İSTANBUL KEMERBURGAZ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KLASİK SORU DEĞERLENDİRME KARAR DESTEK SİSTEMİ

YÜKSEK LİSANS TEZİ

FUAT CANDAN

İSTANBUL, 2015

KLASİK SORU DEĞERLENDIRME KARAR DESTEK SİSTEMİ

Fuat CANDAN

Lisans Derecesi, Sakarya Üniversitesi, 2003

İstanbul Kemerburgaz Üniversitesi, 2015

İstanbul Kemerburgaz Üniversitesi

Yüksek Lisans, Bilişim Teknolojileri'ne

sunulmuştur.

Bu çalışma tarafımızca incelenmiş olup, kapsam ve kalite açısından Yüksek Lisans tezi olmaya yeterli bulunmuştur.

Yrd. Doç. Dr. Tuğçe Ballı ALTUĞLU
Danışman

İnceleme Komitesi Üyeleri (İlk isim jüri başkanına, ikinci isim tez danışmanına aittir.)

Yrd. Doç. Dr. Kerem ALTUN (Jüri)

Yrd. Doç. Dr. Tuğçe Ballı ALTUĞLU (Jüri)

Dr. Mert ÖZKAYA (Jüri)

Bu çalışma bir Yüksek Lisans tezinin tüm gerekli şartlarını taşımaktadır.

Yrd. Doç. Dr. Tuğçe Ballı ALTUĞLU
ABD Başkanı

[Üniversite] onayı 26 / 06 / 2015

Doç. Dr. Oğuz BAYAT
Enstitü Müdürü

Bu dokümandaki tüm bilgilerin akademik kural ve etiğe bağlı kalınarak yazıldığını ve tez yazım kuralları kapsamında bu çalışmada bulunan ve orijinal olmayan bütün bilgi ve materyallerin referanslandırıldığını temin ederim.

Fuat CANDAN



İTHAF

Tez danışmanım Yrd. Doç. Dr. Tuğçe Ballı ALTUĞLU'ya, bu çalışmanın gerçekleştirilmesindeki katkı ve yardımlarından dolayı teşekkürlerimi sunarım.

Tez çalışmam sırasında her türlü desteğini esirgemeyen değerli öğretmen arkadaşım Mustafa Coşkun ve Bahçelievler Ticaret Meslek Lisesi Bilişim Teknolojileri Bölümündeki öğretmen arkadaşlarıma teşekkürlerimi sunarım.

Bu yüksek lisans tezimi, benim bu günlere gelmem için hiçbir fedakârlıktan kaçınmayan ve her zaman yanımda olan sevgili aileme, sabırla bana destek olan canım eşime ve en önemlisi de kendilerine örnek olabilmek için çocuklarıma ithaf ediyorum.

ÖZET

KLASİK SORU DEĞERLENDİRME KARAR DESTEK SİSTEMİ

Fuat Candan

Yüksek lisans, Bilişim Teknolojileri, İstanbul Kemerburgaz Üniversitesi,

Danışman: Yrd. Doç. Dr. Tuğçe Ballı ALTUĞLU

Bu tez çalışmasında, öğreticilerin klasik soru değerlendirme süreçlerinde kullanmaları için not önerisinde bulunacak bir karar destek sistemi geliştirilmiştir. Bu çalışmanın amacı öğreticilerin kendi not değerlendirmeleri ile sistemin not değerlendirmeleri arasında karşılaştırma yaparak adil bir değerlendirme süreci ortaya çıkarmalarına yardımcı olmaktır. Bu çalışmada klasik soru cevaplarını oluşturan cümlelerin kelimelere ve bu kelimelerin köklerine ayrılması için, aynı zamanda kelime yazımlarının denetlenmesi ve olası düzeltmeler için Zemberek DDİ Kütüphanesi kullanılmıştır. Her bir klasik sorunun doğru cevabı ve öğrenci tarafından verilen cevabın benzerliğini hesaplamak ve buna göre önerilecek notun hesaplamasını yapmak için ise Vektör Uzay modeli ve Kosinüs Benzerliği Teoremi kullanılmıştır. Geliştirilen algoritma sonuçlarının bilimsel geçerliliği, öğreticinin not değerlendirmesi ve sistemin not değerlendirmesi arasındaki ilgileşim (korelasyon) hesaplanarak ispatlanmıştır. Korelasyon hesaplamaları için Pearson's Korelasyon katsayısı (Pearson's Correlation) ve T-test yöntemleri kullanılmıştır. Yapılan analizler sonucunda Pearson's Korelasyon katsayısı 0,667 olarak bulunmuştur. T-test analizi sonucunda da Pearson's Korelasyon katsayısı hesaplamaları ile birebir benzer sonuçlar elde edilmiştir. Bu sonuçlar sistemin not değerlendirmeleri ve öğreticinin not değerlendirmeleri arasında güçlü bir korelasyon olduğunu göstermiştir.

Anahtar Kelimeler: Klasik Soru Değerlendirme, Kosinüs Benzerlik Teoremi, Vektör Uzay Modeli, Zemberek DDİ Kütüphanesi

ABSTRACT

A DECISION SUPPORT SYSTEM FOR EVALUATION OF ESSAY TYPE QUESTIONS

Fuat Candan

Master of Science, Information Technologies, Istanbul Kemerburgaz University,

Advisor: Asst. Prof. Dr. Tuğçe Ballı ALTUĞLU

In this thesis study, a decision support system was developed for instructors to aid them during evaluation of short answer/essay type questions. The aim of this study was to devş6elop a system that will help the instructor to make a fair grading by comparing his/her grade with the system's suggested grade. Zemberek NLP Library was used for separating given answer (by the student) and the correct answer sentences ş2to words and words to roots. This library was also used for spell checking and correction of the typos in the given answers. Vector space model and cosines similarity theory were used to calculate the similarity between correct answers and the given answers. The scientific significance of the results of developed algorithm was tested by calculating the Pearson's correlation and by applying t-test between actual grades (given by the instructor) and the grades calculated by the system. The Pearson's correlation coefficient between given grades and the system grades was found to be 0,667. T-test analysis results have also given very similar results to Pearson's correlation coefficient. These results indicated a moderately strong relationship between the given grades and the system grades.

Keywords: Essay Type Questions, Cosines Similarity Theory, Vector Space Modelling, Zemberek NLP Library

İÇİNDEKİLER

ŞEKİLLER TABLOSU 1
TABLO LİSTESİ 3
KOD LİSTESİ 4
FORMÜL LİSTESİ 5
KISALTMALAR 6
1. GİRİŞ 7
1.1 Tezin Amacı 7
1.2 Tezin Uygulanması 8
2. ALT YAPI ve LİTERATUR ÖZETİ 9
2.1. VERİ MADENCİLİĞİ (DATA MINING) 12
2.1.1 Sınıflandırma 13
2.1.1.1 K-NN 13
2.1.1.2 Karar Ağaçları 14
2.1.2 Kümeleme 14
2.1.3 Birliktelik Kuralı 16
2.2 METİN MADENCİLİĞİ (TEXT MINING) 17
2.1.1 Bilgiye Erişim 17
2.1.2 Bilgi Çıkarımı 18
2.1.3 Web Madenciliği 18
2.2. VEKTÖR UZAYI MODELİ 18
2.3 KOSİNÜS BENZERLİĞİ 20
2.4 DOĞAL DİL İŞLEME 22
2.4.1 Soru Cevap 23
2.4.2 Bilgi Çıkarımı 23
2.4.3 Duygu Analizi 23
2.4.4 Makine Çevirisi 23
2.4.5 Diğer Çalışmalar 24
2.5 ZEMBEREK DDİ KÜTÜPHANESİ 24
3 PROGRAM GELİŞTİRME 28
3.1 Verilerin Çekilmesi 28
3.2 Programlama Dili Seçimi ve Ara yüz Oluşturma 30
3.3 Kütüphaneleri Referans Verme 32
3.4 Zemberek DDİ Kütüphanesinin Projeye Eklenmesi 35

3.5 Verileri Ayrıştırma 36
3.6 Verilerin Sayısal Yapıya Dönüştürülmesi 40
3.6 Ağırlıklı Hesaplama 50
4 DEĞERLENDİRME 55
4.1 Veri Seti 55
4.2 Analiz Sonuçları 58
5. SONUÇ 64
5.1 Gelecekteki çalışmalar 65
5.2 Kısıtlar 65
REFERANSLAR 66

ŞEKİLLER TABLOSU

Şekil 1: Veri Madenciliği Süreci 12
Şekil 2: Karar Ağacı Yapısı 14
Şekil 3: K-Means Yöntemiyle Kümeleme 15
Şekil 4: Euclidean Distance, Manhattan Distance 15
Şekil 5: Vektör Uzayı Modelinde Bitsel Tanımlama 19
Şekil 6: Frekans Tanımlama 20
Şekil 7: Kosinüs Teoremi 21
Şekil 8: Zemberek "Trie Ağaç" Yapısı 26
Şekil 9: Zemberek DDİ Kütüphanesi ile Ayrıştırılmış Kelime 27
Şekil 10: Programın İşleyiş Adımları 28
Şekil 11: Veri Kaynağından, Verilerin Çekilmesi. 29
Şekil 12: Veritabanı Tablolar ve Alanlarının Şeması 29
Şekil 13: Programın Çalıştığı Ana Form 31
Şekil 14: MySQL Data Kütüphanesinin Eklenmesi 32
Şekil 15: Veritabanındaki Bilgilerin Alınması 33
Şekil 16: Cevapların Ayrıştırılması 36
Şekil 17: Cevap ve Öğrenci Cevap Örneği 36
Şekil 18: Kelime Dizisinin Köklerinin Bulunması. 37
Şekil 19: Zemberek DDİ Kütüphanesinin İşleyiş Şeması 39
Şekil 20: Örnek Cevabın Kelimelere Ayrılması, Yazım Önerisi, Kelime Türünün Bulunması 39
Şekil 21: Soru ve Cevap Ekleme Sayfası 41
Şekil 22: Soru Cevapların Tümünün Gösterimi 42
Şekil 23: Veritabanındaki Verilerin Boşaltılması 43
Şekil 24: MySQL Veritabanın Program Tarafından Başlatılması 43
Şekil 25: Soru Cevap ve Öğrenci Cevaplarının Ayrıştırılıp, Hesaplama Diyagramı 45
Şekil 26: Hesaplama Formu 46
Şekil 27: Soru Cevap ve Öğrenci Cevap Verileri Üzerinden Vektör Oluşturulması 47
Şekil 28: "d_{cevap}" ve "$d_{öğrencicevap}$" Dizilerinden q, d1, d2 Vektörlerinin Oluşturulması 48
Şekil 29: Toplu Değerlendirme Ekranı 51
Şekil 30: Toplu Değerlendirme İşleyiş Basamakları 52

Şekil 31: Toplu Değerlendirme Sonuçlarının Gösterilmesi 53
Şekil 32: Soru, Cevap, Soru Puan ve Öğrenci Cevap Kayıtlarının Görünümü 55
Şekil 33: Kosinüs Benzerlik Teoremi 56
Şekil 34: Sorunun Cevabı ile Öğrenci Cevabının Benzerlik Hesabıyla Bulunan Sonuçları 57
Şekil 35: Klasik Soru Değerlendirme Sistemi Algoritması 63

TABLO LİSTESİ

Tablo 1: Sonuçların Ortalaması ve Standart Sapma Değerleri 58

Tablo 2: Pearson Korelasyon Katsayısı Analizi 59

Tablo 3: Pearson Korelasyon KatsayısıLineer İlişki gücü Seviyeleri..... 59

Tablo 4: T-Test Sonuçlarının Gösterimi 60

Tablo 5: Tekrar Düzenlenmiş Veri Seti Üzerinde Analiz Sonuçları. 61

KOD LİSTESİ

Kod 1: Zemberek DDİ Kütüphanesini Referans Verme 26
Kod 2: Zemberek DDİ Kütüphanesinin Kullanımı 27
Kod 3: MDI Form'da Child Form Açma 31
Kod 4: DataReader Nesnesi ile Veritabanından Verilerin Alınması 34
Kod 5: DataSet Nesnesi ile Verilerin Çekilmesi 34
Kod 6: Cümlenin Kelime Dizisine Dönüştürülmesi 37
Kod 7: Kelimelerin Küçük Harfe Dönüştürülmesi. 38
Kod 8: Zemberek DDİ Kütüphanesi ile Kelime Çözümleme 38
Kod 9: DataTable tanımlama ve Veri ekleme 40
Kod 10: DataTable 'kok'e Göre LINQ ile Gruplandırıp, dataGridView Nesnesinde Göst 41
Kod 11: "öğrenci", "soru cevap" ve "öğrenci cevabının" Gösterilmesi 43
Kod 12: MySQL Veritabanın Program Tarafından Başlatılması 44
Kod 13: Kelime Köklerine Ayrılmış Kelimelerin LINQ ile Gruplandırılması. 47
Kod 14: "q" Doğrusunun Elemanı "d1" ve "d2" Vektörünü Oluşturma ve Değer Atama 49
Kod 15: d_{cevap} ve q Verilerinden d1 Doğrusunun Oluşturulması 49
Kod 16: Kosinüs Benzerliğinin Uygulanması 50
Kod 17: DataGridView Nesnesinden Excel'e Veri Aktarımı 54

FORMÜL LİSTESİ

Formül 1: Birliktelik Kuralı 16
Formül 2: Birliktelik Kuralının değerlerle ifade edilmesi 16
Formül 3: Doğruluk ve Duyarlılık Hesaplanması 18
Formül 4: Vektör Uzayı Modelinde Ağırlıkların Hesaplanması. 19
Formül 5: Kosinüs Benzerliği Formülü 20
Formül 6: Ağırlıklı Yöntemle Hesaplanması. 50

KISALTMALAR

ADDIE- Analiz Tasarım Geliştirme Uygulama Değerlendirme – Analysis Design Development Implementation Evaluation

AIS- Agrawal, Imielinski, Swami

API- Uygulama Programlama Ara yüzü - Application Programming Interface

DBMS – Veritabanı Yönetim Sistemi- Database Management System

DDİ- Doğal Dil İşleme – Natural Language Process

dll- Dinamik Link Library

GPL-Genel Kamu Lisansı - General Public License

KDS- Karar Destek Sistemi- Decision Support System

K-Means - K-Ortalama Algoritması

K-NN - En Yakın K Komşu K -Nearest Neighborhood

LINQ- Language Integrated Query

MDI - Multiple Document Interface

OOP-Nesne Tabanlı Programlama - Object Oriented Programming

PARM - P-tree based Association Rule Mining

SQL- Yapısal Sorgu Dili - Structured Query Language

T-SQL - Transact-SQL

TF-IDF – Terim Frekansı Ters Doküman Frekansı - Term Frequency Inverse Document Frequency

Trie- Re**trie**val Kelimesinin 4 Harfi

TÜBİTAK- Türkiye Bilimsel Ve Teknolojik Araştırma Kurumu

XML- Genişletilebilir İşaretleme Dili - Extensible Markup Language

1. GİRİŞ

Eğitim sisteminde teknoloji kullanımı bilişim teknolojilerinin hayatımıza dahil olması ile hızla yayılmaktadır. Eğitim teknolojilerinin en popüler başlığı olarak ise Eğitim Yönetim Sistemleri söylenebilir. Ülkemizde en yaygın kullanılan eğitim yönetim sistemi E-Okul'dur ve bu sistem 2004 yılından bu yana etkin olarak örgün eğitimin tüm seviyelerinde (ilk-orta-lise) kullanılmaktadır.

