref int uzunluk1,ref int uzunluk2, ref int MC,decimal kullanımOrani
Çağırıldığı Yöntemler	Denetle
Çağırdığı Yöntemler	KelimeyiAta, YapiyiOlustur, TutmayanYeriniGuncelle, KelimeyiGetir, TumPuaniHesapla
Açıklama	İki kelime arasındaki benzerlik puanı bulunur. Puan, bulanık mantık puanı, öğrenme puanı ve kullanım oranından elde edilir.

Yöntem	float EkPuaniniBul
Parametreleri	String[] sMorf,DataTable tblEk,ref string Bulundu_ek,ref string geccBulundu_ek,string karsilastirilan_kelime
Çağırıldığı Yöntemler	Denetle
Çağırdığı Yöntemler	UnluleriBul, SonUnluyuBul, UnluMu, KelimeyiAta, YapiyiOlustur, TutmayanYeriniGuncelle, TumPuaniHesapla, UcUnsuzYanyanaOlusuyorMu
Açıklama	Ek ikilisi için ek kaynak kümesindeki en yakın ek ikilisi bulunur. Kaynak kümedeki ek ikilileri ile arasındaki puan elde edilir. En yüksek puanı sağlayan ek ikilisi önerilir. Puan, bulanık mantık puanı, öğrenme puanı ve kullanım oranından elde edilir.

Yöntem	public String[] KokEkAyrıştır
Parametreleri	string kelime
Çağırıldığı Yöntemler	btnEgitim_Click, Denetle
Çağırdığı Yöntemler	
Açıklama	<kök>+<ek>+<ek>... şeklinde kök ve eklerine ayrılmış sözcüğü “+” karakterinden ayırarak katar dizisine dönüştürür.

Dizgi İşlemleri ayrı bir Dizgi İşlemleri sınıfında yer almaktadır. Bu sınıfa ait yöntemler aşağıda sıralanmaktadır.

Yöntem	public String[] KatariDuzenle
Parametreleri	string kelime, bool kokBicimi
Çağırıldığı Yöntemler	DosyayiSatirlaraDonustur,btnDenetim_Click, btnEgitim_Click
Çağırdığı Yöntemler	
Açıklama	Gelen katar dizisinde harf dışındaki karakterlerden temizler. Parametrede gelen işlem seçimine göre, ya katarı boşluk üzerinden kelimelere ayırır, ya da “#” karakterine göre ayırıştırır (Çözümlemiş dosyalarda satırlar, <kök>+<ek>+..#<kök>+<ek>+.. şeklinde yer almaktadır. Bu ayırıştırma işlemi, satırı çözümleme sözcüklerine ayırmaktadır).

Yöntem	public string TersCevir
Parametreleri	string kelime
Çağırıldığı Yöntemler	Denetle, TersCevir (Özyineli olarak kendi kendini çağırmaktadır)
Çağırdığı Yöntemler	TersCevir (Özyineli olarak kendi kendini çağırmaktadır)
Açıklama	Parametre olarak gelen kelime tersine çevrilir. Örneğin “denetim” sözcüğü “mitened” halini alır.

Yöntem	public char SonUnluyuBul
Parametreleri	string kelime
Çağırıldığı Yöntemler	btnEgitim_Click, EkPuaniniBul
Çağırdığı Yöntemler	
Açıklama	Parametre olarak gelen kelimedeki son ünlü harf döndürülür.

Yöntem	public bool UcUnsuzYanyanaOlusuyorMu
Parametreleri	string kelime
Çağırıldığı Yöntemler	EkPuaniniBul
Çağırıldığı Yöntemler	UnluMu
Açıklama	Parametre olarak gelen kelimedede içinde üç ünsüzün yanyana gelip gelmediği kontrol edilir.

Yöntem	public bool UnluMu
Parametreleri	string harf
Çağırıldığı Yöntemler	EkPuaniniBul, UcUnsuzYanyanaOlusuyorMu
Çağırıldığı Yöntemler	
Açıklama	Parametre olarak gelen harfin ünlü olup olmadığı döndürülür.

Harf sınıfına ait yöntemler aşağıda açıklanmaktadır.

Yöntem	public int TPLOludur
Parametreleri	snfYer parpos
Çağırıldığı Yöntemler	TumPuaniHesapla, BulanikMantikPuanHesapla
Çağırıldığı Yöntemler	TPLEkle
Açıklama	Yeri Tutmayan Harf Çifti Listesini oluşturulur.

Yöntem	public int TPLEkle
Parametreleri	snfYeriTutmayanHarfCifti pTutmyn
Çağırıldığı Yöntemler	TPLOludur
Çağırıldığı Yöntemler	
Açıklama	Yeri Tutmayan Harf Çifti Listesine uzaklık sırasında göre yeni bir Yeri Tutmayan Harf Çifti sınıfı eklenir.

Yöntem	YPLPuanHesapla
Parametreleri	
Çağırıldığı Yöntemler	TumPuaniHesapla, BulanikMantikPuanHesapla
Çağırıldığı Yöntemler	
Açıklama	Yer Puanı hesaplanır. Harfler arasındaki uzaklıklara göre, yer puanı elde edilir.

Kelime sınıfına ait yöntemler aşağıda açıklanmaktadır.

Yöntem	public void KelimeyiAta
Parametreleri	String pKelime
Çağırıldığı Yöntemler	PuaniGetir, EkPuaniniBul, btnBulanikPuanHesapla_Click
Çağırıldığı Yöntemler	
Açıklama	Parametrede gelen değer, kelimenin içeriğine atanır.

Yöntem	public string KelimeyiGetir
Parametreleri	
Çağırıldığı Yöntemler	PuaniGetir
Çağırıldığı Yöntemler	
Açıklama	Kelimenin içeriği döndürülür.

Yöntem	public int YapiyiOlustur
Parametreleri	
Çağırıldığı Yöntemler	PuaniGetir, EkPuaniniBul, btnBulanikPuanHesapla_Click
Çağırdığı Yöntemler	YeriGuncelle
Açıklama	Kelime içindeki harfler tek teke ele alınıp, her biri için, Harf Sınıfı ve Yer sınıfı oluşturularak, Harf listesi oluşturulur. Birden fazla kez yer alan harfler için yeni harf sınıfı oluşturulmaz, sadece varolan harf sınıfına yeni yer sınıfı eklenir.

Yöntem	public int YeriGuncelle
Parametreleri	int i
Çağırıldığı Yöntemler	YapiyiOlustur
Çağırdığı Yöntemler	
Açıklama	Kelimede birden fazla kez yer alan harfler için, ilgili harf sınıfına yeni yer sınıfı eklenir.

Yöntem	public int TutmayanYeriniGuncelle
Parametreleri	snfHarf pCls
Çağırıldığı Yöntemler	PuaniGetir, EkPuaniniBul, btnBulanikPuanHesapla_Click
Çağırdığı Yöntemler	
Açıklama	İki harfin iki kelimedeki yerleri karşılaştırılır, tutmuyorsa yeri tutmayanlar listesine eklenir, tutuyorsa tutan sayısı 1 artırılır.

Yöntem	public float TumPuaniHesapla
Parametreleri	snfKelime pKelime,snfEgitim pEgitim
Çağırıldığı Yöntemler	PuaniGetir, EkPuaniniBul
Çağırdığı Yöntemler	TPLOlustur, YPLPuanHesapla, YSAPLOlustur, OgrenmePuaniHesapla
Açıklama	İki Kelime için, bulanık mantık puanı, öğrenme puanı, kullanım oranları değerleri toplamı, toplam puan olarak döndürülür.

Yöntem	public float OgrenmeliPuaniHesapla
Parametreleri	snfKelime pKelime, snfEgitim pEgitim, ref float OgrenmePuani
Çağırıldığı Yöntemler	btnOgrenmeHesapla_Click
Çağırdığı Yöntemler	TPLOlustur, YPLPuanHesapla
Açıklama	Öğrenme Puanı ile Bulanık mantık puanı hesaplanır, toplamaları döndürülür.

Yöntem	public float TumPuaniHesapla
Parametreleri	snfKelime pKelime,snfEgitim pEgitim
Çağırıldığı Yöntemler	PuaniGetir, EkPuaniniBul
Çağırdığı Yöntemler	TPLOlustur, YPLPuanHesapla, YSAPLOlustur, OgrenmePuaniHesapla
Açıklama	İki Kelime için, bulanık mantık puanı, öğrenme puanı, kullanım oranları değerleri toplamı, toplam puan olarak döndürülür.

Yöntem	public int YSAPLOlustur
Parametreleri	snfKelime pKelime
Çağırıldığı Yöntemler	TumPuaniHesapla
Çağırdığı Yöntemler	
Açıklama	Farklı harf çiftlerinden, öğrenme yapısı hazırlanır.

Yöntem	public float OgrenmePuaniHesapla
Parametreleri	snfEgitim pEgitim
Çağırıldığı Yöntemler	TumPuaniHesapla
Çağırıldığı Yöntemler	
Açıklama	Sistemin ağırlıkları ve iki kelimededen elde edilen değerler ile öğrenme puanı hesaplanır.

Klavye sınıfına ait yöntem aşağıda açıklanmaktadır.

Yöntem	public float UzaklikBul
Parametreleri	char ch1,char ch2,char tip
Çağırıldığı Yöntemler	Denetle
Çağırıldığı Yöntemler	
Açıklama	İki harfin klavyedeki yerleri arası uzaklığı, iki nokta arası uzaklık formülünden bulunur. İki nokta arası uzaklık formülü 1. nokta P1(x1, x2) ve 2. nokta P2(x2, y2) olmak üzere, $KU = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}$

4.3 Kullanılan Ortam

Uygulama, Microsoft Development Environment 2003 üzerinde, Microsoft .NET Framework 1.1 yazılım geliştirme platformunda C# programlama dili kullanılarak yazılmıştır. Arayüz bir Windows Form Uygulaması olarak tasarlanmıştır. Veritabanı olarak, Microsoft Office Access 2003 kullanılmıştır.