Teknolojinin eğitimde kullanılmasının bir diğer popüler örneği olarak ise uzaktan eğitim gösterilebilir. Daha çok üniversite düzeyinde kullanımı yaygınlaşan bu sistemlerin (uzaktan eğitim sistemleri) öncelikle özel eğitim ve yaygın eğitim kurumları olmak üzere lise düzeyinde de örneklerine rastlanmaktadır. Örneğin Milli Eğitim Bakanlığı motorlu taşıtlar sürücü kursu sınavlarını artık çevrimiçi (online) bir sistem aracılığıyla yapma çalışması içerisindedir. Bu eğitim sistemlerinin örneklerini (küçük çaplı kullanım oranına sahip yazılımlar, Bilsa, Okul Otomasyon vb.) arttırmak mümkündür.

Ölçme ve değerlendirme hiç şüphesiz eğitim-öğretim sisteminin en önemli kavramıdır. Sanıldığının aksine ölçme ve değerlendirme eğitim sürecinin planlanması aşamasından sonuçlandırılması aşamasına kadar her zaman üzerine düşünülmesi gereken bir konudur.

1.1 Tezin Amacı

Öğrencilerin girilen notlara bakarak başarı durumlarını değerlendirmede e-okul sistemi gerekli imkanları sunmaktadır. Ancak öğrencilerin bilgi düzeylerini içeren, klasik soruları değerlendirmede, herhangi bir çalışmanın olmadığını görmekteyiz. Bu eksiklikten yola çıkarak öğrencilerin klasik sorulara verdikleri cevapları değerlendirecek bir Karar Destek Sistemi (KDS, Decision Support System) tasarlamayı hedefledik.

Klasik Soru Değerlendirme Karar Destek Sistemi, soruların cevaplarına bakarak öğrencilerin verdiği cevapları puanlayacak ve öğretmenler için bir karar verme mekanizması oluşturacaktır. Değerlendirme konusu öğretmen öğrenci bazında göreceli bir süreç olsa da, bilişim yardımıyla bu göreceliğin hataya meyilli yapısı ve olası yanlışlıklar engellenebilir ve daha adil bir notlandırma sistemi sunulabilir.

Ölçme değerlendirme sürecinde öğrencilerin gelişimlerinin öğretmen tarafından sınanması ve bu sınama sonuçlarının -ne tamamen nesnel ne de tamamen öznel olarak- değerlendirilmesi gerekmektedir. Bu bağlamda son yıllarda çoktan seçmeli, doğru-yanlış vb. kesin cevaplı soruların sınavlarda kullanılması popülerliğini yitirmektedir. Eğitim bilimciler açık uçlu, yoruma dayalı, öğrencilerin bilgi birikimlerini sunabilen soruların (klasik soruların) en doğru tercih olduğu konusunda son yıllarda hem fikir olmaktadırlar.

Öğrencinin verdiği bir cevabı en iyi değerlendirecek olan insan zekasıdır, yani öğreticidir. Makinelere de böyle bir işlemi yaptırmak/yaptırabilmek çok da objektif olmayacaktır. Bizim buradaki amacımız değerlendirmede öğretmene öneri sunacak bir değerlendirme sistemi oluşturmaktır.

1.2 Tezin Uygulanması

Bu çalışmada eğitimsel tasarım ve eğitim sistemi geliştirme sürecinde en yaygın kullanılan modellerden biri olan ADDIE (Analysis-Design-Development-Implementation-Evaluation, Analiz-Tasarım-Geliştirme-Uygulama-Değerlendirme) modeli kullanılmıştır. Bu modelin oluşturduğu yapıya dayanarak, “Analiz” çalışması için, veri madenciliği ve metin madenciliği ile ilgili yapılmış çalışmalar araştırılmış ve bulunan sonuçlar Bölüm 2’de özetlenmiştir, ayrıca bu bölümde, alan ile ilgili bilgi verilmeye çalışılmış ve bu bilgiler ışığında genel tasarım planlanmıştır. Bölüm 3’de “geliştirme” aşaması için program arayüzünün yapım aşamaları ve kodlamalardan bahsedilmiştir, kullanılan yöntem ve teknikler anlatılmıştır. Bölüm 4’de sistemin “değerlendirme” süreci ve bu süreçte yapılan test ve analizler sunulmuştur. Ayrıca bu bölümde, “uygulama” için seçilen veri seti ve veriler açıklanmıştır. Son olarak Bölüm 5’de çalışma ile ilgili elde edilen sonuçlar değerlendirilmiş, değerlendirme sisteminin kısıtlılıklarından ve gelecekteki çalışmalara önerilerden bahsedilmiştir.

2. ALT YAPI ve LİTERATUR ÖZETİ

Karar Destek Sisteminin geliştirilmesi sürecinin ilk aşaması olan Analiz için geniş bir literatür taraması yapılmıştır. Yapılan bu araştırmada Metin ve Veri Madenciliğinin Eğitim Yönetim Sistemlerinde kullanıldığı örnekler incelenmiş ve çalışmaya katkısı olabilecek olan uygulamalar aşağıda özetlenmiştir.

Yıldız, "Metin Sınıflandırmada Yeni Özellik Çıkarımı", çalışmasında, ekonomi, sağlık, magazin, spor ve siyaset olmak üzere beş sınıftan seçtiği makaleleri sınıflandırmış, kelimeleri Zemberek DDİ kütüphanesi ile köklerine ayırmış, sık tekrarlanan terimleri ağırlıklarını hesaplamış, metin türlerine göre sınıflandırmaya çalışmıştır. Geliştirdiği algoritma ile metin sınıflandırmada ortala %90 başarı sağlamıştır[1].

İlhan, "Metin Madenciliği ile Soru Cevaplama Sistemi", çalışmasında kullanıcı tarafından girilen bir soruya en uygun cevabı içeren metin keşfedilmeye çalışılmıştır. Bu amaçla Zemberek DDİ (Doğal Dil İşleme) kütüphanesini kullanarak, cevaplar içerisindeki kelimelerin sözcük türünü ve kelime kökünü bulmuş, oluşturduğu bu sözcük listesini(keyword) Vektör Uzay Modelini kullanarak doğruya(sayısal yapıya) dönüştürmüştür. Değerlendirilecek cevap doğrularının arasındaki açılar, Kosinüs Benzerlik Teoremi ile hesaplamış ve bu hesaplamalar sonucunda anahtar sözcük listesi belirlenerek cevaplar için puanlama yapılmıştır. Çalışmanın sonunda doğru cevabı bulma hedeflemiştir[2].

Karaca, "Metin Sınıflandırmada Benzerlik Hesaplama Tekniklerinin Değerlendirilmesi" çalışmasında, internet sitelerinde bulunan benzer köşe yazılarını bulmayı amaçlamıştır. Bu amaçla metinleri vektör uzayına dönüştürmüş, 15 farklı metin sınıflandırma metodunu, günlük yayın yapan internet sitelerinin köşe yazılarından alınan, 500 eğitim ve 250 test dokümanına uygulamıştır. Bu çalışmada metin benzerliklerini ölçmek için kullanılan yöntemler arasında Kosinüs Benzerlik Yöntemi ve Pearson's Korelasyon katsayısı hesaplamalarının %100 başarıya ulaştığı görülmüştür[3].

Şeker, "Sosyal Ağlarda Akan Veri Madenciliği" çalışmasında, Facebook ve Twitter'dan topladığı verileri incelemiştir. Verilerin toplanması, verilerin temizlenmesi, veri özniteliklerinin çıkarılması ve oluşturulan veri kümelerinin (durum analizleri gibi) yeniden

kullanılması üzerinde durmuştur. Yazarların daha önceden yazdığı kelime karakteristiğine bakarak yeni yazıları ve bu yazıların yazarlarını tespit etmeye çalışmıştır[4].

Tantuğ, "Metin Sınıflandırma" çalışmasında, büyük boyutlara ulaşan verilerin işlenebilmesi için önceden belirlenen sınıflara ayrılması üzerinde durmuştur. Sözcük sepeti kullanarak anahtar sözcükleri toplamış, bu sözcük sepetini Vektör Uzay Modeli kullanarak hesaplanabilir hale getirmiştir. 20000 haber metni üzerinde K-NN ve NaiveBayes metotlarını kullanarak daha kısa sürede sınıflandırma yapılabileceğini, terim sayısı az metinler içinde "Destek Vektör Makineleri" ve K-Means yönteminin bir varyasyonu olan "Centroid" yöntemi kullanılmış ve yüksek başarım elde etmiştir[5].

Karadağ, "Metin Madenciliği ile Benzer Haber Tespiti" isimli çalışmasında, haber sitelerindeki benzer haberleri gruplandırarak okuyuculara daha çok seçenek ve kolaylık sunulacağını ispatlamaya çalışmıştır. Çalışmasında kelimeleri köklerine ayırmak için Zemberek DDİ kütüphanesini kullanmıştır. Etiket, metin ve içerik kısımlarını Terim Frekansı – Ters Doküman Frekansı (TF-IDF) metoduyla farklı ağırlıkta hesaplayarak benzer haberleri bulmada en düşük %33 ve en yüksek %95'e kadar başarıya ulaşmıştır[6].

Kaya, "Metin Madenciliği Kavramları ve Uygulama Alanları" çalışmasında, veri madenciliği ve bilgi keşfi üzerinde durmuş, Vektör Uzayı Modelini kullanarak verilerin sayısal bir yapıya dönüştürülmesini açıklamıştır. Daha sonra, bu verilerin üzerinde K-Means ve K-Medoids yöntemlerinin uygulamasını anlatmıştır. Anlatılan bilgiler ışığında, Vektör Uzayı Modeli'nin metin madenciliği için başarım oranı yüksek bir yöntem olacağı üzerinde durulmuştur [7].

Huang, "Similarity Measures for Text Document Clustering" çalışmasında metinleri Vektör Uzay Modelini kullanarak metrik (sayısal) bir yapıya dönüştürmüştür. Bu metrik verilere göre metinlerin benzerliğini Öklid Uzaklığı, Kosinüs Benzerliği, Jaccart, Pearson Korelasyonu ve Kullback-Leibler Açılması benzerlik ölçümleriyle hesaplamıştır. Benzer verileri kümelemede Kullback-Leibler Açılması ve Pearson Korelasyonu ölçümlerinin diğerlerine göre daha etkili olduğu sonucuna varmıştır[8].

Ergün, "İnternet Ortamındaki Tüketici Yorumlarından Özet Bilgi Çıkarımı" çalışmasında internet ortamında sunulan ürünlerle ilgili yapılan yorumlardan otomatik özet çıkarmaya çalışmıştır. Çalışmasında metin madenciliğinin yöntem ve tekniklerinden faydalanmıştır.

Çalışmasında kullanıcıların yaptığı yorumlardan çıkardığı kelimeleri morfolojik olarak ayrıştırmış ve bu kelimelere pozitif ve negatif anlamlı sıfatları bulmuştur. Kelimelerin kökünü bulmak için Zemberek DDİ kütüphanesinden faydalanmıştır. Çalışmasının sonucunda, ürünle ilgili yapılan yorumlara göre sayısal bir derecelendirme (yüzde) elde ederek yorumlamaya çalışmıştır[9].

Sonuç olarak, yapılan literatür taramasında, öğrencilerin klasik sorularının değerlendirilmesi ve karar destek sistemiyle ilgili herhangi bir çalışma bulunamamıştır ve özellikle orta eğitim kurumlarında öğreticilerin eksikliğini hissettiği klasik soru değerlendirme karar destek sistemi çalışmasının yapılmasına karar verilmiştir. Bu yüzden; yukarıda yapılmış olan literatür özetlerinde kullanılan yöntemlerde, elde edilen başarım oranlarından yola çıkarak, bu tez çerçevesinde yapılması amaçlanan çalışmada;

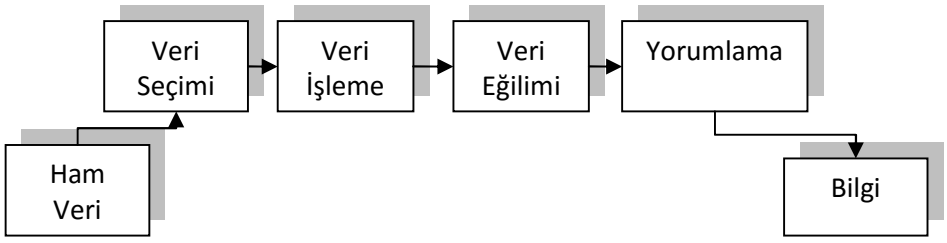
- analizi yapılacak olan öğrenci cevapları ve doğru cevap cümlelerinin kelimelere, kelimelerin köklerine ayrılması, aynı zamanda cevap cümlelerinin yazım hatalarının denetlenmesi ve olası düzeltmelerin yapılası aşamalarında Zemberek DDİ kütüphanesinin,
- kelimelere ve köklerine ayrılan doğru cevap ve öğrenci cevap cümlelerinin sayısal yapıya dönüştürme işleminde Vektör Uzay Modelinin,
- sayısal yapıya dönüştürülen doğru cevap ve öğrenci cevap cümlelerindeki köklerin benzerliklerini hesaplama ve öğrenci cevaplarının doğru cevaba benzerliğine göre sistemin değerlendirme notunu hesaplama sürecinde Kosinüs Benzerliği Teoreminin,
- öğreticinin not değerlendirmesi ve sistemin not değerlendirmesi arasındaki korelasyonun hesaplanmasında Pearson’s Korelasyon ve t-test analizlerinin,

kullanılmasına karar verilmiştir.

2.1. VERİ MADENCİLİĞİ (DATA MINING)

Veri Madenciliği, çeşitli ortamlardan toplanan verilerin anlamlı bilgiye dönüştürülmesidir[9]. Bu veriler metin dosyaları, ses ve görüntü olabilir. Anlamlı bilgiye dönüştürülen bu bilgiler her hangi bir satış stratejisinde, sağlık sektöründe veya sosyal medya analizinde toplumsal olayları anlamaya kadar birçok alanda kullanılmaktadır. Web üzerinde yapılan Web Madenciliği, tüketici alışkanlıklarını inceleyip, daha çok müşteriye ulaşarak pazarlama sektörünü tetiklemektedir. Günümüzde yatırımların hangi sektörde nereye yapılacağı ile ilgili bilgiler veri madenciliğiyle yapılan analizlerin bir sonucudur[9][10][11][12]. Binlerce veriden oluşan bu kayıtlar, doğru bakıldığında anlamlı bir bilgidir[13]. Şekil 1'de gösterildiği gibi kayıtlar (ham veriler) bu basamaklardan geçerek kullanılabilir bilgiye dönüşebilir.

Veri madenciliğinin yaygınlaşmasından sonra büyük veriler (big data) oluşmuş ve big data kavramı kullanılmaya başlanmıştır. Büyük veriler beraberinde depolama, senkronizasyon ve veri inceleme problemlerini de getirmiştir. Toplanan bu veriler sadece sayısal değerler olmayıp yarı yapısal bir yapıdadır. SQL dili bu tür verileri (yarı yapısal veriler) işlemekte yetersiz kalmış yeni teknolojiler geliştirilmeye başlanmıştır[9][14][15]. Verilerin büyüklüğü, birçok yerden ulaşımı kolaylaştırmak ve erişim sürekliliğinin sağlanabilmesi için veritabanından farklı yapılarda tutulur. NoSQL ve Dağıtık veri sistemleri diye de adlandırılabilen bu teknolojilerin en bilinen örnekleri Google'ın kullandığı "Map Reduce", Facebook'un kullandığı "Graph API" ve Amazon'un kullandığı "DynamoDB" dir.



Şekil 1: Veri Madenciliği Süreci

Şekil 1'de gösterildiği gibi, veri madenciliği süreci, veriyi seçme, veriyi işleme, veri eğilimini bulma ve bilgiye ulaşma (yorumlama) aşamaları şeklinde özetlenebilir[13]. Verinin evrimi

diyebileceğimiz Şekil 1'deki basamaklardan geçen veri, anlamlı bilgilere dönüştürüldükten sonra farklı uygulama alanlarında kullanılabilir[9][10][11][12].

Diğer bir tanım olarak, veri madenciliği ham verilerden değerli bilgiye ulaşma işlemidir[9][16]. Yorumlanan bu bilgiler karar destek sistemlerinde önemli bir yere sahiptir[9]. Şekil 1'de gösterildiği gibi ham veriler, seçildikten ve işlendikten sonra ortaya çıkan "veri eğilimi" bilginin yönelimini verir. Bunun sonucunda veriler uygulama alanına göre (yönetim, satış stratejisi vb) yorumlanabilir.

Veri madenciliğinde veri inceleme yöntemlerinden en klasik olanları istatistiksel analiz içeren yöntemlerdir[9]. Bu yöntemlerde regresyon analizi, standart dağılım, standart sapma ve güven aralıkları gibi metotlarla veriler arasındaki olası ilişkiler incelenir[9][10]. Bunlara ek olarak yapay zekanın, veri madenciliğinde kullanılmasıyla makinelere insan gibi karar verme yeteneği kazandırılmaya çalışılmıştır[9].

Veri madenciliğinde günümüzde sıkça başvurulan yöntemler aşağıdaki gibi özetlenebilir[9][16][17].

2.1.1 Sınıflandırma

Sınıflandırma (Clasification) en genel tanımıyla, verileri belirli gruplara veya etiketlere yerleştirme işlemi yapar. Bu yöntemle var olan veriler sınıflandırılır. Bu süreçte, eğiticili öğrenme (supervised learning) yöntemi kullanılır. Bu yöntemde toplam veri setinin bir kısmı eğitim seti olarak alınır ve geri kalan kısmı geliştirilen algoritmanın test edilmesi için kullanılır. Aşağıdaki alt bölümlerde metin sınıflandırmada sıklıkla kullanılan bazı sınıflandırma yöntemleri açıklanmıştır.

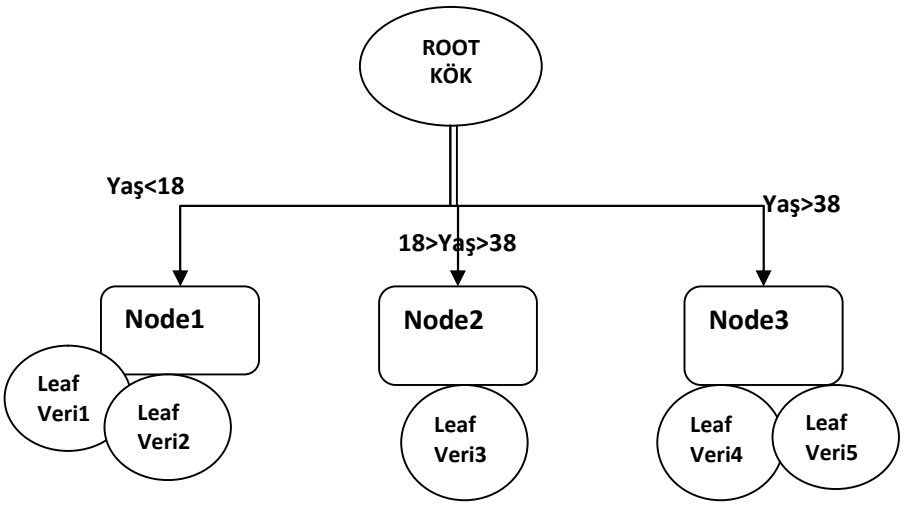
2.1.1.1 K-NN

Sınıflandırmada sıklıkla başvurulan yöntem K-NN (K-Nearest Neighbour-En Yakın Komşuluk) algoritması, verinin en yakın grup veya etikete göre hangi sınıfta yer alacağını belirler[9][16][17].

2.1.1.2 Karar Ağaçları

Sınıflandırmada en yaygın kullanılan yöntemlerden biride karar ağaçlarıdır (Decision Tree). Karar Ağaçları düğüm (Root), dallar (node) ve yapraklardan (leaf) oluşur. Karar düğümü veriyi dallara ayırarak yerleştirir. Karar ağaçlarında, sınıflandırmaya ayıracak asıl kısım, karar düğümüdür. Bu karar düğümünü geçen veri bir sınıf oluşturmuyorsa, yeni bir dallanma oluşturarak başka bir düğüm oluşturur. Her düğüm karar verme mekanizmasıdır, şartları sağlayan veri düğümde yerleşerek bir yaprak oluşturur. Karar Ağaçlarında işlem kök düğümden başlar ve yapraklara doğru ilerler[10][12].

Karar ağaçları oluşturulurken güvenilir bir test verisiyle ağacın yapısı belirlenir, model oluşturulur ve test edilecek veri belirlediğimiz kıstaslara göre düğümden başlar, dalları oluşturarak yapraklara yerleşir[10][12]. Şekil 2'de gösterildiği gibi "Yaş"a bağlı veri değerinin, karar düğümlerinden geçerek yapraklara yerleşimi yapılır.

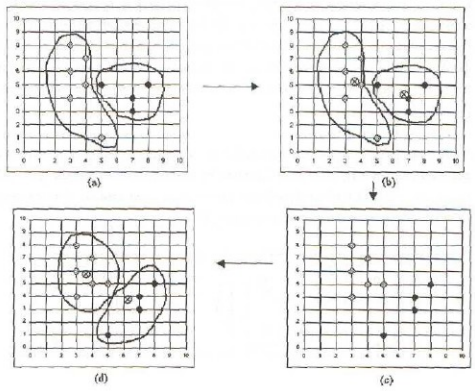


Şekil 2: Karar Ağacı Yapısı

2.1.2 Kümeleme

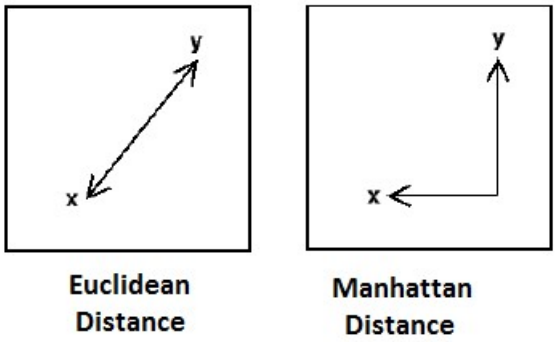
Kümeleme (Clustering) veri madenciliği, istatistik, biyoloji ve makine öğrenmesi gibi alanlarda çok sık kullanılır[10]. Kümeleme en bilindik Eğiticisiz öğrenme (Unsupervised learning) çeşitlerindendir. Kümelemenin en önemli özelliği eğitim setine ihtiyaç duymamasıdır, bu

yönteme eğiticisiz öğrenme denilmesinin sebebi de budur. Şekil 3'de gösterildiği gibi benzer verileri aynı kümelere yerleştirme suretiyle sınıflandırma yapmaya çalışılır[10][18].



Şekil 3: K-Means Yöntemiyle Kümeleme[10]

Günümüzde veri tabanlarında toplanan verinin artmasıyla kümeleme algoritmalarının işlevleri giderek önem kazanmaktadır[10]. Yapısal sorgu dili (SQL) özellikle Web tabanlı sistemlerde, toplanan verilerin büyük olması ve bu verilerin karmaşık yapılarından dolayı, sonuç bulmada ve işlem süresinin uzaması gibi konularda zorlanmaktadır. SQL'in yetersiz kaldığı bu durumlarda kümeleme algoritmaları tercih edilmektedir.



Şekil 4: Euclidean Distance, Manhattan Distance

Kümeleme yöntemi, verilerin birbirine uzaklıkları baz alınarak, yakın verilerin birbirleri ile kümelenmesi olarak özetlenebilir. Bu süreçte veriler arası uzaklıklar bulunurken, Şekil 4'de gösterilen Euclidean Distance veya Manhattan Distance yöntemleri kullanılır[9][16][17][19]. K-Means algoritması kümelemede kullanılan bir algoritmadır. Bu yöntemle elimizdeki n tane

veri bizim belirlediğimiz k adet kümeye ayrılır. Burada küme sayısı n den küçük ya da eşit olmalıdır(k<=n).

K-Means yönteminde k adet nesne kümenin merkezi seçilir, geri kalan her nesne kendilerine en yakın küme merkezlerine göre kümeye yerleşir ve kümenin merkezi yeniden hesaplanır. Bu değer kümenin yeni merkezi olarak kabul edilir. Bu işlem tüm nesneler kümeye yerleşinceye kadar devam eder[10][20].

2.1.3 Birliktelik Kuralı

Birliktelik kuralları (Assocation Rules) büyük veriler içerisinde verilerin veya veri kümelerinin ilişkilerini açıklamaya çalışarak bilgi çıkarmaya çalışılmasıdır. Birliktelik kurallarını açıklayan en iyi örnek market-sepet uygulamasıdır. Bu tür bir bilgi çıkarımında ürünlerin yerleştirilmesinden tutun, stok miktarı, reklam ürünlerinde hangi ürünlerin yan yana geleceğine kadar birçok konuda karar verme mekanizmasına yardım eder.

A=>B [destek=%d,güven=%d]

Formül 1:Birliktelik Kuralı

Birliktelik kuralı Formül 1'deki gibi ifade edilir. Bu formül A olay ile B olaylarının birbirleriyle ne kadar ilişki halinde olduğunu gösterir[10]. Bu ilişki destek ve güven ifadeleri ile gösterilir. Buna göre;

- destek: A olayının oluşum sıklığını,
- güven: tanımlanan birlikteliğin kabul edilebilirliğini ifade eder.

Birliktelik Kuralını bir örnekle açıklayacak olursak. Bir marketten alışveriş yapan değişik müşterilerin aldıkları ürünlerin incelenmesinde, bu müşterilerin çoğunlukla süt, bebek bezi, et, kuruyemiş ve su gibi ürünler aldıkları görünüyor. Bu ürünlerin alım sıklıkları tek tek incelendiğinde en fazla alınan ürünün çocuk bezi olduğu görünüyor. Çocuk bezi alan müşterilerin çoğunlukla süt aldıkları görülür. Bu durum;

$A_{[\text{çocuk bezi}]}=>B_{[\text{süt}]}$ [destek=%2,güven=%60]

Formül 2: Birliktelik Kuralının değerlerle ifade edilmesi

Formül 2'ye göre markette toplam alınan ürünler arasında %2 çocuk bezi alınmıştır. Çocuk bezi alan müşteriler incelendiğinde de, "çocuk bezi alan müşterilerin %60'ı süt de alır" sonucunu çıkarabiliriz.

Bu yöntemle yapılan analizlerde kullanıcı tarafından minimum destek eşik değeri ve minimum güven değeri belirlenir. Birliktelik ilişkileri belirlenmeye çalışılırken, önce sık tekrarlayan öğeler bulunur ve en çok tekrarlayan öğe ile birlikte geçen diğer öğeler arasında güçlü bağ oluşturduğu görülür[22]. Birliktelik ilişkisi bulunurken en çok kullanılan yöntem Apriori Algoritmasıdır[10]. Buna ek olarak AIS (Agrawal, Imielinski, Swami), PARM (P-tree based Association Rule Mining) yöntemleri de sıklıkla kullanılır[9][10][23].

2.2 METİN MADENCİLİĞİ (TEXT MINING)

Metin Madenciliği, yarı yapısal veriler (metin, resim, ses vb.) üzerinde bilgiye ulaşma yöntemidir[9]. Veri madenciliğinde veriler sayısaldır ya da sayısal bir yapıya dönüştürülebilir özelliktedir. Metin madenciliğinde ise veriler yarı yapısal bir yapıdadır[24]. Yarı yapısal veriler üzerinde işlem yapmak oldukça zordur. Yapısal sorgu dili olan SQL (Structured Query Language) verileri depolarken aynı türden verileri aynı alan içerisinde tutar ve bu alanlar üzerinde çeşitli sorgulamalar yapılabilir. Yazı türünde(text, char) tutulan bir veriden "hangi tür bilgiyi içeriyor" gibi soruyu SQL dili ile yanıtlamak neredeyse imkânsızdır.

Metin Madenciliğinde yarı yapısal verilerin kullanılabilir bir bilgiye dönüştürme işlemleri, Bilgiye Erişim, Bilgi Çıkarımı ve Web Madenciliği şeklinde üç grupta toplanabilir[25].

2.1.1 Bilgiye Erişim

Bilgiye erişim (Information Retrieval-IR) sistemlerinde kullanılan standart iki ölçüt vardır, doğruluk (Recall) ve duyarlılık (Precision)[26].

Doğruluk aradığımız bilginin toplam bilgiye oranıdır[9][26]. Duyarlılık; ayrılmış verinin kullanılabilir verilere oranı şeklinde hesaplanır[9][26]. Doğruluk ve Duyarlılık Formül 3'deki gibi hesaplanabilir.

$$\text{Doğruluk} = \text{Recall} = \frac{\sum ilgilibelge}{\sum toplamdöküman}$$

$$\text{Duyarlılık} = \text{Precision} = \frac{\sum ilgiliverisayısı}{\sum kullanılanverisayısı}$$

Formül 3: Doğruluk ve Duyarlılık Hesaplanması

Terim frekansı, hesaplanarak metinlerin hangi konularda bilgi içerdiği bulunmaya çalışılır. Bir dokümanda sıkça geçen terimler, o dokümanın hangi sektörle ilgili olduğu konusunda ipuçları içerir.

2.1.2 Bilgi Çıkarımı

Metinler üzerinde Doğal Dil İşleme yöntemleri kullanarak özetleme, benzer metinleri çıkarımları yapılmaktadır[9][27]. Benzer metinler çıkarımlarında anahtar kelime (keyword) bağlı olarak yapılmaktadır[9]. Sistemin başarı oranı, anahtar kelimelere bağlıdır.

2.1.3 Web Madenciliği

Web madenciliği; site içerikleri, sayfa yapıları, web linkleri istatistikleri, sosyal medya incelemeleri gibi web ile ilişkili olan verilerin analizlerinden (Benzer haber tespiti, duygu analizi, çevrimiçi satış strateji belirleme analizleri vb.) oluşmaktadır [3][4][6][28].