.NET Framework'ün tercih edilme sebebi, arayüz tasarımının kolaylığı, farklı programlama dili seçeneklerinin olması ve uygulama geliştirmeye yönelik birçok ortamı aynı yapı içerisinde barındırmasıdır.

4.4 Uygulamanın Çalışması

Uygulama 3 ana kısımdan oluşmaktadır: Eğitim, Denetim ve İşlemler. Bu kısımlar ekran üzerinde 3 ayrı sekmede yer almaktadır. Eğitim sekmesi altında eğitim yapılabilir, eğitim ile elde edilmiş Kök ve Ek Kaynak kümeleri incelenebilir ya da silinebilir. Eğitim sekmesi ekran görüntüsü Şekil 4.4'de yer almaktadır.

Türkçe Yazım Denetleyici

EĞİTİM | DENETİM | İŞLEMLER | **TÜRKÇE YAZIM DENETLEYİCİSİ** | Çıkış

Biçimbirimsel Çözümleyici

☐ Zemberek

☒ Morfolojik Analiz (Eryiğit)

☐ Biçimbirimsel Çözümleyici (Ofıazer)

Eğitim İşlemleri

Kaynak Küme Listesi: **arkeoloji_MorfolojikAnaliz**

Dosya Türü: ☐ Çözümlemiş Dosya ☒ Asıl Dosya

Eğitime Başla ☒ Önce Eski Kaynak Kümeleri Sil

Dosya: **cozumlenmis_arkeoloji.txt**

Kaynak Kümeleri

Toplam Kök Sayısı :	25.441
Farklı Kök Sayısı :	4.627
Toplam Ek İkili Sayısı :	41.397
Farklı Ek İkili Sayısı :	1.396

Kaynak Kümeleri Getir Kaynak Kümeleri Temizle

Kök Kaynak Kümesi (arkeoloji_MorfolojikAnaliz_Kökl)			Ek Kaynak Kümesi (arkeoloji_MorfolojikAnaliz_Eklr)				
kök	kullanım_sayısı	kullanım_oranı	on_sesler	ek1	ek2	ek_ikilisi_kullanım_sayısı	ek_ikilisi_kullanım_oranı
ney	4	0,0002	ao	tik		34	0,0008
ne	5	0,0002	aio	ti	k	116	0,0028
içinde	5	0,0002	aijeio	k		144	0,0035
geçildiğin	2	0,0001	ouai	u		371	0,009
geçildiği	2	0,0001	aeoiiüü	i		1354	0,0327
geçildik	2	0,0001	iauo	da	ki	115	0,0028
geçil	2	0,0001	iaüeuüü	ki		254	0,0061
sapta	16	0,0006	iauo	nda	ki	61	0,0015
maliye	1	0	iaou	n	da	187	0,0045
apısında	4	0,0002	ioaeuüü	n		2460	0,0594
apısın	2	0,0001	ieüüiaüü	in		1140	0,0275
apısı	4	0,0002	i	miz	in	1	0
apıs	4	0,0002	i	ler	imiz	2	0
açı	11	0,0004	i	imiz	in	1	0
verim	13	0,0005	iauo	da		495	0,012
kuzey	18	0,0007	auoie	an		391	0,0094
suruy	3	0,0001					

Şekil 4.4: Uygulamanın Eğitim sekmesi görünümü

Bu kısımda 3 grup ve sonuçların listelendiği 2 tane liste bulunmaktadır. Birinci grup Biçimbirimsel Çözümleyici grubudur (Şekil 4.5). İçerisinde eğitim için kullanılacak biçimbirimsel çözümleyici uygulaması seçilebilmektedir.

Biçimbirimsel Çözümleyici

☐ Zemberek

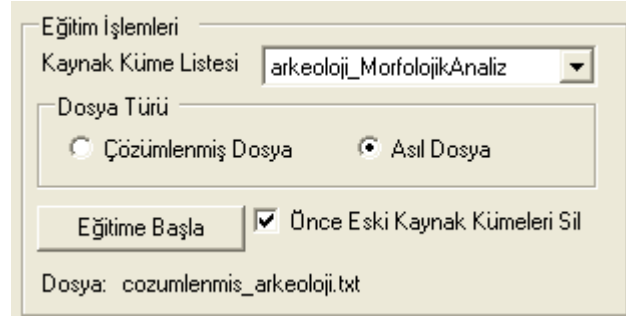
☒ Morfolojik Analiz (Eryiğit)

☐ Biçimbirimsel Çözümleyici (Ofıazer)

Şekil 4.5: Biçimbirimsel Çözümleyici grubu

İkinci grup Eğitim İşlemleri grubudur (Şekil 4.6). İçerisinde eğitim ile ilgili seçenekler bulunmaktadır. İlk seçenek eğitimin yapılacağı veritabanı nesnelerinin listesidir. Veritabanında bulunan tüm nesneler burada listelenmektedir. Böylece farklı kaynak kümeleri üzerinde çalışma imkanı bulunmaktadır. Her Kaynak Kümesine ait bir adet kök, bir adet ek kaynak kümesi yer almaktadır. Bu grup içerisinde yer alan Dosya Türü grubu, eğitimin yapılacağı dosyanın türünü

belirtmektedir. Çözümlemiş bir dosya ise, doğrudan eğitime katılabilir. Dosyanın aslı ise, önce harf dışındaki karakterlerden temizlenmeli, her sözcük bir satıra gelecek şekilde satırlara ayrılmalı ve Morfolojik Analiz (Eryiğit) uygulaması ise biçimbirimsel çözümlemelidir. Sadece Morfolojik Analiz uygulaması, uygulama ile tümleştirilmiş olduğu için, asıl dosya seçeneği, sadece bu çözümleyici için aktif olmaktadır. Diğer çözümleyiciler ile çalışılırken, önceden bu çözümleyiciler ile (uygulamadan bağımsız olarak) elde edilmiş çözümlemiş dosyalar kullanılmaktadır. “Eğitime Başla” tuşu ile kullanıcıya dosya seçimi yaptırılır ve eğitim işlemine başlanır. “Önce Eski Kaynak Kümeleri Sil” onay kutusu seçili ise, bu kaynak kümesine ait eski değerler eğitimden önce silinir. Aksi takdirde eğitimle elde edilen değerle eski kaynak kümesine eklenir. Kullanıcı tarafından seçilmiş dosyanın ismi de yine bu grup içerisinde yazılır.



Şekil 4.6: Eğitim İşlemleri grubu

Kaynak Kümeleri grubu altında da varolan kaynak kümeleri ya da eğitimi bitmiş kaynak kümeleri hakkında bilgi edinilebilir (Şekil 4.7). “Kaynak Kümeleri Getir” tuşu ile Kaynak Küme Listesinde seçili olan kaynak küme için Kök ve Ek kümeleri sayfanın alt tarafında yer alan listelerde listelenir (Şekil 4.8). “Kaynak Kümeleri Sil” tuşu ile de seçili olan kaynak kümesi silinir. Kaynak kümesi ile ilgili toplam kök ve ek sayıları, birbirinden farklı (birden fazla kez geçmiş olanlar tek sayılarak) kök ve ek sayıları da yine bu grup içerisinde belirtilir.

Kaynak Kümeleri	
Toplam Kök Sayısı :	25.441
Farklı Kök Sayısı :	4.627
Toplam Ek İkili Sayısı :	41.397
Farklı Ek İkili Sayısı :	1.396
Kaynak Kümeleri Getir Kaynak Kümeleri Temizle	

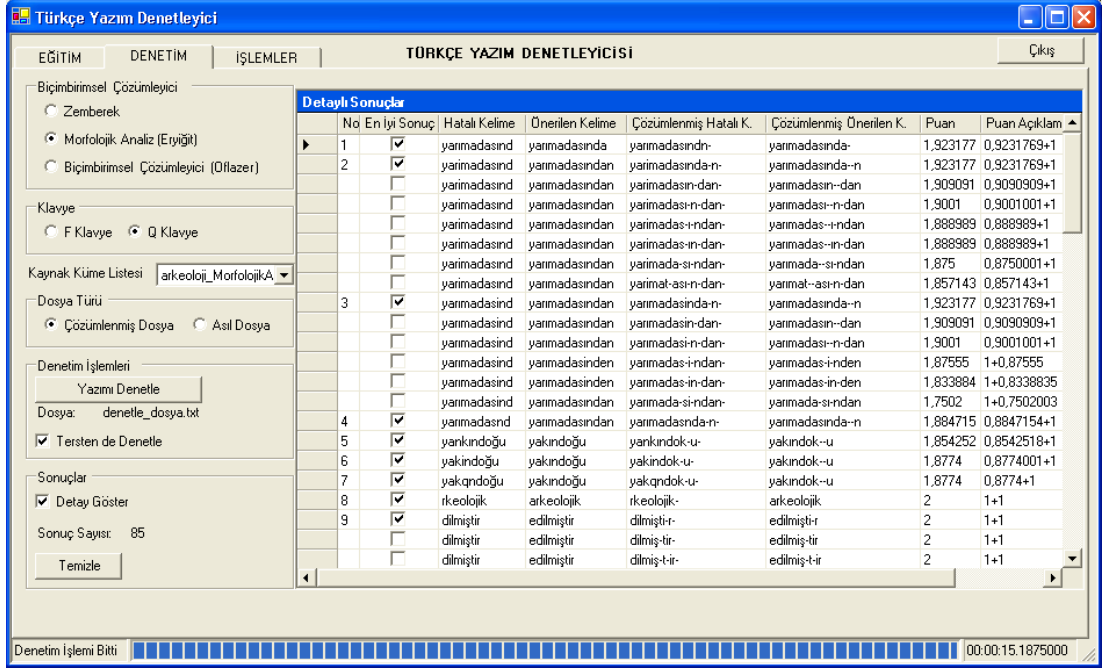
Şekil 4.7: Kaynak Kümeleri grubu

Kök Kaynak Kümesi (arkeoloji_MorfolojikAnaliz_Kokler)			Ek Kaynak Kümesi (arkeoloji_MorfolojikAnaliz_Ekler)				
kök	kullanım_sayısı	kullanım_oranı	on_sesliler	ek1	ek2	ek_ikilisi_kullanım_sayısı	ek_ikilisi_kullanım_oranı
► ney	4	0,0002	► ao	tık		34	0,0008
ne	5	0,0002	áo	ti	k	116	0,0028
► içinde	5	0,0002	áuieio	k		144	0,0035
geçildiğin	2	0,0001	ouai	u		371	0,009
geçildiği	2	0,0001	aeoiüü	i		1354	0,0327
geçildik	2	0,0001	iauo	da	ki	115	0,0028
geçil	2	0,0001	iaüieuü	ki		254	0,0061
sapta	16	0,0006	iauo	nda	ki	61	0,0015
maliye	1	0	iaou	n	da	187	0,0045
► açısında	4	0,0002	ioaeüüü	n		2460	0,0594
► açısın	2	0,0001	ieoiüüü	in		1140	0,0275
► açısı	4	0,0002	i	miz	in	1	0
► açıs	4	0,0002	i	ler	imiz	2	0
► açı	11	0,0004	i	imiz	in	1	0
verim	13	0,0005	iauo	da		495	0,012
kuzey	18	0,0007	auoie	an		391	0,0094
suriv	3	0,0001					