2.2. VEKTÖR UZAYI MODELİ

Metinler üzerinde işlem yapmak oldukça zordur. Bu zorluğun başlıca nedenlerinde biride yarı yapısal yapıda olmasıdır. Bu yapıdaki verileri ölçme sistemlerine uyarlamak zordur. Vektör Uzayı Modeli (Vector Space Model) yarı yapısal verilerin (text, char vb.) sayısal (metric) bir yapıya uyarlama çalışmasıdır. Metinler için uyarlanacak sayısal yapı metni bir vektör yada doğru şekline dönüştürmek olacaktır[29][30][31]. Metinler bir vektör ya da doğruya dönüştürülürken her kelime bir terim olarak alınır ve metinde terimin geçme sayısı o terimin

ağırlığını ifade eder. Eğer kıyaslanacak başka bir metin varsa tek bir metinmiş gibi kabul edilir ve bütün terimler bir **d** doğrusun da toplanır. Bu doğru da sözlük olarak ifade edilmektedir[31]. Bu sözlüğün çıkarılmasında iki yöntem vardır. Birincisi bitsel tanımlama, ikincisi ise frekansa göre tanımlamadır.

d = (ω1*(t1)*, ω2(*t2*), ω3(*t3*), …,ωi(*ti*))

$$\omega i = \sum_{i=1}^{toplamterim} ti$$

ωi=*ti* teriminin geçme ağırlığı, *ti*=terim

Formül 4: Vektör Uzayı Modelinde Ağırlıkların Hesaplanması.

Formül 4'de gördüğümüz gibi oluşturacağımız denklem doğrusaldır. Formül4'de görüldüğü gibi ω katsayısı iki şekildedir. Hesaplamalarımızda terimlerin metinde geçme sıklığı önemliyse(frekans tanımlama)ω katsayısını kullanırız, değilse bitsel tanımlama (boolean) kullanarak 1 yada 0 sayıları ile ifade edilir. Örnek olarak Şekil 5'i incelersek soru ve cevapların bitsel olarak Vektör Uzayı Modeline göre doğruya dönüştürülmesi gösterilmiştir.

i	*Cevaplar*
1	Osmanlı İmparatorluğuna Bursa, Edirne ve İstanbul başkentlik yapmıştır.
2	Osmanlı İmparatorluğuna Edirne ve İstanbul başkentlik yapmıştır.
3	Konya ve İstanbul başkentlik yapmıştır.
4	Son başkenti İstanbuldur.

Sözlük=d_{Cevap}={ Bursa, Edirne, İstanbul}
d1={1, 1,1}
d2={0,1,1}
d3={0,0,1}
d4={0,0,1}

Şekil 5: Vektör Uzayı Modelinde Bitsel Tanımlama

Şekil 6'da gösterildiği gibi frekansa göre tanımlamada metinde geçen terimlerin sıklıkları doğrumuzdaki katsayıları oluşturur, bu katsayılar frekansını gösterir. Bu tür

hesaplamalar,genellikler metin sınıflandırma gibi belirli gruplar altında toplanmak istenirse kullanılır[29][30][31].

d1={3, 1,5}
d2={0,4,1}
d3={0,0,1}
d4={0,2,1}

Şekil 6: Frekans Tanımlama

2.3 KOSİNÜS BENZERLİĞİ

Kosinüs benzerliği (Cosıne Similarity), trigonometrideki kosinüs teoremini temel alır. Şekil 7'de gösterildiği gibi vektörel düzlemde oluşturduğumuz iki doğrunun arasındaki açının kosinüsü, benzerlik oranını ölçmek için kullanılır[32].

$$\mathrm{Cos}(\theta) = sim\left(d_{cevap}, d_{öğrcevap}\right)$$

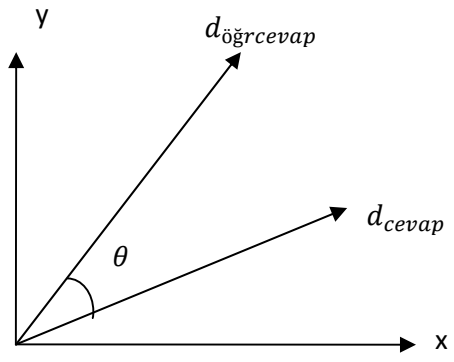
$$Cos\left(d_{cevap}, d_{öğrcevap}\right) = \frac{d_{cevap}.\, d_{öğrcevap}}{\left|\left|d_{cevap}\right|\right| * \left|\left|d_{öğrcevap}\right|\right|}$$

$$Cos\left(d_{cevap}, d_{öğrcevap}\right) = \frac{\sum_{i=1}^{n} d_{cevap} x d_{öğrcevap}}{\sqrt{\sum_{i=1}^{n} (d_{cevap})^2 X \sum_{i=1}^{n} (d_{öğrcevap})^2}}$$

$d_{cevap}.\, d_{öğrcevap}$: iki dokümanın vektör çarpımı

$||d_{cevap}||$: d_{cevap} dokümanının uzunluğu

Formül 5: Kosinüs Benzerliği Formülü



Şekil 7: Kosinüs Teoremi

Formül 5 ve Şekil 7'de gösterildiği gibi "d_{cevap}" ve "$d_{öğrencicevap}$" vektörarasındaki kosinüs bağlantısı için iki vektörün çarpımının (dot product) "d_{cevap}" ve "$d_{öğrencicevap}$" boylarının çarpımına oranı alınmıştır.

	Cevaplar
d_{cevap}	Bursa, Edirne, İstanbul
$d_{öğrencicevap}$	Osmanlı İmparatorluğuna Edirne ve İstanbul başkentlik yapmıştır.

d_{cevap} ={1, 1, 1}
$d_{öğrencicevap}$={0, 1, 1}

d_{cevap} = 1, 1, 1
$d_{öğrencicevap}$= 0, 1, 1
$d_{cevap} \bullet d_{öğrencicevap}$= 1*0 + 1*1 + 1*1 = 2
|| d_{cevap} || = $(1*1+1*1+1*1)^{0.5} = (3)^{0.5}$ = 1,732
|| $d_{öğrencicevap}$|| = $(0*0+1*1+1*1)^{0.5} = (2)^{0.5}$ = 1,414
cos(d_{cevap}, $d_{öğrencicevap}$) = 2/2.449=0.816

Yukarıdaki örnekte, Formül 5'e göre "d_{cevap}" ve "$d_{öğrencicevap}$" doğrularının "Vektör Uzayı Modeline" göre hesaplanması gösterilmiştir. Yukarıda ki örnekte yarı yapısal bir yapıda olan verilerin sayısal veya lineer bir yapıya dönüştürülmesi gösterilmiştir.

Vektör uzay modeline dönüştürülen metinler bir doğruyu ifade eder. Bu iki doğrunun arasındaki açının kosinüsü bize benzerlik oranını verecektir. "d_{cevap}" ve "$d_{öğrencicevap}$" doğrularının ifade ettikleri metinler aynı kelimelere sahip olma sayısı arttıkça aralarındaki açı

azalacak ve kosinüs benzerlikleri 1'e yaklaşacaktır. Bir başka deyişle "d_{cevap}" ve "$d_{öğrencicevap}$" aynı metinse, doğrular birbirine yaklaşacak ve aralarındaki açı 0^0 olacaktır(Cos(0)=1). "d_{cevap}" ve "$d_{öğrencicevap}$" doğrularını ifade eden metinler tamamen farklı kelimelerden oluşmuşsa aralarındaki açı genişleyecek ve 90^0 olacaktır, cos(90)=0 değerini alacaktır.

2.4 DOĞAL DİL İŞLEME

Veri madenciliği ile ilgi yazılımlarda dilin yapısı oldukça önemlidir. İnsanlar, düşüncelerini, duygularını veya isteklerini farklı kelimelerle veya kelimelere getirdikleri eklerle ifade edebilirler. Buda dilin güzellikleri arasında sayılabilir. Fakat bu durum metin madenciliğiyle uğraşan kişiler için geçerli değildir. Metin madenciliğinde farklı kelimelerin kullanılması kelime çeşitliliğini artırır, bu da analiz sürecinde işleri içinden çıkılmaz hale getirebilir. İfadeleri değerlendirmede kelimelerin yapısı(isim, sıfat fiil vb) oldukça önemlidir. Ayrıca gözden kaçırmamamız gereken bir diğer ayrıntıda insanların ifade ederken kendi dilleri konusundaki dikkatsizlikleri, yazım yanlışlarıdır[33].

Doğal bir dili anlama süreci; çözümleme, yorumlama ve kelime üretme konularını ele alan bir daldır. Doğal Dil İşlemenin(DDİ) temel amacı doğal dillerin yapısını ve işleyişini anlamak, bilgisayara dil çevirisini yaptırabilmek, bilgisayar ile insan arasında iletişim kurabilmektir. DDİ, veri madenciliğiyle gelişmiş ancak günümüzde kendi başına bir dal olmuştur.

DDİ beş ana seviyede incelenebilir.

1- Ses Bilimi (Pholoji): Seslerin, harflerin dilde nasıl kullanıldığını inceler.
2- Biçim Bilimi (Morpholoji): Kökler ve eklerin ayrıştırılması ve tespitini inceler.
3- Söz Dizimi (Syntax): Kelimenin cümle oluşturabilmesi, cümle içersindeki yapısal görevini inceler.
4- Anlam Bilimi (Semantics) : Sözcüklerin bir araya gelerek anlattıkları eylem ya da duygular açısından inceler.
5- Kullanım Bilimi (Progmatics): Cümlelerin ya da kelimelerin farklı anlamlarda kullanıldığı durumları inceler[33].

Günümüzde Doğal Dil İşleme ile ilgili çalışmalardan kısaca bahsedecek olursak.

2.4.1 Soru Cevap

Karmaşık soruları cevaplamaya çalışan sistemlerdir[34][35]. Bunun en iyi örneği IBM firmasının Watson Projesi'dir.

Watson Projesinde, kişilerin karmaşık sorulara hızla bilimsel yanıtlar bulabilmelerini sağlaması amaçlanmıştır. Bu teknolojinin, sağlık hizmeti gibi alanlarda hastalığın doğru şekilde tanılanması, çevrimiçi self-servis yardım masalarını iyileştirme, turistlere ve vatandaşlara şehirler hakkında belirli bilgileri sağlama ve telefon aracılığıyla müşteri desteğini isteme gibi konularda ve daha pek çok alanda uygulanması hedeflenmiştir[36].

2.4.2 Bilgi Çıkarımı

Bilgi çıkarımı (Information Exraction) metinleri analiz ederek istenmeyen e-postaların (spam) hangi konuda yazıldığını bulmaya çalışır. Bu çalışmalar kullanıcıyı sıkan reklam içerikli e-postalardan kurtarır. Bilgi çıkarımının bir başka uygulaması ise haber metinlerini gruplayarak çeşitli analizler yapmadır. Ekonomi, spor gibi alanlarda yapılan gruplamalarla insanların eğilimleri ölçeklenebilir ya da yönleri konusunda tahminlerde bulunabilir.

2.4.3 Duygu Analizi

Duygu analizi (Sentiment Analysis) günümüzde kullanımı hızla yaygınlaşan sosyal medya (Facebook, Twitter, Google+, Instagram vb) kullanıcılarının paylaşımlarından yola çıkarak duygularını ve yönelimlerini belirlemeye çalışır. Sosyal medya üzerinden toplanan verilerle mutluluk, üzüntü analizi yapılabilir.

2.4.4 Makine Çevirisi

Makine çevirisi (Machine Translate) bilgisayar yardımıyla dilden dile çeviri yapmayı hedefler. Bunun en yaygın örneklerinden biri Google Translate'dir.

2.4.5 Diğer Çalışmalar

Doğal Dil İşlemede yapılan diğer tür çalışmalardan biride ses - konuşma tanıma ve metni konuşmaya dönüştürme projeleridir. Özellikle akıllı telefonlar ve tabletlerde geliştirilen ses tanıma, sesli arama gibi çalışmalarda Doğal Dil işleme alanlarındandır.

Yukarıda belirtilmiş olan Doğal Dil İşleme uygulama alanlarında karşılaşılan sorunlar, bazı dillerdeki kuralsız kullanımlar, deyimlerin farklı farklı anlamlarda kullanılması, kısaltmalar, diğer dillerden dile geçen kelimelerin farklı anlamlarda kullanılması, eş anlamlı ve mecaz kelimelerin kulanım farklılıklarıdır[37].

Türkiye'de Birçok üniversite Doğal Dil işleme ilgili, ekipler kurarak akademik çalışmalar yapmaktadır. Bu üniversiteler Doğal Dil işlemenin gerekliliğini anlayarak ekipler oluşturmuş, çalışmalarını bu ekiplerle akademik düzeyde sürdürmektedir[38][39]. Türkçe ile ilgili yapılan Doğal Dil çalışmalarının en kapsamlısı TÜBİTAK bünyesinde geliştirilen Zemberek Projesidir.

2.5 ZEMBEREK DDİ KÜTÜPHANESİ

Türkçe dilleri için DDİ çalışmaları neticesinde oluşturulmuş bir çalışmadır. Bu çalışma açık kaynak kodlu olarak geliştiricilere sunulur, yani DDİ ile çalışma yapacaklar kütüphaneyi ve kodları istediği gibi kullanabilir ve değiştirebilirler[40].

Zemberek, TÜBİTAK Bilgem tarafından geliştirilen açık kaynak kodlu dağıtımında da sistem genelinde yazım denetimini sağlayan bir araç (tool) olarak sunulmaktadır[41].

Zemberek DDİ Kütüphanesi, Java dili kullanarak geliştirilmiş ve halen geliştirilmeye devam etmektedir. Bu çalışmada Java dilinin tercih edilme sebebi platformlardan bağımsız olarak çalışabilmesidir[42].

Bunlara ek olarak Zemberek, sadece Türkçe dilinde değil Azeri ve Tatarca dillerinde de ayrıştırma yapabilmektedir. Ayrıca Kırım ve Arap alfabesi kullanan Türk dilleri için çalışmalar devam etmektedir[43].

Zemberek DDİ Kütüphanesi ile yazım denetimi, biçimsel analiz, ek ayırma, sözcük türetme, sözcük önerme ve heceleme gibi işlemleri yapabilir. Ayrıca Zemberek DDİ Kütüphanesi yazım yanlışları için kelime önerisinde bulunabilmektedir.

Zemberek DDİ Kütüphanesi Pardus İşletim Sisteminde kullanılan diğer açık kaynak kodlu yazılımlara (Firefox, Thunderbird, OpenOffice vb) yazım önerisi sunabilir. Pardus işletim sisteminde Zemberek Sunucu, yazım kontrol (Kspell- Zspell) programlarının oluşturdukları isteklere cevap vererek çalışır[44]. Bu tür Genel Kamu Lisanslı (General Public License, GPL) yazılımlarda "açık kaynak kodlu bir DDİ Kütüphanesi" geliştirilmesine büyük önem verilmiştir.

Zemberek DDİ Kütüphanesinin, bu çalışmada kelimelerin köklerini bulmak için kullanılması hedeflenmektedir.

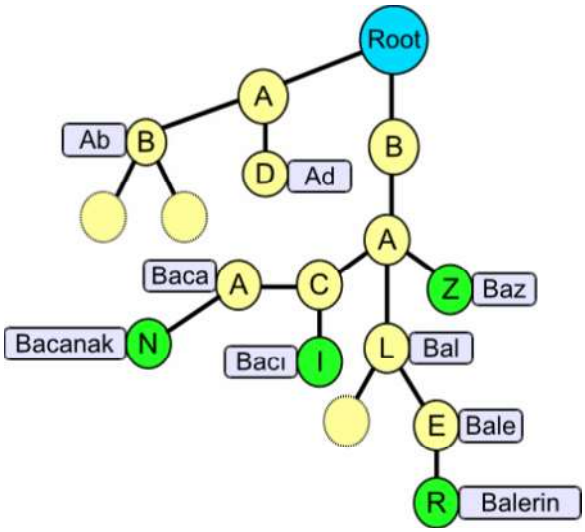
Zemberek DDİ Kütüphanesi adresinden[1] indirilebilir. Zemberek DDİ Kütüphanesi Microsoft Visual Studio ortamlarında kullanılmak üzere "NZembrek.dll" isminde bir dll dosyası sunar. Bununla beraber kelimelerin bulunduğu bir dosya sunar. Kelime haznesinde şuanda 1.400.000 kelime vardır ve bu kelimeleri ayrıştırabilir.

Zemberek DDİ Kütüphanesinin başarıyla çözüme ulaştırdığı konulardan birisi kelimelerdeki yumuşamayı ayrıştırabilmesi (örneğin: burun- burnunda), diğeri de yabacı dilden geçen kelimelerdeki ekleri ayırabilmesidir (örneğin: red- reddi, Arapça kökenli)[43].

Zemberek DDİ kütüphanesinde bulunan kelimeler "trie ağaç" (retrieval) yapısında tutulur. Bu şekildeki yapı sözcüklere erişimi kolaylaştırmakta yaklaşık; 75.000 kelimenin olduğu bir kelime havuzunda saniyede 12000 kelime tarayabilmektedir[40].

Zemberek DDİ Kütüphanesi kelimeleri "trie ağaç" yapısında Şekil 8'deki gibi tutar. Kök (Root) kelimeye ulaşmak en başta hedeflenendir[45]. Kök bulma morfolojik(biçimbilim) analiz için gereklidir[46]. "Trie Ağaç" modeli sözlük ve sözcük aramalarında en çok tercih edilen yöntemlerden biridir. Bu model ile kök kelimeye kolay ve hızlı bir şekilde ulaşılabilir.

[1] http://code.google.com/p/zemberek



Şekil 8: Zemberek "Trie Ağaç" Yapısı[40][43]

Zemberek Projesini kullanabilmek için öncelikle projemizin kütüphanelerine Zemberek DDİ Kütüphanesinin classlarını Kod 1'deki gibi eklemeliyiz.

```
using net.zemberek.erisim;
using net.zemberek.yapi;
using net.zemberek.tr.yapi;
```

Kod 1:Zemberek DDİ Kütüphanesini Referans Verme

Ayrıca Zemberek DDİ Kütüphanesini eklemek için Microsoft Visual Studio'da Solution Explorer bölmesindeki References kısmında NZemberek.dll kütüphanesi referans gösterilmelidir.

```
Zemberek z = new Zemberek(new TurkiyeTurkcesi());
Kelime[] cozumler = z.kelimeCozumle(giris);
System.Console.WriteLine("cozumlemeler:");
        foreach (Kelime kelime_ in cozumler){
System.Console.WriteLine(kelime_);
         }
Console.Write("heceleme sonucu:");
    String[] heceler = z.hecele(giris);
```

```
Array.ForEach<String>(heceler, HeceYazAction);
Array.ForEach<String>(heceler, HeceYazAction);
Console.WriteLine("\nÖnerme sonucu: ");
String[] kelimeOner = zemberek.oner(giris);
```

Kod 2:Zemberek DDİ Kütüphanesinin Kullanımı

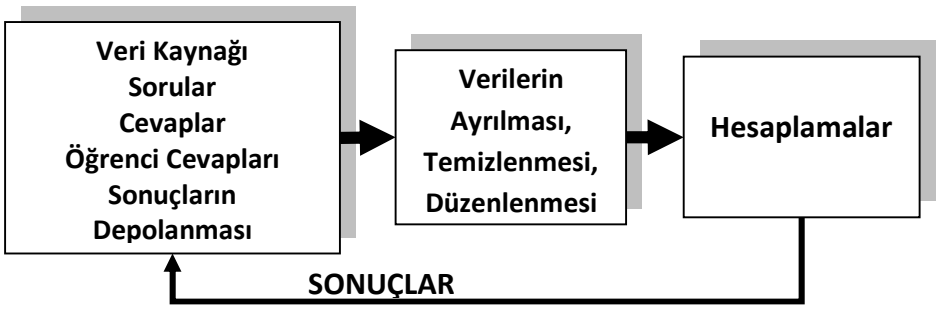
Kod 2'deki gibi kullanarak çözümleme ve heceleme işlemlerini yapar. Bunun sonucunda Şekil 9'daki gibi bir sonuç alırız.

```
KELİME : bulaşıcı
cozumlemeler:
{Icerik: bulaşıcı Kok: bulaşıcı tip:SIFAT}  Ekler:ISIM_KOK
{Icerik: bulaşıcı Kok: bulaş tip:FIIL}  Ekler:FIIL_KOK + FIIL_TANIMLAMA_ICI
{Icerik: bulaşıcı Kok: bula tip:FIIL}  Ekler:FIIL_KOK + FIIL_BERABERLIK_IS +
FIIL_TANIMLAMA_ICI
{Icerik: bulaşıcı Kok: bu tip:ZAMIR}  Ekler:ZAMIR_KOK + ISIM_DONUSUM_LE +
FIIL_BERABERLIK_IS + FIIL_TANIMLAMA_ICI
heceleme sonucu: - bu - la - şı - cı
Önerme sonucu:
bulatıcı
bulayıcı
bulaşışı
```

Şekil 9: Zemberek DDİ Kütüphanesi ile Ayrıştırılmış Kelime

3 PROGRAM GELİŞTİRME

Klasik soruları değerlendirirken ilk yapılacak işlemler; veri kaynağından çekilen doğru cevabı ve verilen cevabı kelime köklerine ayırmak, daha sonra da aradığımız kelimeleri puanlamak olacaktır. Programın işleyişi temel olarak 3 aşamada gerçekleşir.

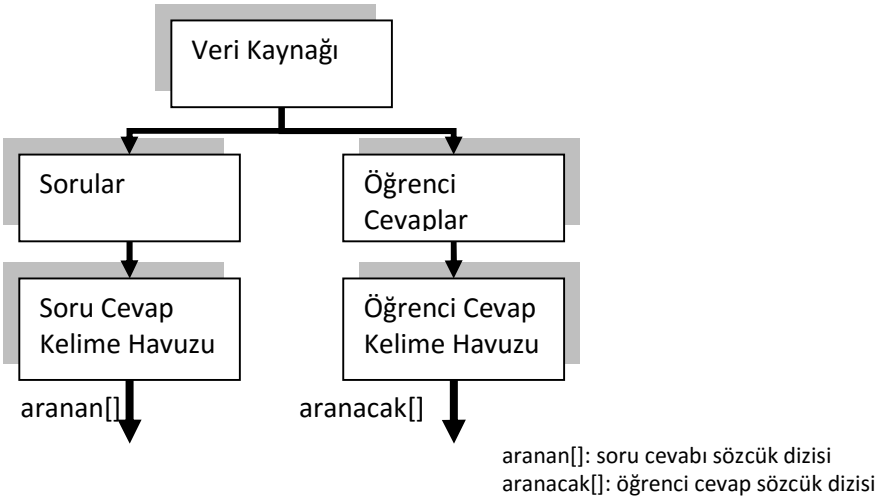


Şekil 10: Programın İşleyiş Adımları

3.1 Verilerin Çekilmesi

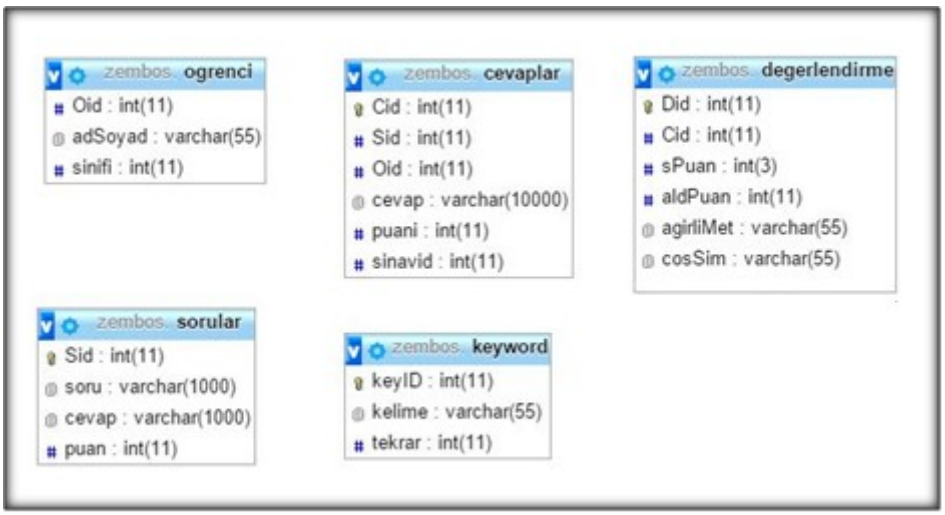
Tasarlayacağımız uygulamanın safhalarını, Şekil 10'da gösterildiği gibi, "Veri Kaynağı", "Verilerin İşlenmesi" ve "Hesaplamalar" şeklinde özetleyebiliriz. Soruların cevapları ile bir kelime havuzu, test edeceğimiz öğrencilerin cevaplarını da başka bir kelime havuzu olarak tasarladık. Veri kaynağımızdaki verilerin çekilmesi de Şekil 11'de gösterilmiştir. Bu uygulamada analiz edilecek olan verilerin veri kaynağında tutulması gerekiyor. Bahsi geçen veri kaynağı veritabanındaki tablolardır, bu tablolar Şekil 12'de gösterilmiştir.

.



Şekil 11: Veri Kaynağından, Verilerin Çekilmesi.

Köklere ayrılacak verileri tutmak için MySQL- DBMS (Database Management System) veritabanı sistemini kullandık. MySQL-DBMS Veritabanı Yönetim Sistemleri ücretsizdir. Birçok işletim sisteminde çalışabilen versiyonu mevcuttur ve web sitelerinde de tercih edilen bir Veritabanı Yönetim Sistemidir. Ayrıca projemizde öğrencilerin klasik sorulara cevabını tutan, başka bir sitenin[2] veritabanından verileri çekecektir. Bu tez çalışması için tasarlamış olduğumuz veritabanındaki tablolar ve alanlar Şekil12'de verilmiştir.



Şekil 12: Veritabanı Tablolar ve Alanlarının Şeması

[2] http://sinav.btteknik.com

Şekil 12'deki "ogrenci" tablosu öğrenci adı soyadı ve sınıf bilgileri tutar. Programda bu verilere ulaşmak için "Oid" alanı birincil anahtar (primary key) olarak tanımlanmış, "Oid" alanı her öğrenci için tanımlayıcı ve tekrarsız bir alandır.

"sorular" tablosu soru, cevabı ve sorunun kaç puan olduğu bilgilerini tutmaktadır. "Sid" eklenen her soru için bir ID (identity) oluşturur. "Sid" alanı birincil anahtar olarak tanımlanmıştır.

"cevaplar" tablosunda, "Sid" sorular tablosuna kaydedilmiş soru numarasını, "Oid" öğrenci tablosunda kayıtlı öğrenci numarasını, cevap alanı öğrencinin verdiği cevabı ve "puani" alanı öğrencini cevaptan kaç puan aldığı bilgisini tutar.

"keyword" tablosu, soruya verilen bütün cevaplarda geçen kelimeleri ve tekrar sayılarını tutmaktadır.

"degerlendirme" tablosu, hesaplamaların ve teste tutulacak verilerin tutulduğu tablodur. "degerlendirme" tablosunda "Cid" cevaplar tablosundan değerlendirilecek cevabın ID'sini tutar. Testte kullanılacak olan sorunun puanı ve aldığı puan bilgisi de burada tutulmaktadır. Hesaplamalarımız sonucu oluşan veriler bu tabloya kaydedilecektir.

3.2 Programlama Dili Seçimi ve Ara yüz Oluşturma

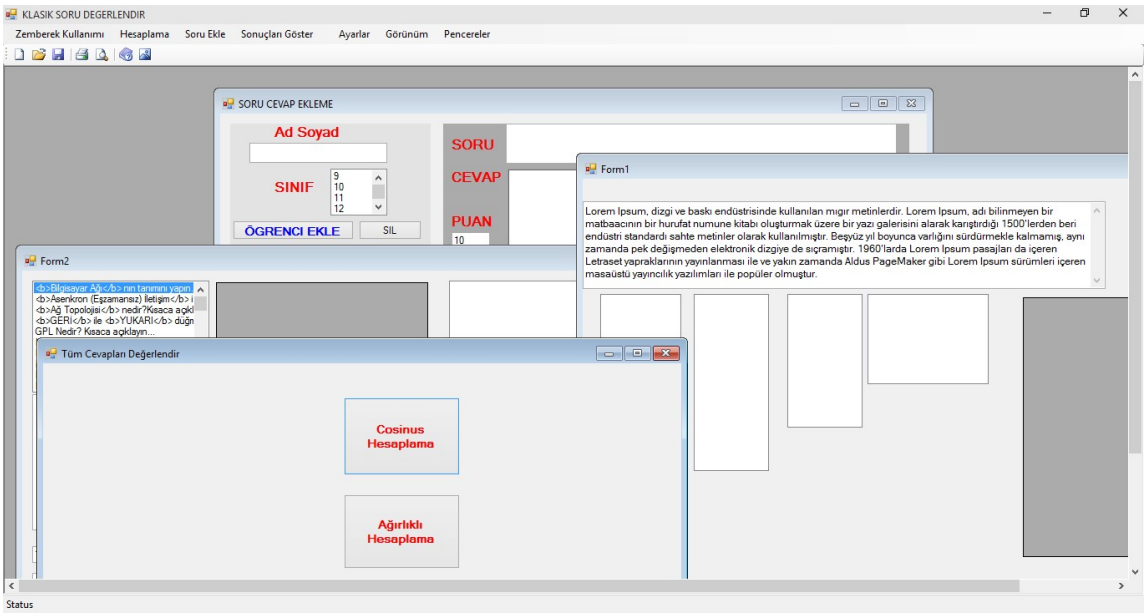
Program hazırlanırken kullanılan dil C# dilidir. C# dili nesne tabanlı programlama (OOP-Object Oriented Programming) dilleri arasında olup, kullanımı kısa sürede yaygınlaşmıştır. C# dili Microsoft tarafından oluşturulmuş ve geliştirmesi de devam etmektedir. C# programı için kullanılan geliştirme ortamı Visual Studio'dur. Visual Studio'nun Express sürümü adresinden[3], ücretsiz indirilip kullanabilir.