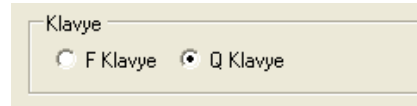
Şekil 4.8: Kaynak Kümeleri Listesi

Denetim sekmesi altında denetim işlemi yapılabilir, denetim sonuçları izlenebilir (Şekil 4.9). Bu sekme altında denetim işlemi ile ilgili seçeneklerin yer aldığı gruplar ve sonuçları izleme bilgi ve işlemlerini içeren Sonuçlar grubu bulunmaktadır.

İlk seçenek grubu denetimin hangi biçimbirimsel çözümleyici üzerinden yapılacağı ile ilgilidir. Eğitim kısmında anlatıldığı gibi bu seçeneğe göre Dosya Türü seçimi aktif/pasif hale gelebilir ve sadece Morfolojik Analiz (Eryiğit) biçimbirimsel çözümleyici seçeneği ile aktif olabilir. Denetim kısmında da yer alan Dosya Türü seçeneği Asıl Dosya seçildiği takdirde, denetimden önce dosyanın harf dışındaki karakterlerden elenmesi, kelimelerin satırlar halinde oluşturulması ve çözülmesi işlemi ele alınır. Yine eğitim kümesinde de yer alan Kaynak Kümesi listesi seçilerek, sistemde varolan hangi kaynak kök ve ek kümeleri üzerinden denetim yapılacağı bilgisi seçilir. Ayrıca denetim için önem taşıyan klavye türü seçeneği de “Klavye” başlıklı grup altında yer almaktadır (Şekil 4.10). Seçilen klavyeye göre puanlama işlemi özelleşmektedir.

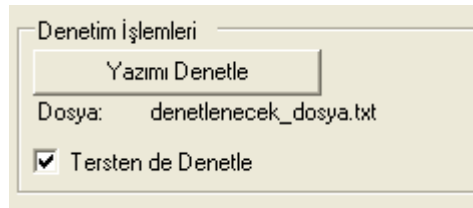


Şekil 4.9: Uygulamanın Denetim sekmesi görünümü



Şekil 4.10: Denetim sekmesi Klavye grubu

Denetim İşlemleri grubu altında (Şekil 4.11) yer alan “Yazımı Denetleme” tuşu ile kullanıcıdan denetimi yapılacak dosya alınır ve denetleme işlemine başlanır. Denetlenen dosya ismi “Dosya” etiketi yanında belirtilir. Denetlenecek sözcüklerin Kaynak Kümelerinde sözcükler ile karşılaştırılması sırasında varsayılmış olarak, soldan sağa yer numarası atanmaktadır. Bu grupta yer alan “Tersten de Denetle” onay kutusunun seçilmesi ile, hem soldan sağa, hem de sağdan sola sıra numarası verilerek, denetim hem düz hem tersten yapacak şekilde çalıştırılır. Bu seçenek kelime başında yapılmış hatalar için bir iyileştirme anlamı taşımaktadır.



Şekil 4.11: Denetim sekmesi Denetim İşlemleri grubu

Denetleme sırasında ekranında en altında yer alan durum çubuğunda denetleme işleminin durumu ve geçen süre izlenebilir.

Denetleme sonuçları ekranın sağında yer alan listede görüntülenmektedir. Sonuçlar grubunda bu sonuçlara ait sayı bilgisi ve detay seçeneği yer almaktadır. Detay onay kutusu seçildiği takdirde Şekil 4.12’de görüldüğü gibi hatalı ve önerilmiş kelimelerin bütünleşik halleri yanında, çözümlenmiş halleri de yer almaktadır. Elde edilmiş puan ve puanın kök+ek puanı şeklinde detaylandırılmış hali de detaylı sonuçlar listesinden izlenebilir. Ayrıca bir hatalı kelime için farklı önermeler yer almaktadır. Bunlardan en yüksek puanı almış, en iyi önerme liste içinde onay kutusu seçilerek belirtilmiştir. Hatalı kelimenin farklı çözümlenmeleri için de aynı bütünleşik yapıya sahip fakat farklı çözümlenmelere sahip seçenekler de listelenmektedir. Bu yüzden detaylı sonuç listesinde bütünleşik halleri aynı olan, fakat farklı çözümlenmelere sahip tekrarlı önermeler de izlenebilmektedir. Bu durum sistemin çalışmasını daha detaylı inceleyebilmek açısından önemlidir. Her ayrı hatalı sözcük denetimi yanında No alanı ile sıralanmıştır.

Detaylı Sonuç onay kutusu seçimi kaldırılınca daha sade bir sonuç listesi görüntülenmektedir. Şekil 4.13’de görülen bu listede hatalı ve önerilen kelimelerin bütünleşmiş halleri, tekrarlı önermeler teke indirilerek listelenmektedir. Yine detaylı listede de olduğu gibi, aynı hatalı kelimeye ait farklı önermeler listelenmiş, en iyi önerme, aynı hatalılar için yapılan önermeler arasında en üstte, yanında onay kutusu seçili şekilde belirtilmiştir. Her ayrı hatalı sözcük denetimi yanında No alanı ile sıralanmıştır. Sonuçlar grubunda yer alan “Temizle” tuşu ile sonuç listeleri temizlenmektedir.

Detaylı Sonuçlar								
	No	En İyi Sonuç	Hatalı Kelime	Önerilen Kelime	Çözümlemiş Hatalı K.	Çözümlemiş Önerilen K.	Puan	Puan Açıklam
▶	1	☒	yarımadasiınd	yarımadasiında	yarımadasiındn-	yarımadasiında-	1,923177	0,9231769+1
	2	☒	yarımadasiınd	yarımadasiından	yarımadasiında-n-	yarımadasiında-n	1,923177	0,9231769+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiın-dan-	yarımadasiın--dan	1,909091	0,9090909+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiı-n-dan-	yarımadasiı-n-dan	1,9001	0,9001001+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiı-ından-	yarımadasiı-ından	1,888989	0,888989+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiın-dan-	yarımadasiın--dan	1,888989	0,888989+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiı-sından-	yarımadasiı-sından	1,875	0,8750001+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiı-ası-n-dan-	yarımadasiı-ası-n-dan	1,857143	0,857143+1
	3	☒	yarımadasiınd	yarımadasiından	yarımadasiında-n-	yarımadasiında-n	1,923177	0,9231769+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiın-dan-	yarımadasiın--dan	1,909091	0,9090909+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiı-n-dan-	yarımadasiı-n-dan	1,9001	0,9001001+1
		☐	yarımadasiınd	yarımadasiından	yarımadasiı-ından-	yarımadasiı-ından	1,87555	1+0,87555
		☐	yarımadasiınd	yarımadasiından	yarımadasiın-dan-	yarımadasiın--dan	1,833884	1+0,8338835
		☐	yarımadasiınd	yarımadasiından	yarımadasiı-sından-	yarımadasiı-sından	1,7502	1+0,7502003
	4	☒	yarımadasiınd	yarımadasiından	yarımadasiında-n-	yarımadasiında-n	1,884715	0,8847154+1
	5	☒	yakındoğu	yakındoğu	yakındok-u-	yakındok--u	1,854252	0,8542518+1
	6	☒	yakındoğu	yakındoğu	yakındok-u-	yakındok--u	1,8774	0,8774001+1
	7	☒	yakındoğu	yakındoğu	yakındok-u-	yakındok--u	1,8774	0,8774+1
	8	☒	rkeolojik	arkeolojik	rkeolojik-	arkeolojik	2	1+1
	9	☒	dilmiştir	edilmiştir	dilmiştir-	edilmiştir-	2	1+1
		☐	dilmiştir	edilmiştir	dilmiştir-tir	edilmiştir-tir	2	1+1
		☐	dilmiştir	edilmiştir	dilmiştir-tir	edilmiştir-tir	2	1+1

Şekil 4.12: Detaylı Denetim Sonucu Listesi

Sonuçlar				
	No	En İyi Sonuç	Hatalı Kelime	Önerilen Kelime
▶	1	☒	yarımadasiındn	yarımadasiında
	2	☒	yarımadasiından	yarımadasiından
	3	☒	yarımadasiından	yarımadasiından
		☐	yarımadasiından	yarımadasiından
	4	☒	yarımadasiından	yarımadasiından
	5	☒	yakındoğu	yakındoğu
	6	☒	yakındoğu	yakındoğu
	7	☒	yakındoğu	yakındoğu
	8	☒	rkeolojik	arkeolojik
	9	☒	dilmiştir	edilmiştir
	10	☒	arkolokları	arkeolokları
		☐	arkolokları	arkeolokları
	11	☒	ilgilendşren	ilgilendiren
	12	☒	amüzelerinde	müzelerinde
	13	☒	bilimsell	bilimsel
	14	☒	ğitim	eğitim
		☐	ğitim	eğitim
	15	☒	bilginin	bilginin
	16	☒	geçitirilmesi	geçitirilmesi
		☐	geçitirilmesi	geçitirilmesi
	17	☒	ayıda	kayıda
		☐	ayıda	yıda
	18	☒	uzamın	uzmanın

Şekil 4.13: Denetim Sonucu Listesi

Son sekme olan İşlemler sekmesinde uygulama içerisinde eğitim ve denetim kısımları içerisinde kullanılan işlemler için ayrı ayrı uygulamaları ele alınmaktadır. Böylece tümleşik kullanımları ile denetim/eğitim işlemlerinin yapıldığı alt işlevlerin

bağımsız kullanımları incelenebilmektedir. Şekil 4.14’de görülen İşlemler sekmesinde 6 ayrı grup yer almaktadır.