C# dili kullanıcıya bazı kütüphaneleri hazır olarak sunmaktadır, buda program geliştirenlerin işini kolaylaştırmaktadır. Program geliştiriciler kütüphanelerden yararlanabilmeleri için "Microsoft. NETFramework" kütüphanesini kurmaları gerekmektedir. Microsoft firması NETFramework 1, 2, 3.5, 4, 4.5 olarak yayınlamıştır. Bizim hazırladığımız proje "NETFramework3.5" kütüphanesi ile çalışmaktadır. Sadece geliştirme ortamında değil aynı

[3] https://www.visualstudio.com/tr-tr/products/visual-studio-express-vs

zamanda kullanıcının kullanacağı ortamlarda da “NETFramework3.5” kurulu olması gerekmektedir.

Windows uygulamalarında kullanılan formlar Tekli form (Single) ya da çoklu form (MDI-Multiple Document Interface) şeklindedir. Bu çalışma için oluşturulan uygulama da birden fazla form olacağı için MDI form kullanılmıştır. Şekil 13’de gösterilen bu ana form (MDI); hesaplama, soru ekleme, değerlendirme ve çeşitli ayarlamaların yapıldığı formları kendi içerisinde barındırmaktadır.



Şekil 13: Programın Çalıştığı Ana Form

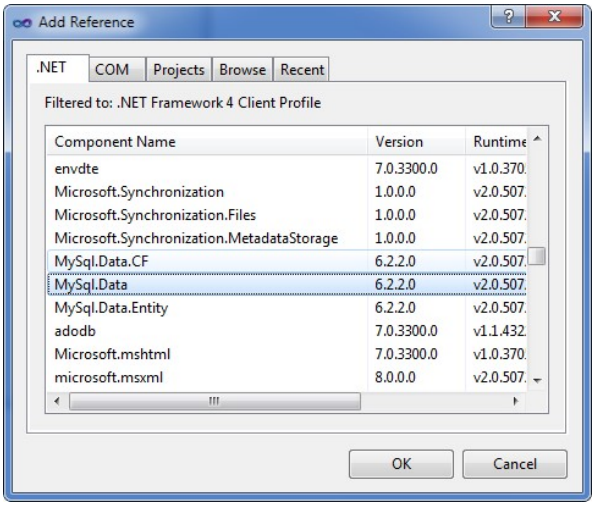
Kullandığımız formun IsMdiContainer özelliği True seçilirse, diğer formları içersinde açan MDI form özelliği verilmiş olur. Burada gösterilmek istenen her form için Kod 3’deki kod kullanılmalıdır. MDI form seçilince “Parent” özelliği diğer formları barındıran ebeveyn özelliği kazanmış olur. MDI form içersinde açılan her bir form yavru (Child) form özelliği kazanır.

```
Form6 mdiForm = new Form6();
mdiForm.MdiParent = this;
mdiForm.Show();
```

Kod 3: MDI Form’da Child Form Açma

3.3 Kütüphaneleri Referans Verme

Daha önce de belirtildiği gibi sistemin verileri MySQL Veritabanı Yönetim Sisteminde tutulmaktadır. MySQL'in bağlantı ve sorgulama işlemleri için MySQL tarafından yazılmış olan bağlantı aracı"MySQL net connector"dür. "Mysql-connector-net-6.9.4"versiyonunu, adresinden [4] ücretsiz olarak indirilip, kurulabilir. Visual Studio ortamında MySQL'e bağlanabilmek için, Şekil 14'deki gibi Add References bölümünden ".NET" Sekmesinden "Mysql.Data" seçilip eklenir. Bu işlem MySQL Veritabanı Yönetim Sisteminin çalışabilmesi için gerekli olan dll dosyalarını programımızın kütüphanelerine ekler.



Şekil 14: MySQL Data Kütüphanesinin Eklenmesi

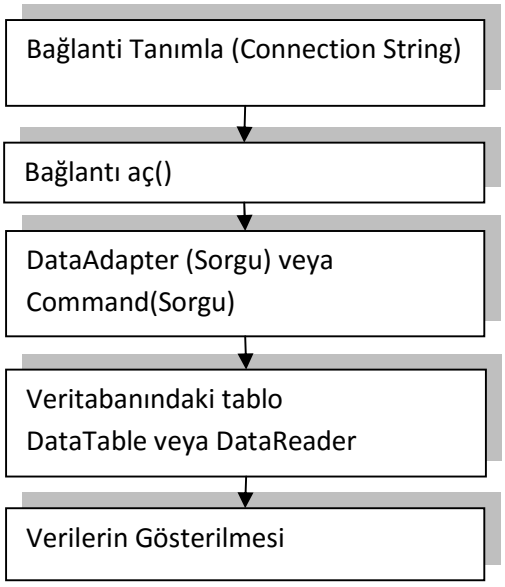
Şekil 14'de referans olarak eklenen MySQL veritabanını kullanabilmemiz için, sayfa içersindeki kütüphaneler kısma "using MySql.Data.MySqlClient;" satırı mutlaka eklenmelidir.

Şekil 15'deki işlem basamaklarında gösterildiği gibi ilk olarak MySQLConnection sınıfı kullanarak "baglanti" isminde MySQLConnection nesnesi türetilerek, "baglanti" nesnesi tanımlanır. "baglanti" nesnesi "ConnectionString" içerir. "ConnectionString" nesnesi veri kaynağının adresi, bu çalışma da yerel bilgisayarda çalışıldığı için "localhost" olarak tanımlanmıştır. "ConnectionString" nesnesinde veritabanı kullanıcı adı ve şifre bilgileri de yazılmalıdır.

[4] http://downloads.mysql.com/archives/c-net/

“ConnectionState” bağlantımızın durumunu kontrol ediyor, eğer bağlantı açık değilse baglanti.Open() metodu kullanarak bağlantı açılır ve veri akışına izin verilmiş olur.

Command ve DataAdapter nesnesi tablodan çekilecek veriyi barındıran SQL sorgusunu içermelidir.



Şekil 15: Veritabanındaki Bilgilerin Alınması

Visual Studio Programı ile oluşturduğumuz veritabanına bağlanmak için, Kod 4’deki gibi bağlantı cümleleri yazılmalıdır.

```
MySqlConnection baglanti = new MySqlConnection("Server=localhost;Port=3306;
Database=zembos;Uid=root; Pwd='';Encrypt=false;
AllowUserVariables=True; UseCompression=True;");
if (baglanti.State != System.Data.ConnectionState.Open)
        try{
baglanti.Open();
    string sorgu = "select * from sorular where Sid=" + soru1;
      MySqlCommand cmd = new MySqlCommand(sorgu, baglanti);
      MySqlDataReader dr = cmd.ExecuteReader();
      while (dr.Read())  {
```

```
            secSoru = dr["soru"].ToString();
}
baglanti.Close();
}catch (MySqlException ex){
MessageBox.Show("Hata Oluştu : {0}", ex.Message);
}
```

Kod 4: DataReader Nesnesi ile Veritabanından Verilerin Alınması

Kod 4'de ayrıca veritabanındaki verileri göstermek için DataReader nesnesi kullanılmıştır. DataReader nesnesi genellikle verileri okuma amaçlı kullanılır, oldukça kolay, performansı yüksek ve güncel çalışır. Eğer veriler üzerinde değişiklik yapılacaksa veya gruplama verilerimiz Kod 5'deki gibi DataSet nesnesi ile alınır. DataSet ve DataAdapter nesneleri veritabanındaki tablo yapısı şeklinde verileri tutabilen bir yapıdadır. DataSet ve DataAdapter nesneleri, tablodaki her bir kaydı (satır) çeker. Dizilerin kullanıldığı gibi, indis numaraları ya da alan adları ile kullanılabilir.

```
if (baglanti.State != System.Data.ConnectionState.Open){
try{
baglanti.Open();
MySqlDataAdapter da = new MySqlDataAdapter("select * from sorular", baglanti);
DataTable dt = new DataTable();
da.Fill(dt);
listBox1.DataSource = dt;
baglanti.Close();
}catch (MySqlException ex) {
MessageBox.Show("Hata 2 : " + ex.Message);
}
}
```

Kod 5: DataSet Nesnesi ile Verilerin Çekilmesi

Visual Studiodaki nesnelerden ListBox, ComboBox, dataGridView gibi nesnelerin "DataSource" özelliği vardır. DataSource özelliği DataReader veya DataAdapter nesnesi ile doldurulabilir. DataSource özelliğine yüklenen veriler, ListBox yada ComboBox nesnelerinin birer "item" 'ı gibi davranabilir ve seçme işlemlerinde kullanılabilir.

3.4 Zemberek DDİ Kütüphanesinin Projeye Eklenmesi

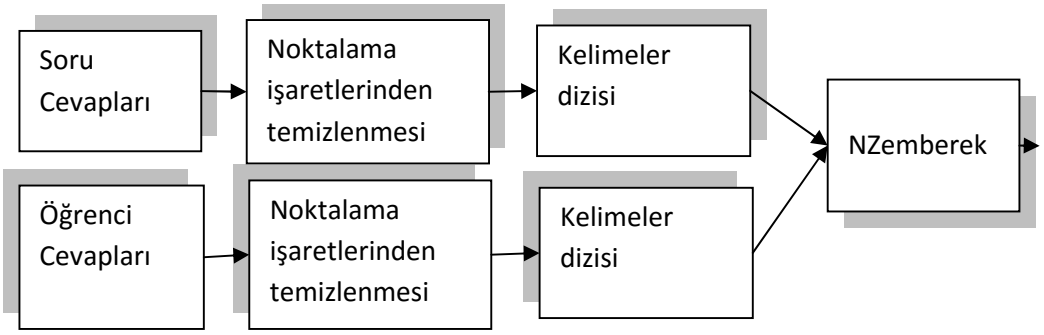
Bu çalışmada Zemberek projesi kelimeleri köklerine ayırma sürecinde kullanılmıştır. Zemberek Projesi Java ile geliştirilmiş olup hala yeni eklemeler ve yeni işlemlerden geçmektedir. Sistemin geliştirildiği platform C# olduğundan dolayı ve Zemberek projesinin diğer platformlara uygun hale getirilme süreci devam ettiğinden birçok zorlukla karşılaşılmıştır. Java platformu için geliştirilen Zemberek'in Visual Studio ortamı için uyarlanmış hali NZemberek'tir. NZemberek'i sistemimize dahil edebilmek için bir dll dosyası kullanılmıştır. "NZemberek.dll" adındaki bu dll dosyası, "MySQL.Data" klasörünün eklendiği gibi, "Add Referance" kısmından sisteme entegre edilmiştir. Kelimelerin köklerini bulmak için sadece NZEmberek Projesini eklemek yeterli değildir. Bununla beraber sisteme;

- Iesi.Collections.dll,
- Iesi.Collections.Generic.dll,
- log4net.dll,

kütüphaneleri de eklenmiştir. Ayrıca kelimeleri, kök yapılarını, yumuşama, kelime türlerinin (isim, sıfat, fiil vb.) bilgilerini tutan Zemberek ekibi tarafından geliştirilmiş, XML dosyalarını içeren "kaynaklar" klasörünün de projemizin çalışacağı ortamda bulunması gerekmektir.

Visual Studio'da proje geliştiren kullanıcılar, programın yazılması ve derlenmesi sırasında kullanacağı dosyalar ./Debug/bin klasörü altında toplanır. NZemberek Projesinin sayfasından indirdiğimiz dosyaların içerisinde bulunan "kaynaklar" klasörü, bizim çalışma dizinimizin içerisine kopyalanır. "kaynaklar" klasörünün içerisinde sözlük, ek bilgisi, kelime türleri, harfler gibi Türkçe dilinin yapısını gösteren kurallar bulunur.

3.5 Verileri Ayrıştırma



Şekil 16: Cevapların Ayrıştırılması

Kelime havuzu olarak nitelendirdiğimiz kısımda, karşılaştırma için kullanacağımız sorunun cevabını orijinal haliyle tutuyoruz. Burada tutulan veriler cümleler halindedir. Öğrencilerin yazım yanlışları, noktalama işaretleri ve kelimelerin ek almış halleri ölçme-değerlendirme sistemimize uygun değildir ve verilen cevaplarda birçok farklılık olmasına sebep olur. Şekil 16'daki gibi sorunun cevabı ve öğrencinin verdiği cevap önce kelimelere ayrılmalıdır. Kelimelere ayrılırken ilk önce yapılması gerekenlerden biriside noktalama işaretlerinden temizlenmesidir.

Cevap= "Osmanlı İmparatorluğuna Bursa, Edirne ve İstanbul'da başkentlik yapmıştır."

Öğrenci cevap=" 1-) Bursa,
2_)Konya,
3.İstanbul"

Şekil 17: Cevap ve Öğrenci Cevap Örneği

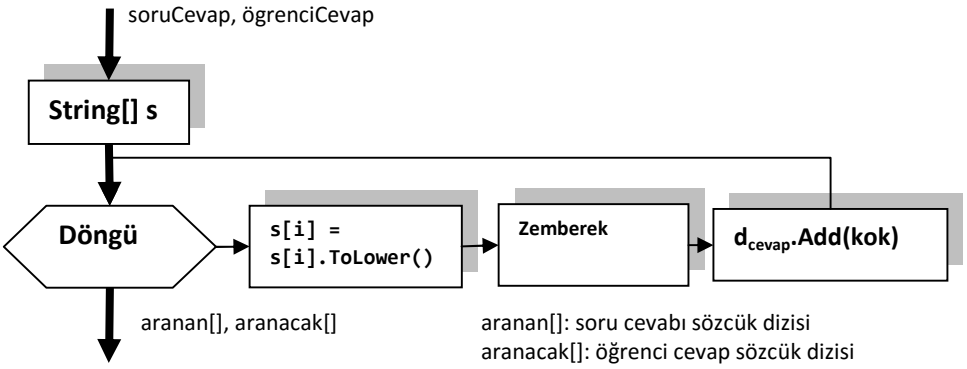
Şekil 17'de cevap ve öğrenci cevap cümleleri gösterilmiştir. Bu cümleler Zemberek Projesiyle kelimelerin köklerini bulmamızı engelleyecek noktalama işaretlerini içermektedir. Şekil 17'deki cevap örneklerinde " , - .) _ ' 1 2 3 " işaretleri cevabı etkilememektir. Aynı zamanda bu işaretlerin kelimeye bitişik yazılması, kelimenin kökünün bulunmasına engeldir. Kelimelere ayrıştırma işleminde en önemli ayıraç, boşluktur (space karakteri). C# dilinde Split fonksiyonu bize cümleleri kelime dizisine dönüştürür. Ayrıştırma işleminde cevabın

içerebileceği her türlü noktalama işaretini (tab ve satırbaşı dahil) dikkate almalıyız. Bunun için Kod 6'daki gibi uygulanmalıdır.

```
cevap=" 1-) Bursa,
                    2_)Konya,
                    3.İstanbul"
String[] s=cevap.Split(new Char[] { ' ', ',', '.', ':', '\t', '(', ')', '-', '_', '+', '%', '*', 'é', '\'', '/',
'=', '"', '!', '\n', '\r', '?', '"', '[', ']', '{', '}', '\\', '&', '@', ';', '|' })
s={bursa,konya,istanbul}
```

Kod 6: Cümlenin Kelime Dizisine Dönüştürülmesi

Kod 6'da gösterildiği gibi, s kelime dizisi artık Zemberek tarafından kökleri bulunabilecek bir yapıdadır.