Türkçe Yazım Denetleyici

EGİTİM | DENETİM | **İŞLEMLER** | ÇIKIŞ

TÜRKÇE YAZIM DENETLEYİCİSİ

Dosya İşlemleri

Dosyayı Satırlara Ayr

Kaynak Dosya:

Satırlara Ayrılmış Dosya:

Dosyayı Biçimbirimsel Çözümle ☒ Önce Satırlara Ayr

Kaynak Dosya:

Çözümlemiş Dosya:

Türkçe Kurallarının Uygulanması

Çözümlemiş Biçim:

<kok>-<ek> (Örnek: kitap-ı)

Bütünleştir

Kurallı Şekilde Bütünleşmiş Biçim:

Klavye/Yazım Uzaklıkları

Klavye

☐ F Klavye ☒ Q Klavye

1. Harf 2. Harf

Uzaklığı Bul

Uzaklık:

Bulanık Mantık Puanlama

1. Sözcük

2. Sözcük

☒ Tersten de Denetle

Hesapla

Bulanık Mantık Puanı:

Öğrenme Puanı

1. Sözcük

2. Sözcük

Klavye

☐ F Klavye ☒ Q Klavye

Hesapla

Öğrenme Puanı:

Öğrenme+B.M. Puanı:

Tutan Ağırlığı	
Tutmayan Ağırlığı	
Karşılaşılma Ağırlığı	
Klavye Ağırlığı	

Tekli Denetim

Çözümlemiş Biçim:

<kok>+<ek>+<ek>... (Örnek: kitap+lar+ı)

Kaynak Küme Listesi:

Klavye

☐ F Klavye ☒ Q Klavye ☒ Tersten de Denetle

Denetle

Önerme

Şekil 4.14: Uygulamanın İşlemler Sekmesi

İlk grup Bulanık Mantık Puanlama grubudur. Şekil 4.15’de görüldüğü gibi, kullanıcı tarafından girilmiş olan iki sözcük için elde edilmiş bulanık mantık puanı, “Hesapla” tuşuna basılması ile ekrana yazılmaktadır. Bulanık mantık puanlaması normalde soldan sağa harflere yer numarası verilerek yapılan bir işlemdir. Fakat bazı durumlarda, harf ekleme veya çıkarma şeklinde yapılmış bir hatanın kelimenin baş tarafında olması, bulanık mantık puanını olumsuz yönde etkilemektedir. Bu yüzden kelimenin baş tarafındaki hatalara da duyarlı bir iyileştirme yapmak amacıyla, tersten denetim mantığı geliştirilmiştir. Şekil 4.15’de görüldüğü gibi başta yapılmış bir hatada normal bulanık mantık puanı çok düşük olurken, tersten bulanık mantık puanı yüksek çıkmaktadır.

Bulanık Mantık Puanlama

1. Sözcük: karşılaştırma

2. Sözcük: karşılaştırma

☒ Tersten de Denetle

Hesapla

Bulanık Mantık Puanı : 0,543956

Tersten Bulanık Mantık Puanı : 0,910256

Şekil 4.15: İşlemler sekmesi Bulanık Mantık Puanlama grubu

Tersi durumda ise, Şekil 4.16’da görüldüğü gibi kelimenin sonunda yapılmış bir hata, tersten bulanık mantık puanı etkilemezken, normal bulanık mantık puanını yüksek çıkarmaktadır. Denetleme sekmesi içinde tersten denetleme seçeneği seçili ise, hem normal bulanık puanı, hem de tersten bulanık puanı hesaplanıp, büyük olan kullanılmaktadır.

Bulanık Mantık Puanlama

1. Sözcük: karşılaştırma

2. Sözcük: karşılaştırmag

☒ Tersten de Denetle

Hesapla

Bulanık Mantık Puanı : 0,928572

Tersten Bulanık Mantık Puanı : 0,476191

Şekil 4.16: İşlemler sekmesi Bulanık Mantık Puanlama grubu, tersten denetleme

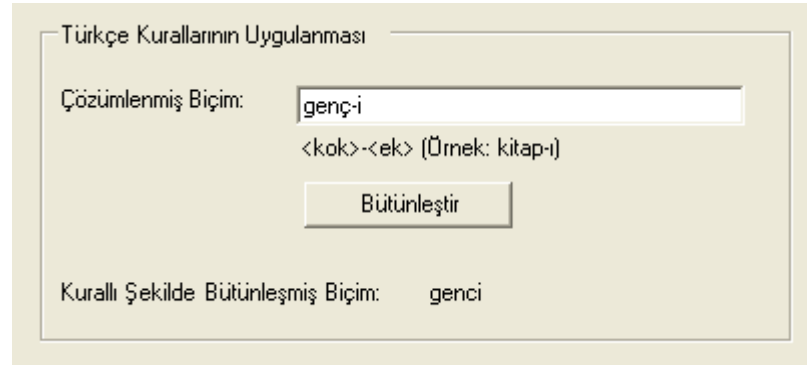
İkinci grup olan Tekli Denetim grubunda, çözümlenmiş haldeki bir hatalı sözcük için denetim işlemi yapılır. Genel Denetim kısmında olduğu gibi burada da kaynak alınacak Kaynak Küme ismi, Kaynak Küme Listesinden seçilmekte, Klavye Tipi ve Tersten Denetim onay kutusu seçenekleri belirlenmektedir. Denetle tuşu ile elde edilen sonuç Şekil 4.17’de görüldüğü gibi Önerme etiketi yanında yazılmaktadır.

Şekil 4.17: İşlemler sekmesi Tekli Denetim grubu

Dosya İşlemleri adı altında yer alan grupta ilk olarak seçilen bir dosyayı satırlara ayırma işlemi yapılmaktadır. “Dosyayı Satırlara Ayır” tuşuna basılarak, kullanıcıya dosya seçimi yaptırılıp, seçilen dosya harf dışındaki karakterlerden arındırılıp, her kelime bir satıra gelecek şekilde satırlara ayrılır. Ele alınan dosya ve sonuç dosyası ilgili alanlara yazılır. Diğer bir işlem olan “Dosya Biçimbirimsel Çözümle” tuşuna basılarak, kullanıcının seçtiği dosya uygulama ile tümleştirilmiş olan Morfolojik Analiz (Eryiğit) uygulaması ile çözülür. “Önce Satırlara Ayır” onay kutusu seçili ise, seçilen dosya önce satırlara dönüştürülür, daha sonra çözülür. Giriş ve sonuç dosya isimleri Şekil 4.18’de görüldüğü gibi ilgili alanlara yazılır.

Şekil 4.18: İşlemler sekmesi Dosya İşlemleri grubu

İşlemler sekmesindeki gruplardan biri olan “Türkçe Kurallarının Uygulanması” grubunda (Şekil 4.19), çözümlenmiş biçimdeki sözcüğün, bütünleştirilmesi ile alacağı şekil elde edilmeye çalışılır. Burada Türkçe’nin ses uyumu kuralları ele alınmış olup, bir köke ek geldiği zaman kökte oluşacak değişiklikler kurallar dahilinde işlenmiştir. Bu kısımda amaç, kök ve ekleri ayrı ayrı ele aldıktan sonra, birlikte oluşacak sonuç biçiminin kurallara uygun şekilde elde edilmesinin sağlanmasıdır. Ayrıca belli bazı kuralları uygulama içerisine katarak iyileştirme sağlanmaya çalışılmıştır.



Şekil 4.19: İşlemler sekmesi Türkçe Kurallarının Uygulanması grubu

İşlemler sekmesindeki diğer bir grup olan “Öğrenme Puanı” grubu, uygulama içerisinde kullanılan öğrenme işlemini modellemeye çalışmaktadır. Şekil 4.20’de görülen sözcük alanlarına girilmiş olan iki sözcük için bulanık mantık ve öğrenme puanı toplamı bulunur. Elde edilen puan belli bir seviyenin (0,749999) üzerinde ise bu puan üzerinden öğrenme yapılır. Bu seviye kontrolü sayesinde sadece anlamlı hatalar öğrenilir. Tamamen farklı iki sözcüğün karşılaştırılması bir hata durumunu yansıtmayacağı için böyle bir karşılaştırma bilgilerinin öğrenilmesi anlamlı değildir. İlk açılışta ağırlıklar 0, öğrenme puanı ise hata payından kaynaklanan çok küçük bir değere sahiptir. Her yeni puan bulma işleminde, karşılaştırılan sözcüklere sistemin ağırlıkları güncellenir. Bu ağırlıklar her denetleme sonrası ekranda ilgili yerlere yazılır. Böylece yapılan denetlemenin sistemin ağırlığını nasıl etkilediği izlenebilir. Örneğin o ana kadar çok fazla klavye hatası yapılsıysa Klavye ile ilgili ağırlık yüksek olacaktır. Burada dikkat edilmesi gereken ağırlıkların bulunması için gerekli şartları sağlayan karşılaştırmaların yapılmış olmasıdır. Bölüm 3.5.4.2’de anlatıldığı gibi öğrenme formülü (3.4) üzerinde ağırlıkların bulunması için 4 bilinmeyenli denklem 2 bilinmeyenli 2 kısıma ayrılmaktadır. Örneğin Tutan ve Tutmayan ağırlıklarını elde

etmek için sistem içinde Karşılaşılma ve Klavye/Yazım ağırlıkları aynı, puanları farklı iki karşılaştırma olmalıdır. Aynı şekilde Karşılaşılma ve Klavye/Yazım ağırlıklarını bulmak için sistemde Tutan ve Tutmayan ağırlıkları aynı, puanları farklı iki karşılaştırma olmalıdır. Bu şartlar sağlanmazsa, sistem denklemleri çözmeye yetecek veriye sahip olmadığından ağırlıkları güncelleyemeyecektir. Uygulamanın Denetim sekmesinden yapılan denetimde çok fazla sayıda karşılaştırma yapıldığı için bu durumla karşılaşmak neredeyse imkansızdır. Fakat burada modelleme amaçlı gerçekleştirilen öğrenme sırasında bu konuya dikkat edilmelidir.