Şekil 18: Kelime Dizisinin Köklerinin Bulunması.

Şekil 18'de Kelime dizisinin Zemberek Kelime işlemciyle köklerinin bulunmasını, akış diyagramıyla göstermektedir. Kısaca kelime dizisindeki kelimeler tek tek alınır, küçük harfe dönüştürülür, kelime kökü bulunur ve "d_{cevap}" dizisine eklenir. Şekil 18'de "d_{cevap}" dizi primitif bir dizi yerine Collection sınıfının bir üyesi olan ArrayList dizi yapısına eklenmektedir. ArrayListler klasik dizilere göre daha esnek bir yapı sunar. ArrayListlerde dizinin eleman sayısını ve dizinin türünü önceden belirtmeye gerek yoktur, her türlü veri istenilen sayıda

eklenebilir. ArrayList dizi kullanılmadan önce “using System.Collections;” satırı kütüphanelere eklenmelidir. Öğrencinin verdiği cevapta aynı işlemlerden geçerek “$d_{ögrencicevap}$“ dizisine aktarılır.

Cümlenin kelimelere ayrılması işleminde dikkat etmemiz gereken bir diğer husus da kelimenin büyük harfle yazılmasıdır. Bu yüzden Kod 7’deki gibi kelime Zemberekle kökü bulunmadan önce “.ToLower()” metodu ile küçük harfe çevrilir.

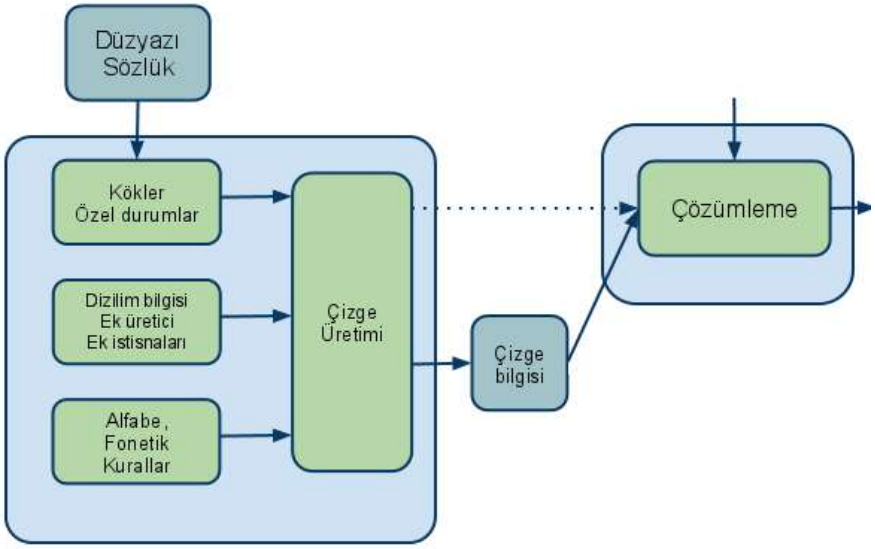
```
s[i] = s[i].ToLower();
```

Kod 7: Kelimelerin Küçük Harfe Dönüştürülmesi.

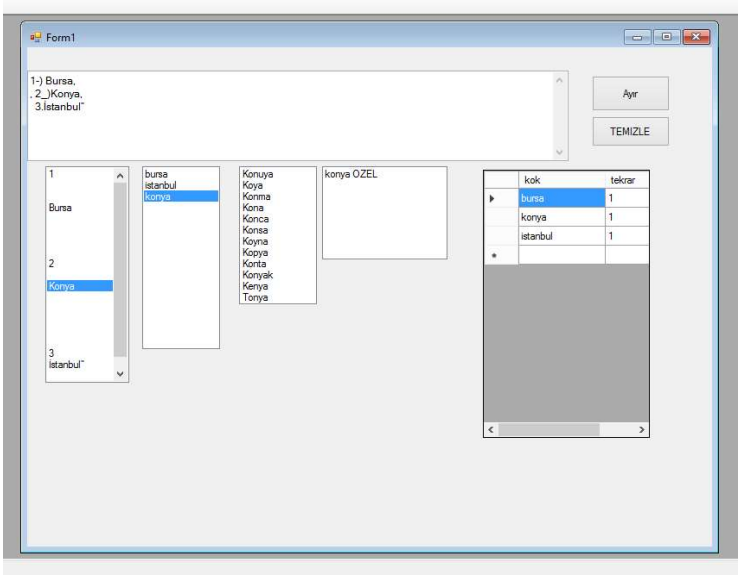
Zemberek DDİ Kütüphanesiyle, kelimelerin köklerinin bulunması için, önce nesnemizin oluşturulması gerekmektedir. Zemberek Projesinde “kelimeDenetle()” fonksiyonu kelimeni sözlükte olup olmadığının bilgisini verir. Kelime denetlenecekse önce kelime havuzunda olmalıdır, değilse Türkçe bir kelime değildir. “kokler.bin” dosyasında 1400.000 civarında kelime vardır. “kelimeCozumle” fonksiyonu kelimemizi kökünü verir, aynı zamanda kök kelimenin hangi tür kelime (fill, sıfat,isim vb.) olduğu bilgisini verir. Şekil 19’da Zemberek Kütüphanesinin çözümleme şeması özetle anlatılmaya çalışılmıştır.

```
Zemberek zemberek = new Zemberek(new TurkiyeTurkcesi());
…….
if (zemberek.kelimeDenetle(s[i].ToString())){
Kelime[] cozumler = zemberek.kelimeCozumle(s[i].ToString());
String kelimeKok = cozumler[0].kok().ToString();
Dizi.Add(kelimeKok[0]);
}
```

Kod 8:Zemberek DDİ Kütüphanesi ile Kelime Çözümleme



Şekil 19: Zemberek DDİ Kütüphanesinin İşleyiş Şeması



Şekil 20: Örnek Cevabın Kelimelere Ayrılması, Yazım Önerisi, Kelime Türünün Bulunması

Şekil 20'de hazırladığımız ara yüzle örnek olarak alınan bir cevabın kelimelere, köklere ve kelime türlerine ayrılması gösterilmiştir. Bu ara yüz projemize eklediğimiz Zemberek Kütüphanesini test etme imkânı sunmuştur. Burada eklediğimiz Zemberek Kelime işlemcisi ile daha çok kelimeyi, cümleyi, örnek cevapları test etme olanağımız vardır.

3.6 Verilerin Sayısal Yapıya Dönüştürülmesi

Zemberek Kütüphanesinin köklerine ayırdığı kelimeleri program içerisinde tutan veri yapısı ArrayListler ve DataTable nesneleridir. DataTable nesnesin kullanılmasının nedeni verileri iki boyutlu tutabilmesi, ToolBox'da sunulan birçok nesnenin DataSource özelliğini desteklemesi ve içersindeki veriler üzerinde LINQ çalıştırılabilmesidir. LINQ'in C#3.0 ile kullanımı yaygınlaşmaya başlamıştır. LINQ ile veritabanına ve T-SQL'a başvurmadan herhangi bir liste üzerinde sorgulama yapılabilir. Kod 9'da verileri tutmakta kullanılan DataTable nesnesi gösterilmiştir. Burada "aranacak" isminde bir DataTable oluşturulmuş ve"aranacak" DataTable nesnesine "kok" ve "tekrar" isminde 2 tane sütün eklenmiştir. "Rows.Add()" kodunu kullanılarak kelime kökü ve tekrar sayısı olarak da 1 eklenmiştir.

```
DataTable aranacak=new DataTable();
aranacak.Columns.Add("kok", typeof(string));
aranacak.Columns.Add("tekrar", typeof(int));
……..
aranacak.Rows.Add(new object[] { kelimeKok[0], 1 });
```

Kod 9: DataTable tanımlama ve Veri ekleme

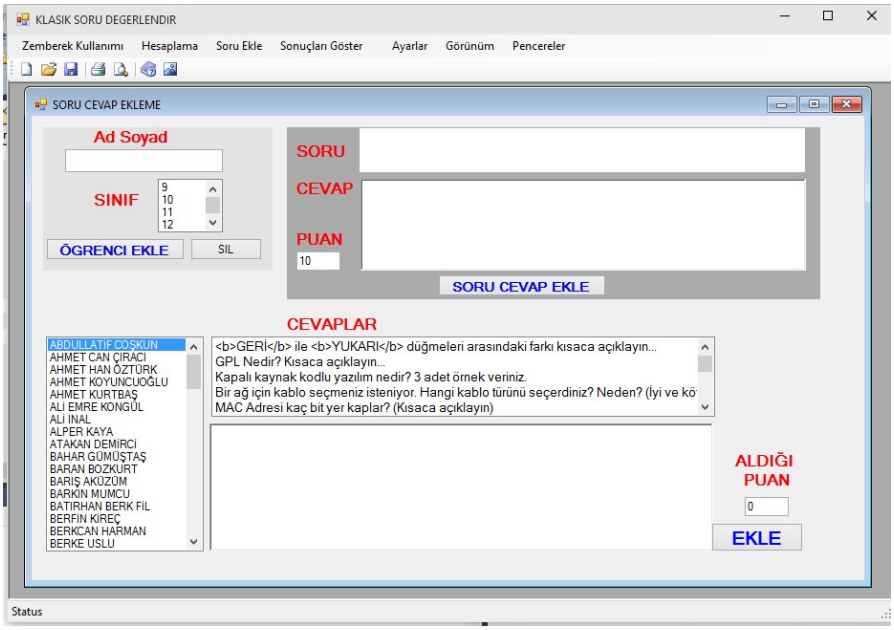
Kod 8 ve Kod 9'da oluşturulan, çözümleme ve ekleme yapıları programda 2 ayrı şekilde kullanılmıştır. Bunlardan birincisi aranılan cevabı içeren cümle veya cümleleri ayırmak için, ikincisi de öğrenci cevabı olarak değerlendirilecek cümleler için kullanılmıştır. Her iki DataTable nesnesindeki verileri göstermek için dataGridView nesnesi kullanılmıştır. "dataGridView1" nesnesinin DataSource'u olarak, "aranacak2" DataTable'ı seçilmiştir.

Kod 10'da aranacak DataTable üzerinde yapılan sorguyla aynı türden kelimeleri gruplandırarak kelime ve tekrar sayısı şeklinde, "aranacak2" DataTable'ı oluşturulup eklenmiştir.

```
var queryCevap = from rowC in aranan.AsEnumerable()
group rowC by rowC.Field<string>("kok") into grpC
select new
{
Id = grpC.Key,
sum = grpC.Sum(r => r.Field<int>("tekrar"))
};
foreach (var grpC in queryCevap)
{
aranan2.Rows.Add(new object[] { grpC.Id, grpC.sum });
dataGridView2.DataSource = aranan2;
}
```

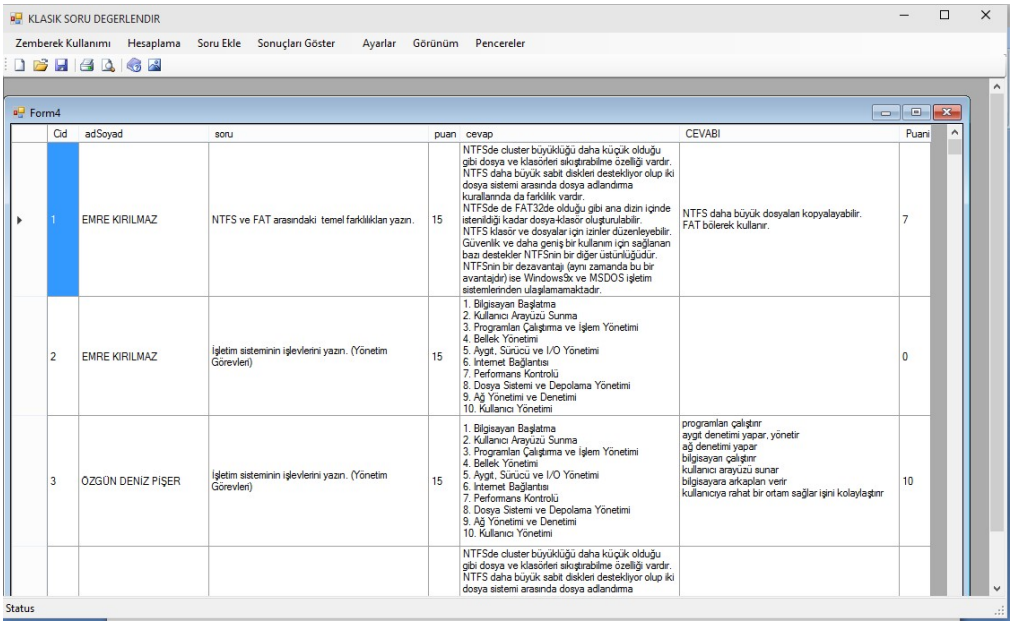
Kod 10: DataTable Verileri 'kok'e Göre LINQ ile Gruplandırıp, dataGridView Nesnesinde Gösterilmesi

Bu işlemlerden sonra artık veriler Vektör Uzayı Modeline göre bir doğru oluşturacak yapıya dönüştürülmüş olur.



Şekil 21: Soru ve Cevap Ekleme Sayfası

Programda soruların eklenmesi için Şekil 21'de gösterilen form oluşturulmuştur. Soru cevap ekleme kısmı birden fazla öğrenci için birden fazla sınav girilebilecek şekilde ayarlanmıştır. İlk önce öğrenci adı soyadı ve sınıfı eklenmektedir. Bir sınıfın cevaplarının değerlendirileceği düşünülerek sınav soru ve cevaplarının girilmesi için, "SORU CEVAP EKLE" kısmı hazırlanmıştır. "CEVAPLAR" kısmında, veritabanına eklenen öğrenciler ve sorular listbox'tan seçilip, öğrencilerin verdikleri cevabı ve aldığı puanı TextBox'a yazan ve veritabanına ekleyen bir ara yüz tasarlanmıştır.