Öğrenme Puanı :	0,0084233100
Öğrenme+B.M. Puanı :	0,8417567000
Tutan Ağırlığı	0,9999010000
Tutmayan Ağırlığı	-0,0000988841
Karşılaşılma Ağırlığı	-0,4308399000
Klavye Ağırlığı	3,4520820000

Şekil 4.20: İşlemler sekmesi Öğrenme Puanı grubu

Son grup olan “Klavye/Yazım Uzaklıkları” grubunda klavyedeki iki harf arası uzaklık bulunmaktadır. Şekil 4.21’de görülen grupta kullanılan klavyenin seçilebilmesi için Klavye tipinin belirleneceği seçim radyo düğmeleri bulunmaktadır.

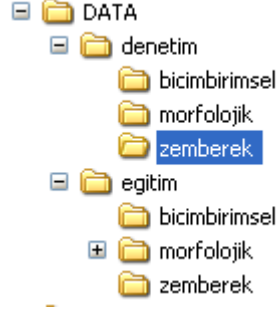
Uzaklık: 1,802776

Şekil 4.21: İşlemler sekmesi Türkçe Kurallarının Uygulanması grubu

Ekranın ortak kullanım alanındaki Çıkış tuşu ile uygulama sona erdirilir.

Dizin Yapısı

Uygulamada, seçilecek eğitim ve denetim dosyaları “eğitim” ve “denetim” isimli iki farklı dizin yapısında bulunmaktadır. Seçilen biçimbirimsel çözümleyiciye göre de bu dizinler altında üç farklı dizin oluşturulmuştur. Bu şekilde uygulama içerisinde kullanılacak dosyalar için işlem ve BÇ bazında bir sınıflandırma hedeflenmiştir. Bahsedilen dizin yapısı Şekil 4.22’de gösterilmektedir.



Şekil 4.22: Uygulama veri dosyaları dizin yapısı

5. BÖLÜM

KAYNAK TARAMASI

Yazım denetleyici konusunda yapılmış çok sayıda çalışmakta bulunmaktadır. Bu bölümde, bu konuda çalışma yapacaklar için bir kaynak taraması sunmak amacıyla, tez çalışması süresince incelenmiş çalışmalar gruplandırılmış, çalışmalar hakkında bilgi verilmeye çalışılmıştır.

Yazım denetleyici konusunda yapılmış çalışmalar çeşitli yöntemler ve dillere göre sınıflandırılabilir. Bazı çalışmalarda ise genel amaçlı denetleyiciler üretilmeye çalışılmıştır.

İncelenen kaynaklar doğrultusunda aşağıdaki sonuçlara varılmıştır.

Yazım denetleyiciler doğru sözcüğü tarama yöntemine göre ikiye ayrılmaktadırlar.

1. Sözlük Tabanlı Yazım Denetleyiciler: Bu konuda yapılmış çalışmalar aşağıda sıralanmıştır:

- “An expert system for automatically correcting OCR output – Kazem Taghva, Julie Borsack, Allen Condit” (OCR* sonuçlarını otomatik olarak düzelten uzman sistem) [6]: İngilizce için yapılmış, OCR ile elektronik hale dönüştürülmüş metinleri denetleme çalışmasıdır. İngilizce sözlük ve etki alanına özel sözlük kullanılmaktadır. Sözlüklerle hatalı sözcük arasındaki en kısa uzaklık (minimum edit distance) bulunarak, doğru sözcük önermesi yapılmaktadır. Ayrıca OCR’dan yapılan hatalara özel ek kontroller de kullanılmıştır.
- “Spelling Correction for Search Engine Queries – Bruno Martins, Mario J.Silva” (Arama motoru sorguları için hata düzeltici) [7]: Portekiz dili için, Internet üzerindeki arama motorlarına gönderilen arama sözcüklerindeki

*OCR: (Optical Character Recognition) Optik Karakter Tanıma

hataları bulmaya ve düzeltmeye yönelik yapılmış bir çalışmadır. Çalışmada yapılmış bir araştırmaya göre, yapılan aramaların %10-12'si hatalı yazım içermektedir. Sözlük tabanlı yapılan çalışmada, sözlükte bulunamayan sözcük için uygun önermeler türetilip, bu önermelerden en uygunu seçilmektedir. Uygun önerme türetimi için, hatalı sözcükten değişik yaklaşımlara (bir harf eksiltme, bir harf ekleme, harflerin yerini değiştirme, tekrarlı harfleri bire indirme, fazladan boşluk kullanımı ele almak için sözcüğü ikiye bölme vb.) göre yeni sözcükler türetilip, bu yeni sözcüğün sözlükte olup olmadığı kontrol edilmektedir. Sözcük sözlükte var ise önerilenler arasına alınmaktadır. En iyi önerme seçimi için öncelikle önerilenler içinde aksan hatasından türetilmiş hatalar alınır. Portekizce'de yapılan yazım hatalarının %20'sini aksan hataları oluşturmaktadır. Yoksa, sırasıyla sadece bir harf farkı ile türetilmiş ama bu farklı harflerin klavyede aynı satırda olan sözcükler, sadece bir harf farkı ile türetilmiş sözcükler vb. önerilenler arasından seçilir. En kötü ihtimalle önerilenler arasındaki en kısa sözcük en iyi önerme olarak seçilir. Bu çalışmadaki bir önemli nokta da başarımlarını arttırmak için sözlük yapısının TST (Ternary Search Trees, Üçlü Arama Ağacı) şeklinde tutulmasıdır.

2. İstatistik Tabanlı Yazım Denetleyiciler: Bu konuda yapılmış ve diğer istatistik tabanlı çalışmalara kaynaklık etmiş bazı temel çalışmalar aşağıda belirtilmiştir.

- “A Spelling Correction Program Based on a Noisy Channel Model – Mark D. Kernighan, Kenneth W. Church, William A. Gale” (Gürültülü Kanal Modeli tabanlı yazım düzeltme programı) [9]: Hata düzeltme üzerine yapılmış bir çalışmadır. Sadece Gürültü Kanal Modeline uygun hatalar için düzeltme yapılır. Gürültü Kanal Modeli anlam ya da dilbilgisi hatası değil, yazım sırasında karakter hatasıdır. Yazım sırasında parmağın iki tuşa birden basmasından oluşan hata buna bir örnek sayılabilir. Hatalı sözcükler farklı bir denetleme programı tarafından tespit edilip uygulamaya gönderilmektedir. Hatalı sözcük için bir harf ekleme/çıkarma, yer değiştirmesi, farklı harf yazma işlemleri uygulanarak ve elde edilen yeni kelimelerin geçerli kelimeler olup olmadığı sözlük ve büyük bir eğitim kümesi üzerinden test edilerek, aday kelimeler listesi oluşturulmaktadır. Bu aday kelimeler için puan hesaplanmıştır. Puan hesaplamasında büyük bir eğitim kümesinden yararlanılmış ve 4 işlem için 4 farklı hesaplama kullanılmıştır. Aday, harf

silme olduđu düşünölerek oluşmuşsa, bu tip hataların eğitim kümesinde yer alış sayısı bulunur. Örneğin hatalı kelime “dnetim”, hatanın harf silmeden kaynaklandığı varsayılarak oluşturulmuş aday kelime “denetim” ise eğitim kümesi içerisinde “en” yerine “n” yazma işleminin kaç kere gerçekleştiği sayısı bulunur. Aynı şekilde harf ekleme, yer değiştirerek yazma, yerine farklı harf yazma hatalarından kaynaklanan hatalar için de bu tip hataların eğitim kümesinde yeralma sayılarına göre puanlama yapılır. Puan, elde edilen sayının (aynı tip hatanın eğitim kümesinde yer alış sayısı), ele alınan harf çifti ya da harfin eğitim kümesinde toplam yer alma sayısına bölünmesiyle hesaplanır. Hataların eğitim kümesinde yer alma sayıları önceden tüm harf çiftleri için hesaplanıp, her işlem için kendine ait bir matriste saklanmıştır. Bu matrisler Good-Turing düzleme yöntemi ile düzlenmiştir. Puan hesabı sırasında bu matrizen ilgili hücre değerini okumak yeterli olmaktadır. Adayların aldığı puanlara göre en yüksek puanlı en öncelikli önerme olarak listelenir. Puanlamaya getirdiği farklı bakış açısı ile önemli bir çalışmadır.

- “Decision Lists For Lexical Ambiguity Resolution: Application To Accent Restoration In Spanish And French – David Yarowsky” (Çokyapılılık çözümünde karar ağaçlarının kullanımı) [28] : Karar ağaçları kullanılarak, istatistiksel tabanlı çokyapılılık çözme işlemi ele alınmıştır.
- “A Winnow-Based Approach to Context-Sensitive Spelling Correction- A. Golding, Dan Roth” (Bağlam duyarlı hata düzeltmeye Winnow tabanlı yaklaşım) [29]: Bağlama duyarlı hata düzeltme konusunda farklı algoritmalar temel alınarak yeni bir algoritma geliştirilmiştir.
- “A Bayesian hybrid method for context-sensitive spelling correction - A. Golding” (Bayesian hybrid yöntemi ile bağlama duyarlı hata düzeltimi) [30]: Bayesian sınıflandırması kullanılarak geliştirilmiş melez bir yöntem ile bağlama duyarlı hata düzeltimi, çok yapılılık çözümlemesi üzerine yapılmış bir çalışmadır.

Yukarıda bahsedilen temel çalışmalardan faydalınarak yapılan çalışmalara örnekler aşağıda açıklanmıştır:

- “A Statistical Approach to Automatic OCR Error Correction in Context, Xiang Tong, David A. Evans” (Otomatik OCR hata düzeltimine istatistiksel

bir yaklaşım) [11]: Bu çalışmada optik karakter tanıma programları ile elektronik hale geçirilmiş metinlerdeki hatalar düzeltilmeye çalışılmıştır. Bu amaçla farklı birkaç istatistiksel yöntem birleştirilerek sonuca ulaşmak hedeflenmiştir. Bu yöntemler hatalı sözcüğe önerilecek adayların belirlenmesi için kullanılan harf n-gramları*, sonuçları iyileştirmek amaçlı kullanılan sıkça karıştırılan harflerin oranlaması ve en iyi aday önermeyi bulmak için kullanılan Viterbi* algoritmasıdır.

- “Spelling Correction as an Iterative Process that exploits the collective knowledge of web users - Silviu Cucerzan, Eric Brill” (Web kullanıcı bilgileri kullanılarak tekrarlı yazım düzeltme) [3]: İnternet arama motorlarına yapılan kullanıcı girişlerinin kayıtları tutularak bunlardan toplanan istatistiksel bilgiler kullanılmaktadır. Bu bilgiler, kelimelerin kullanım sıklıkları, sıkça yapılan klavye hataları, ses bilgisi ya da anlamsal karmaşa sebebiyle oluşmuş hatalara ait bilgilerdir. Tüm kelimeleri doğru olduğu varsayılan bir kelime kümesi kaynak alınarak, bu küme içerisinde hatalı kelimeye en yakın kelime bulunmaya çalışılmaktadır. En yakın kelime, hatalı kelime ile arasındaki en küçük uzaklık değeri en az olandır. En kısa uzaklık için farklı değişik yöntemler denenmiştir.
- “The Development of a Statistical Spell Checker for Maltese - Ruth Mizzi” (Malta dili için istatistiksel yazım denetleyici geliştirimi) [10]: Çalışma, büyük bir sözcük kümesinden okunan kelimelerin öğrenilmesi, istatistiklerinin çıkarılması ve çıkarılmış bu istatistikler kullanılarak oluşturulmuş n-gram modeli ile yazım hatası denetimi yapılması kısımlarından oluşmaktadır. Çalışma hem İngilizce için hem de Malta dili için denenmiş ve iyi sonuçlar elde edilmiştir.

Yazım Denetleyiciler yöntemlerine göre de farklılık göstermektedirler. Örneğin İngilizce gibi yalın dillerde değişik sözcük karşılaştırma yöntemleri sıklıkla kullanılmıştır. En sık kullanılan yöntem en kısa uzaklık (minimum edit distance) yöntemidir. Bu yöntem, bir sözcükten diğer bir sözcüğü elde etmek için gereken harf değişiklikleri sayısı şeklinde tanımlanmıştır. Bu çalışmalarda denetlenen sözcüğün

* n-gram: olasılık değerlerine göre bir sonraki grubu tahmin etme esasına dayanan istatistiksel model

* Viterbi algoritması: Saklı katmanlar içinde en olası sıralamayı bulma algoritmasıdır.

tamamı karşılaştırılmaktadır. Yani yalın diller için uygun bir yöntemdir. En kısa uzaklık yöntemi istatistiksel tabanlı diğer yöntemlerle güçlendirilerek de kullanılmaktadır. Bu çalışmalara örnek olarak aşağıdakiler sıralanabilir:

- “Finding Approximate Matches in Large Lexicons – Justin Zobel, Philip Dart” (Büyük sözlüklerde yaklaşık eşleştirmeleri bulmak) [12]: Bu çalışmada farklı yöntemler birleştirilerek büyük sözcük kümeleri üzerinde sözcük karşılaştırma işlemi test edilmiştir. Kullanılan yöntemler, en kısa uzaklık yöntemi, n-gram yöntemi ve ses kodlama (Phonetic Coding) yöntemidir. Ses kodlama yöntemi söylenişi birbirine benzeyen sözcükleri sınıflandırmak için kullanılmıştır. Her harf bir koda sahiptir. Bir sözcüğün kodunu bulmak için ilk harf hariç tüm harflerin kod karşılığı yazılır. Tekrarlı kodlar teke indirilir ve 0’lar silinir. 0 kodlu harfler, a, e, h, y gibi söylemi sesli olan harflerdir. Son olarak elde edilen kodun ilk 4 karakteri alınır ve sözcüğün ses kodlaması böylece elde edilmiş olur.

Yazım denetleyicilerin büyük bir çoğunluğunun İngilizce ve benzeri yalın diller için yapıldığı görülmüştür. Bu çalışmalar genelde sözlük tabanlı basit uygulamalardır. Çok daha karmaşık bir yapıya sahip olan bağlantılı diller için ise daha zor ve özel çalışmalar yapılmıştır. Bu çalışmalara bir örnek olarak (Türkçe için yapılmış çalışmalar hariç) aşağıdaki çalışma gösterilebilir:

- “Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset - Jan Hajic, Barbora Hladka” (Bağlantılı Dillerde Etiketleme) [13]: Çek dili için yapılmış bir sözcük etiketleme çalışmasıdır.

Türkçe için yapılmış çalışmalar açısından bakıldığında, genellikle kural tabanlı çalışmalar yapılmıştır. Kök ve ek sözlükleri kullanılmıştır. Kural tanımları Sonlu Durum Makinaları ile kullanılmıştır. Bu tür çalışmaları aşağıdaki şekilde sıralayabiliriz:

- ”Design And Implementation of a Spelling Checker for Turkish - Aysin Solak, Kemal Oflazer - 1993” (Türkçe için yazım denetleyici tasarımı ve gerçekleştirimi) [14]: Kural Tabanlı bir yaklaşım ile yaklaşık 25,000 sözcükten oluşan bir kök sözlüğü ve Türkçe’deki ekler kullanılarak tasarlanmıştır. Biçimbirimsel çözümleyici ile sözcükler çözümlenmiş, sonlu durum makinaları üzerinden öncelikle Türkçe’nin ses uyumu kuralları, daha

sonra Türkçe'nin biçimbirimsel kuralları (kök ve eklerin sıralanışı ile ilgili kurallar) denetlenerek, sözcüğün hatalı olup olmadığı incelenmiştir. Kural dışı kullanımlar için de çalışmalar yapılmıştır. Sadece kelime düzeyinde hata bulmaya yönelik bir çalışma olup, anlamsal hataları ve hata düzeltme işlemini ele almamaktadır.

- “Error-Tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction - Kemal Oflazer, 1996” (Biçimbirimsel analiz ve yazım denetimi uygulamalarında hata toleranslı sonlu durum tanınma) [15]: Sonlu durum makinaları ile kelimeleri tanımaya dayalı bir çalışmadır. Fakat önemli olan nokta doğru kelimeler yanında belli bir yakınlık seviyesine kadar olan hatalı kelimeleri de tanıyabilme özelliğidir. Yakınlık seviyesi, X kelimesini Y kelimesine dönüştürmek için yapılması gereken harf silme, ekleme ve yer değiştirme sayısı olarak tanımlanmaktadır.
- “Spelling Correction in Agglutinative Languages - Kemal Oflazer, Cemaleddin Güzey- 1994” (Bağılantılı dillerde yazım düzeltme) [16]: Kemal Oflazer'in “Error-Tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction” [15] çalışması temelinde yapılmıştır. Bir kök sözlüğü üzerinden hatalı sözcüğe en yakın kökler elde edilmekte (2-gramlar kullanılarak) ve sonlu durum makinaları ile bu köklerden türetilebilecek tüm sözcükler türetilmektedir. Sözcüklerin türetilmesi için belli bir yakınlık seviyesi gözetilmektedir. Türetilmiş tüm sözcükler test edilip, en uygunu seçilmektedir. Değişik bir yaklaşım açısı ile Türkçe'ye ait kurallı ek dizilişi sayesinde yapılmış bir çalışma olması sebebiyle önemlidir.

Ayrıca farklı dillerde yapılmış çalışmalarla da karşılaşmıştır. Bunlara birkaç örnek aşağıda verilmiştir:

- “Tamil Spell Checker –Thanabalan, Ranjani Parthasarathi, T.V. Geetha” (Tamil Yazım Denetleyici) [17]: Hindistan'da konuşulan dillerden biri olan Tamil dili için yapılmış bir çalışmadır. Tamil dili eklemeli bir dildir. Çalışmada denetimi yapılacak sözcükler, hatalı sözcükleri çözümleyemeyecek şekilde hazırlanmış bir biçimbirimsel çözümleyiciden geçirilir. Hatalı sözcükler sözlükteki en yakın sözcükler ile düzeltilmek üzere

önerilir. Hata düzeltme sırasında eylem, isim vb. sözcük türlerine göre özel kurallar aracılığıyla öneri türetimi yapılmaktadır. Ayrıca ses benzerliği olan ve klavyede yakın yer alan karakterler için de özel yaklaşımlar uygulanmaktadır.

- “Improving a Spelling Checker for Afrikaans – Menno van Zaanen, Gerhard van Huyssteen” (Bir Afrikaans dili Yazım Denetleyicisinin İyileştirilmesi) [19]: Güney Afrika’da konuşulan dillerden biri olan Afrikaans dili (Afrika dili değil) için yapılmış bir çalışmadır. Afrikaans eklemeli bir dildir. Biçimbirimsel çözümleyici tabanlı yapılmış bir yazım denetimi uygulamasını geliştirmek üzere yapılmıştır. Yapılan en önemli iki geliştirme, hata sözlüğü ve n-gram kullanımıdır. Hatalı sözcükleri içeren bir kümede yapılan arama ile sıkça rastlanan bir hataya anında ulaşılabilir. Bunun dışında, kelimenin hatalı olup olmadığı yapılan n-gram analizi ile de denetlenmektedir.

Tüm bunlar dışında farklı yaklaşımlarla gerçekleştirilmiş çalışmalar da bulunmaktadır. Bunlara örnek olarak aşağıdaki çalışmalar verilebilir:

- “Reversed word dictionary and phonetically similar word grouping based spell-checker to Bangla text – Bidyut Baran Chaudhuri” (Bangladeş metinleri için ters sözlük ve sessel benzer sözcük gruplamasına dayalı yazım denetleyici) [19]: Hata bulma ve düzeltmede kullanılan ana mantık düz ve ters (son harf başa gelecek şekilde harflerin sıralanışı ters çevirilmiştir) sözlüklere dayanmaktadır. Hatalı sözcüğün soldan hatalı harfe kadar olan kısmı düz sözlükten, sağdan hatalı harfe kadar olan kısmı ters sözlükten elde edilir. Bu iki kısım arasında kalan kesişim kümesinin, hatalı sözcüğün doğru halini içerdiği varsayılmaktadır. Kalan kısım içinde en kısa uzaklığa sahip olan sözlük önerme olarak seçilir. Bu çalışmada, bahsedilen mantık bağlantılı bir dil olan Bangladeş dili için kullanılmıştır. Sözcükler kök ve eklerine göre iki kısımda ele alınmaktadır. Eğer hatalı sözcüğün baş tarafından bir kısmını içeren bir kök sözcük bulunmaktaysa, kalan kısım için ek sözlüğüne bakılmaktadır. Ek sözlüğünde de kalan kısım (ya da uygun en kısa uzaklığa sahip bir sözcük) bulunamaz ise hata kökte aranır. Kelimenin tamamı düz sözlükte bulunamaz ise kökte hatayı düzeltme mantığı yukarıda bahsedildiği şekilde çalışmaktadır. Kökte hata için uygun önerme bulunamaz ise ekler için

aynı mantık kullanılarak sonuca varılmaya çalışılmaktadır. Tek harf hatalarında kesin sonuç veren çalışma, ters sözlük kullanma ve kelimenin düz ve ters sözlükte bulunamayan kısmı üzerinde çalışması bakımından değişik bir bakış açısına sahiptir.

- “Automatic Rule Acquisition for Spelling Correction - Lidia Mangu, Eric Brill” (Yazım düzeltimi için otomatik kural çıkarımı) [20]: Anlamsal yazım hataları ile ilgilenen bir çalışmadır. Anlamsal olarak karıştırılan sözcükler için bir kaynak listesi oluşturulur. Bu liste yanında bir de kural tanım listesi yer almaktadır. Kurallar, hatalı sözcüğün etrafındaki sözcüklere göre tanımlanmıştır. Dilbilgisi hataları ele alınmadığı için bu tez kapsamında ilgilenilen konudan farklı bir taraftadır.

Yazım Denetimi ile doğrudan ilgili olmasa da bu tez çalışması sırasında faydalanılmış ve incelenmiş, Türkçe’nin biçimbirimsel yapısı ile ilgili çalışmalar da aşağıda belirtilmektedir:

- “Two-level Description of Turkish Morphology – Kemal Oflazer” (Türkçe’nin iki seviyeli biçimbirimsel yapısı) [21]: Türkçe’nin sözcük yapısını modelleyen bir çalışmadır. PC-KIMMO ortamında 23,000 kök sözcük içeren bir sözcük kümesi kullanılarak, sonlu durum makineleri ile gerçekleştirilmiştir. Sonlu durum makineleri üzerinde 22 tane kural tanımı yapılmış, bu kurallar ile bağlantılı dil yapısı modellenmeye çalışılmıştır. Kural dışı kullanımların tamamına yakını ele alınmıştır. Bu çalışma temel alınarak yapılmış birçok çalışma bulunmaktadır.
- “An Affix Stripping Morphological Analyzer for Turkish – Gülşen Eryiğit, Eşref Adalı” (Sözlüksüz Köke Ulaşma Yöntemi) [22]: Sözlük kullanmadan biçimbirimsel çözümleme yönteminin gerçekleştirilmesi çalışmasıdır. Türkçe’nin biçimbirimsel ve ses uyumu kuralları sonlu durum makinaları ile tanımlanmış, sözcükler sondan başlanarak eklerine ayrılmış ve köke ulaşılmıştır. Sözlük kullanılmaması açısından da önemli bir çalışmadır.
- “Statistical Morphological Disambiguation for Agglutinative Languages – Dilek Z. Hakkani Tür, Kemal Oflazer, Gökhan Tür” (Bağlantılı diller için istatistiksel biçimbirimsel çökyapılılık çözümlemesi) [23]

- “Unsupervised Learning in Constraint-based Morphological Disambiguation - Kemal Oflazer, Gökhan Tür-1996” (Kısıt tabanlı biçimbirimsel çökyapılılık çözümlemesinde denetimsiz öğrenme) [24]: Kural tabanlı çalışmalarda kullanılan Sonlu Durum Makinaları yanında öğrenmenin de ele alındığı bir çalışmadır.
- “A statistical information extraction system for Turkish – Gökhan Tür, Dilek Hakkani Tür, Kemal Oflazer” (Türkçe için istatistiksel bilgi çıkarım sistemi) [25]
- “Tagging and Morphological Disambiguation of Turkish Text – Kemal Oflazer, İlker Kuruöz” (Türkçe metinlerin işaretlenmesi ve biçimbirimsel çökyapılılık çözümlemesi) [26]
- “Morphological Disambiguation by Voting Constraints – Kemal Oflazer, Gökhan Tür” (Oylama yöntemi ile biçimbirimsel çökyapılılık çözümlemesi) [27]
- “On Lexicon Creation for Turkish LVCSR (Large Vocabulary Continious Speech Recognition) – Kadri Hacıoğlu, Bryan Pellom, Tolga Çiloğlu, Özlem Öztürk, Mikko Kurimo, Mathias Creutz” (Türkçe LVCSR için sözlük yaratımı) [8]: Konuşma Tanıma üzerine yapılmış, Kemal Oflazer’in Turklex zöcük kümesi ile PC-KIMMO ortamında geliştirilmiş bir çalışmadır.

Tüm bu çalışmalar Türkçe açısından düşünülürse, Türkçe için yapılmış çalışmalar genellikle sözlük kullanan ve kural tabanlı yaklaşımlardır. İngilizce ve benzeri yalın diller için yapılmış çalışmalar ise Türkçe’nin bağlantılı yapısını karşılayamadığı için, Türkçe açısından yetersiz kalmaktadırlar.

Bu tezde yapılmış olan çalışma, tüm bu incelenen yöntemlerden farklıdır ve şimdiye kadar kullanılmış birkaç yöntemi aynı çalışma içerisinde barındırmaktadır. Türkçe’nin yapısı düşünülerek yola çıkılmış olup, sözlükten bağımsız ve kurallar yerine istatistiksel temellere dayanan bir çalışmadır (sadece birkaç temel kural, çözümlenmiş metinlerin bütünleşik biçimlerini elde etmede, kök ve ek önermelerini birleştirerek yorumlama aşamasında kullanılmıştır). Türkçe yazım denetleme işlemini kök ve ekler açısından farklı şekilde yorumladığı, öğrenme mantığını da kullanarak ortam ve kullanıcı davranış biçimine duyarlı olması açısından da değişik bir bakış açısına sahiptir.

6. BÖLÜM

SONUÇ VE ÖNERİLER

Bu tez kapsamında Türkçe'nin bağlantılı bir dil olmasından yola çıkılarak, kök ve ekleri temel alan bir yazım denetleyici geliştirilmesi hedeflenmiştir. Kök sonuna getirilebilecek değişik sıra ve sayıdaki ekler, Türkçe'de sözlük kullanımını zorlaştırmaktadır. Yönetilemeyecek düzeyde büyük bir sözlük ile çalışmak yerine, sistemin belli bir konudaki (ya da daha genel) eğitim kümesi ile eğitilmesi ile oluşturulmuş kaynak kümeleri üzerinden denetim yapması üzerinde çalışılmıştır. Bu amaçla eğitim kümeleri seçilen bir biçimbirimsel çözümleyici ile kök ve eklerine ayrılarak işlenmiş ve kök ve ek olmak üzere, başvuru amaçlı iki kaynak küme oluşturulmuştur. Türkçe'de eklerin sıralanışının sahip olduğu önem sebebiyle, ekler bağımsız değil, ek çiftleri şeklinde ele alınmıştır.

Türkçe kurallı bir dildir. Fakat günlük kullanımda yer alan birçok yabancı ve kural dışı sözcük, kurallar doğrultusunda yapılan çalışmaları olumsuz yönde etkilemektedir. Bu sebeple sadece belli başlı birkaç ses uyumu kuralını gözetip, en temelde istatistiksel tabanlı bir çalışma ortaya çıkarılmıştır.

Denetleme işlemi hatayı bulma ve düzeltme şeklindedir. Bir sözcüğün hatalı olup olmadığını bulmak için öncelikle sözcüğü seçilen bir biçimbirimsel çözümleyici ile kök ve eklerine ayırmak, elde edilen kök ve ekleri kendi kaynak kümelerinde taramak gerekmektedir. Düzeltme işlemi için de yine bu kaynak kümelerden hatalı kök ya da ek için en uygun kaynak küme elemanları aranmıştır. Bu amaçla kaynak kümede sözcükleri ile hatalı sözcük arasında derecelendirme esaslı bir karşılaştırma yöntemi geliştirilmiştir. Bu derecelendirme kavramı 3 farklı mantığın birleşiminden oluşmaktadır. İlk kısım, bulanık mantık ile elde edilen sözcüklerin benzerlik oranıdır. İkinci kısım, sistemde geçmişte yapılmış ve öğrenilmiş hatalar üzerinden elde edilmiş öğrenme puanıdır. Son kısım da sözcüklerin kaynak eğitim sırasında elde edilmiş karşılaştırma oranlarıdır. Kök ve ek çiftleri için ayrı ayrı önermeler bulunmuş, bu önermelerin tek bir sözcük (çözümlemenin tersi) oluşturulması sırasında belli Türkçe ses uyumu kuralları gözetilerek sonuç önermeler elde edilmiştir. Bu temel

kuralların gözetilmesinin bir önemli sebebi de sözcüklerin çözümlemesi sırasında biçimbirimsel çözümleyici tarafından bu kuralların uygulanmış olması ve çözümleme işleminin tersi yapılırken de doğru sonuca ulaşmak için bu kuralların tekrar uygulanması gerekliliğidir. Çünkü bazı durumlarda çözümleme sonucunda sözcüğün içeriği değişebilmektedir (ses düşmesi, ses yumuşaması vb.).

Her denetim işleminin ardından yapılmış hatalar sistem tarafından öğrenilir. Öğrenme yapısı Yapay Sinir Ağları kullanılarak gerçekleştirilmiştir. Öğrenilmiş bilgiler, bir sonraki denetimde elde edilecek öğrenme puanında etkisini göstermektedir.

Sistem farklı biçimbirimsel çözümleyiciler ile elde edilmiş eğitim ve denetim metinleri ile test edilmiş çözümleyicilere göre Bölüm 3.5.7’de belirtilen sonuçlar elde edilmiştir. Uygulamanın başarımı biçimbirimsel çözümleyicinin başarımından doğrudan etkilenmektedir. Çünkü biçimbirimsel çözümleyicinin yanlış ürettiği bir sonuç, sistemi yanıltabilmektedir. Farklı biçimbirimsel çözümleme uygulamaları ile sistem başarımı daha da geliştirilebilir.

KAYNAKLAR

- [1] **Manning, C. D. and Schütze, H.,** 1999. Foundations Of Statistical Natural Language Processing, The MIT Press.
- [2] **Jurafsky, D. and Martin J. H.,** 2000. Speech and Language Processing, Prentice-Hall,Inc.
- [3] **Cucerzan S. and Brill E.,** 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users, Microsoft Research, Proceedings of EMNLP 2004, Barcelona.
- [4] **Schiller, N., Greenhall, J. A., Shelton, J. R., Caramazza A.,** 2001. Serial Order Effects in Spelling Errors: Evidence from Two Dysgraphic Patients, *Neurocase*, 7, 1±14.
- [5] **Hengirmen, M.,** 2002. Türkçe Temel Dilbilgisi.
- [6] **Taghva, K., Borsack J. and Condit, A.,** 1994. An expert system for automatically correcting OCR output, iST/SPIE 1994 International Symposium on Electronic Imaging Science and Technology (San Jose, Calif.). 270-278.
- [7] **Martins, B. And Silva, M. J.,** 2004. Spelling Correction for Search Engine Queries,EsTAL - España for Natural Language Processing, Alicante, Spain.
- [8] **Hacıoğlu, K., Pellom, B., Çiloğlu, T., Öztürk, Ö., Kurimo, M., Creutz, M.,** 2003. On Lexicon Creation for Turkish LVCSR (Large Vocabulary Continious Speech Recognition) , INTERSPEECH-2003/Eurospeech-2003, pp. 1165-1168, Geneva, Switzerland.
- [9] **Kernighan, M. D., Church, K. W., Gale W. A.,** 1990. A Spelling Correction Program Based on a Noisy Channel Model, In COLING-90 II: 205--211. Helsinki.
- [10] **Mizzi, R.,** 2000. The Development of a Statistical Spell Checker for Maltese, *B.Sc. Thesis*, Department of Computer Science & AI University of Malta.
- [11] **Tong, X. and Evans, D. A.,** 1996. A Statistical Approach to Automatic OCR Error Correction in Context, Proceedings of the Fourth Workshop on Very Large Corpora (WVLC-4), Copenhagen, Denmark, 88--100.
- [12] **Zobel, J. and Dart, P.,** 1995. Finding Approximate Mathes in Large Lexicons , *Software Practice and Experience*, Vol 25(3).
- [13] **H., Jan and H., Barbora,** 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset, COLING-ACL'98. Montréal.

- [14] **Solak, A. and Oflazer, K.**, 1993. Design And Implementation of a Spelling Checker for Turkish, *M. Sc. Thesis*, Bilkent University, 10/89 -- 5/91.
- [15] **Oflazer, K.**, 1996. Error-Tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction, *Computational Linguistics* Vol.22 No:1.
- [16] **Oflazer, K. and Güzey, C.**, 1994. Spelling Correction in Agglutinative Languages, *Proceedings of the 4th ACL Conference on Applied Natural Language Processing*, Stuttgart, Germany.
- [17] **Thanabalan, Ranjani Parthasarathi, T.V. Geetha**, 2003. Tamil Spell Checker, *Tamil Internet Conference - TI2003*.
- [18] **Zaanen, M. And Huyssteen, G.**, 2002. Improving a Spelling Checker for Afrikaans, *Computational Linguistics in the Netherlands. Thirteenth CLIN Meeting*. Rodopi. Amsterdam/New York. 143-156.
- [19] **Chaudhuri, B. B.**, 2000. Reversed word dictionary and phonetically similar word grouping based spell-checker to Bangla text , *Indian statistical Institute, Kolkata, India*.
- [20] **Mangu, L. and B. Eric**, 1997. Automatic Rule Acquisition for Spelling Correction, *Proceedings of the 14th International Conference on Machine Learning (ICML-97)*. Nashville, TN.
- [21] **Oflazer, K.**, 1994. Two-level Description of Turkish Morphology, *Literary and Linguistic Computing*, Vol. 9, No:2, 1994.
- [22] **Eryiğit, G.**, 2002. Sözlüksüz Köke Ulaşma Yöntemi, *Yüksek Lisans Tezi*, İ.T.Ü. Bilgisayar Mühendisliği Bölümü, İstanbul.
- [23] **Tür, D.Z., Oflazer, K., Tür, G.**, 2002. Statistical Morphological Disambiguation for Agglutinative Languages , *Computers and the Humanities*, Vol. 36 No.4, 2002.
- [24] **Oflazer, K., Tür, G.**, 1996. Unsupervised Learning in Constraint-based Morphological Disambiguation, *Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, USA
- [25] **Tür, G., Tür, D.Z., Oflazer, K.**, 2003. A statistical information extraction system for Turkish, *Natural Language Engineering*, Vol.9 No: 2
- [26] **Oflazer, K. And Kuruöz, İ.**, 1994. Tagging and Morphological Disambiguation of Turkish Text, *M.Sc. Thesis*, Bilkent University, 10/92 -- 7/94.
- [27] **Oflazer, K. and Tür G.**, 1997. Morphological Disambiguation by Voting Constraints, *Proceedings of ACL'97/EACL'97, The 35th Annual Meeting of the Association for Computational Linguistics*, Madrid Spain
- [28] **Yarowsky, D.**, 1994. Decision Lists For Lexical Ambiguity Resolution: Application To Accent Restoration In Spanish And French, *proceedings of ACL 32nd Annual Meeting*.
- [29] **Golding, A. and Roth, D.**, 1999. A Winnow-Based Approach to Context-Sensitive Spelling Correction,” 1999. *Machine Learning* 34. 107.

- [30] **Golding, A.**, 1997. A Bayesian hybrid method for context-sensitive spelling correction, Proceedings of the Third Workshop on Very Large Corpora. ACL. Boston, MA. 39--53
- [31] **Zemberek Yazılımı**, 2005. Uygulama kullanımı.
- [32] **Morfolojik Analiz Uygulaması**, 2002. Uygulama kullanımı.
- [33] **Akın, A.D. ve Akın, M.**, 2005. Zemberek yazılımı hakkında bilgi edinimi.
- [34] **Eryiğit, G.**, 2005. Morfolojik Analiz yazılımı hakkında bilgi edinimi.
- [35] **Oflazer, K.**, 2005. Biçimbirimsel Çözümleyici sonuçları edinimi.

EK-A

KLAVYE KOORDİNAT DEĞERLERİ

Tablo A.1: Q Klavye Harf Koordinatları

Harf	X	Y
A	-6	0
B	-1,5	-1
C	-3,5	-1
Ç	2,5	-1
D	-4	0
E	-4,5	1
F	-3	0
G	-2	0
Ğ	3,5	1
H	-1	0
I	0,5	1
İ	4	0
J	0	0
K	1	0
L	2	0
M	0,5	-1
N	-0,5	-1
O	1,5	1
Ö	1,5	-1
P	2,5	1
Q	-6,5	1
R	-3,5	1
S	-5	0
Ş	3	0
T	-2,5	1
U	-0,5	1
Ü	4,5	1
V	-2,5	-1
W	-5,5	1
X	-4,5	-1
Y	-1,5	1
Z	-5,5	-1

Tablo A.2: F Klavye Harf Koordinatları

Harf	X	Y
A	-3	0
B	-1,5	-1
C	-2,5	-1
Ç	-1,5	-1
D	-1,5	1
E	-4	0
F	-6,5	1
G	-5,5	1
Ğ	-4,5	1
H	1,5	1
I	-3,5	1
İ	5	0
J	5,5	-1
K	0	0
L	2	0
M	1	0
N	0,5	1
O	-2,5	1
Ö	-4,5	-1
P	2,5	1
Q	3,5	1
R	-0,5	1
S	0,5	-1
Ş	4	0
T	-1	0
U	-6	0
Ü	-2	0
V	-3,5	-1
W	4,5	1
X	-2,5	-1
Y	3	0
Z	-0,5	-1

ÖZGEÇMİŞ

1978 yılında Ankara’da doğan Simla DİLSİZ, 1996 yılında Alanya Lisesi’nden mezun olmuştur. Lisans öğrenimini Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümünde, 2000 yılında tamamlamıştır. Simla DİLSİZ, 2000 yılından beri Garanti Teknoloji’de çalışmaktadır.