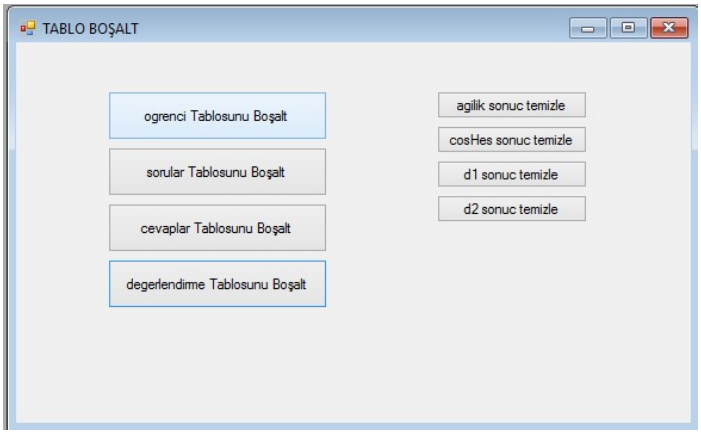


Şekil 22: Soru Cevapların Tümünün Gösterimi

Şekil 22'de sitemde kayıtlı bütün öğrenci ve soru ve cevaplarını görebileceğimiz ayrı bir ara yüz oluşturulmuştur. Bu ara yüz Ayarlar menüsü içerisindedir. Kod 11'de SQL kodu ve DataTable kullanarak tüm sorular ve cevaplar çekilmiş ve dataGridView nesnesinin DataSource'una yüklenmiştir.

```
…..
MySqlDataAdapter da = new MySqlDataAdapter("select C.Cid, O.adSoyad, S.soru, S.puan, S.cevap, C.cevap as CEVABI, C.Puani from ogrenci O,sorular S,cevaplar C where C.Oid=O.Oid and C.Sid=S.Sid", baglanti);
DataTable dt = new DataTable();
da.Fill(dt);
….
```

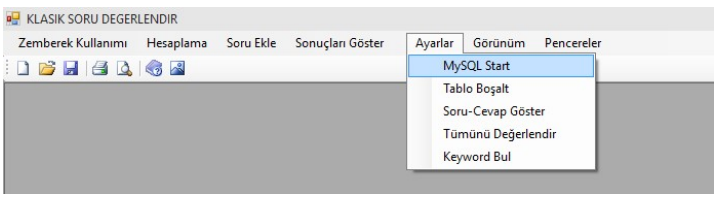
Kod 11: "öğrenci", "soru cevap" ve "öğrenci cevabının" Gösterilmesi



Şekil 23: Veritabanındaki Verilerin Boşaltılması

Oluşturulan veritabanında tablolardaki kayıtlara, deneme işlemleri sırasında gereksiz birçok veri eklenmişti. Deneme işlemlerinde eklenen verileri temizlemek için her seferinde "localhost"tan veritabanı yönetimine girilip veriler siliniyordu. Bu işlemi kolaylaştırmak için Şekil 23'de gösterilen form tasarlanmıştır.

Şekil 24'de gösterilen "MySQL Start" menü seçeneği arka planda, Kod 12'de gösterilen kodları çalıştırarak, veritabanını kullanılabilir hale getirir.

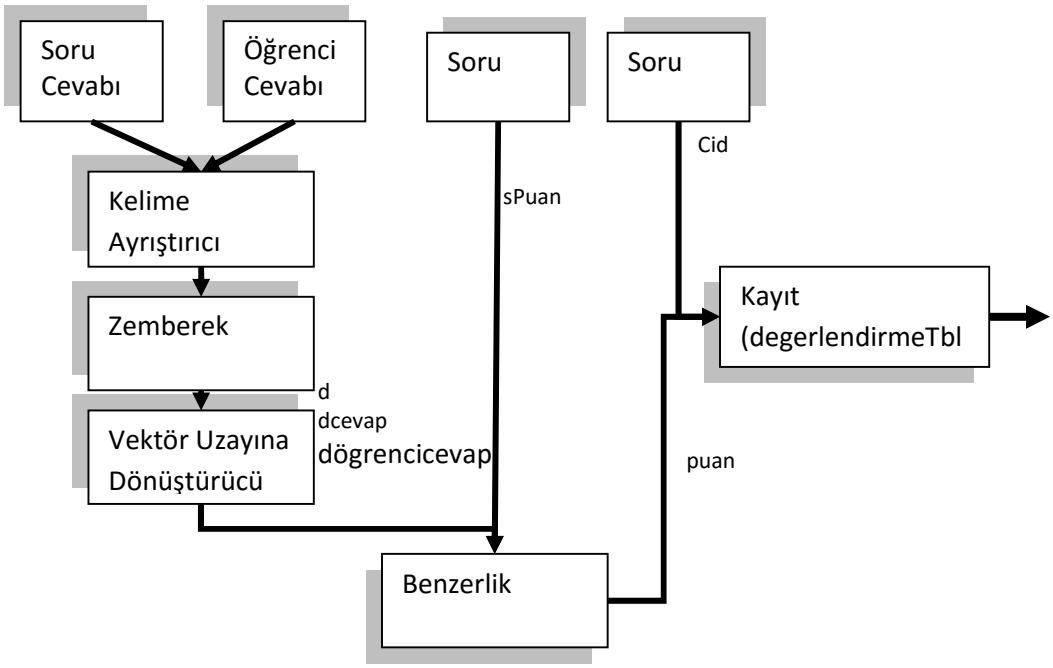


Şekil 24: MySQL Veritabanın Program Tarafından Başlatılması

```
Process[] p = Process.GetProcessesByName("mysqld");
if (p.Length > 0){
foreach (Process process in p){
MessageBox.Show("Zaten Çalışıyor");
}
}else{
try{
Process.Start("C:\\xampp\\mysql\\bin\\mysqld.exe");
MessageBox.Show("BAŞLADI");
}catch (Exception){ }
}
```

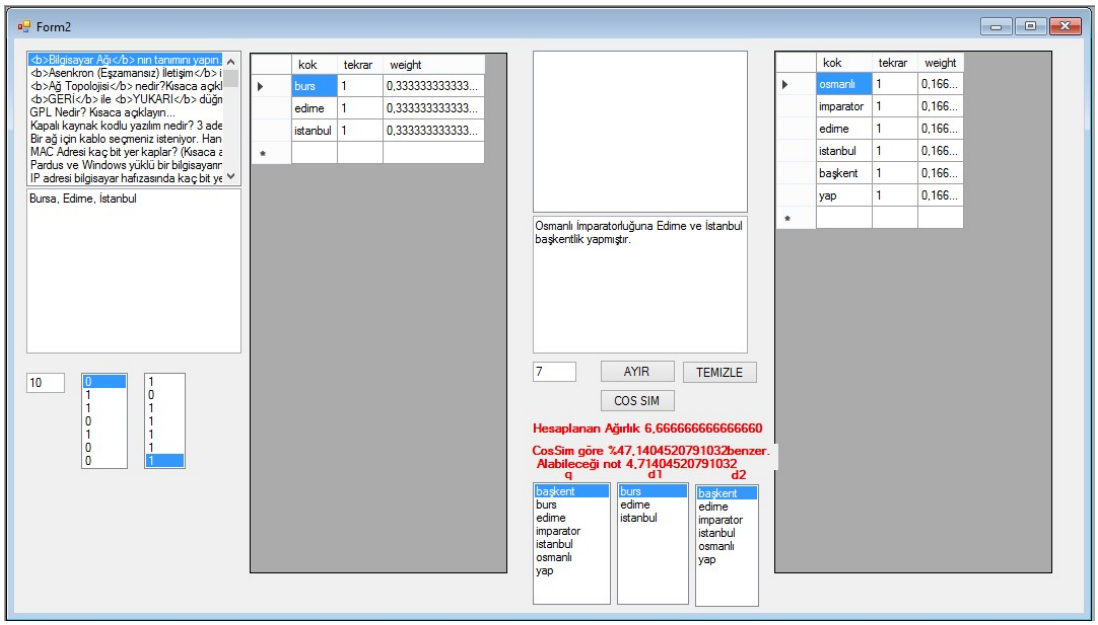
Kod 12: MySQL Veritabanın Program Tarafından Başlatılması

Şekil 25'de programda hesaplama işlemlerini yapılacağı kısmı akış diyagramı ile anlatılmaya çalışılmıştır. Hesaplamalarda baz alınan iki önemli değer vardır, bunlar "soru cevap" ve "öğrenci cevap" 'tır. Şekil 12'deki şemada hangi verilerin hangi alanlarda tutulacağı gösterilmiştir. Hesaplama işlemlerinde kullanılmak üzere için Şekil 26'da gösterilen form oluşturulmuştur.



Şekil 25: Soru Cevap ve Öğrenci Cevaplarının Ayrıştırılıp, Hesaplama Diyagramı

Şekil 25'de, soru cevabı ve öğrenci cevabı veri kaynağından çekilir. "Kelime Ayrıştırıcı" alınan cevaplardaki noktalama işaretlerini ve özel karakterleri temizler. "Zemberek DDİ Kütüphanesi" tarafından kelimelerin kökleri bulunur. "Vektör Uzayına Dönüştürücü", ayrıştırılan bu yarı yapısal verileri sayısal bir değere dönüştürür. "dcevap" soru cevabının vektörel ifadesini, "döğrencicevap" öğrencinin verdiği cevabın vektörel karşılığını ifade etmektedir. "d" ise vektör uzayı modelinde hesaplaması gösterilen "dcevap" ve "döğrencicevap" vektörlerinin birleşimidir. "Benzerlik" alanında "dcevap" ve "döğrencicevap" vektörlerinin hesaplanması yapılmaktadır. "Soru" alanından çekilen "sPuan" değeri ile sorunun puanının hesaplamaya eklenmesini ifade etmektedir. "Soru" alanından hangi sorunun hesaplaması yapıldığını gösteren "Cid" alanı alınarak hesaplanan puan "değerlendirme" tablosuna eklenir.



Şekil 26: Hesaplama Formu

Şekil 26'daki form ekranında Listbox1'da sorular gösterilmektedir. Seçilen sorunun cevabı altındaki Textbox1 nesnesinde gösterilmiştir. Listbox1'den seçtiğimiz soruya, öğrencilerin verdikleri cevaplar sağındaki Listbox2'de gösterilmektedir. Listbox2den seçilen cevap Textbox2'ye aktarılır. Cevaplarla beraber soruların puanları da gösterilmektedir. Textbox1 ve Textbox2 kelime havuzlarını nitelemektedir. Textbox1 soru cevap, Textbox2de öğrenci cevap diye adlandırılmıştır.

"AYIR" butonuna basıldığında, soru cevap ve öğrenci cevapları önce noktalama işaretlerinden, boşluklardan arındırılır. Temizlenen bu veriler Zemberek Kelime işlemcisi tarafından kelime köklerine ayrılır. Ayrılan bu veriler "aranan" ve "aranacak" diye ayrı ayrı DataTable nesnesine eklenir. "aranan" ve "aranacak" DataTable nesneleri Kod 13'deki LINQ sorgusu ile aynı kelimeler gruplanır ve kelimelerin karşısına tekrar edilme sayıları eklenir. Buradaki amaç, soru cevap ve öğrenci cevaplarına iki doğru benzetmesi yapılıp, bu doğruların birleşimi olan "q" doğrusuna göre aralarındaki açıyı bularak benzerliklerinin hesaplanmasıdır.

```
var queryCevap = from rowC in aranan.AsEnumerable()
  group rowC by rowC.Field<string>("kok") into grpC
    select new {
      Id = grpC.Key,
```

```
    sum = grpC.Sum(r => r.Field<int>("tekrar"))
};
object toplam= aranan.Compute("SUM(tekrar)","");
foreach (var grpC in queryCevap) {
   double sayi1 = grpC.sum;
   double sayi2 = Convert.ToInt32(toplam);
   aranan2.Rows.Add(new object[] { grpC.Id, grpC.sum, toplam });
}
```

Kod 13: Kelime Köklerine Ayrılmış Kelimelerin LINQ ile Gruplandırılması.

Kod 13'deki gruplandırma işlemi soru cevap ve öğrenci cevap'lara ayrı ayrı uygulanır. Sonuç olarak vektör uzayına dönüştürebilecek iki adet veri kümesi oluşur. "aranacak" olarak isimlendirilen DataTable aslında "$d_{öğrencicevap}$" doğrusu, "aranan" olarak isimlendirilen DataTable ise "d_{cevap}" doğrusudur. Şekil 27'de örnek bir "soru cevap" ve "öğrenci cevapları" üzerinde vektör oluşturma süreci görülmektedir. Burada, q doğrusu d_{cevap} ve $d_{öğrencicevap}$ doğrularının birleşimidir. "q" doğrusunda birleştirilen kelimeler ayrı ayrı d_{cevap} ve $d_{öğrencicevap}$ doğruları ile karşılaştırılır, aranılan kelimeyi içeriyorsa "1" değerini değilse "0" değerini alır.

	Cevaplar
dcevap	Bursa, Edirne, İstanbul
döğrencicevap	Osmanlı İmparatorluğuna Edirne ve İstanbul başkentlik yapmıştır.

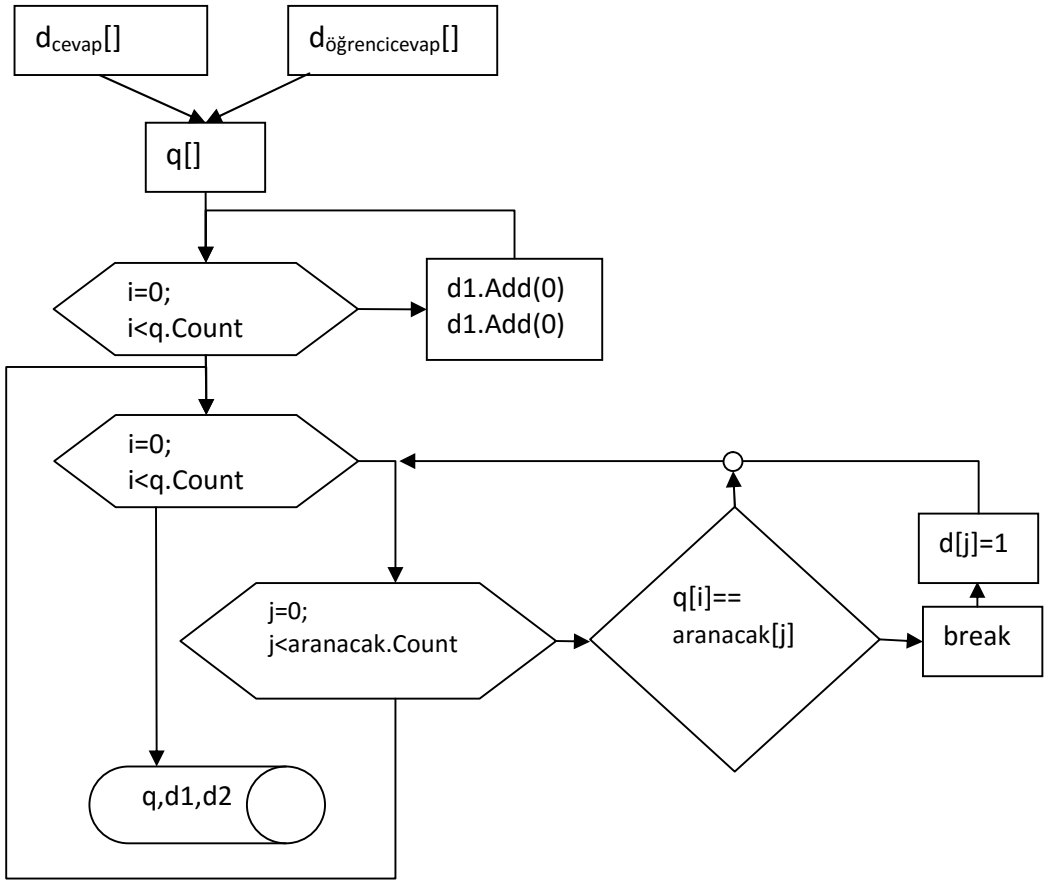
d_{cevap} = { bursa, istanbul, edirne }
$d_{öğrencicevap}$={ başkent, edirne, imparator, istanbul, osmanlı, yap }
q= d_{cevap} U $d_{öğrencicevap}$ ={ başkent, bursa, edirne, imparator, istanbul, osmanlı, yap }

q	başkent	bursa	edirne	imparator	istanbul	osmanlı	yap
$d_{cevap=}$d1	0	1	1	0	1	0	0
$d_{öğrencicevap=}$d2	1	0	1	1	1	1	1

Şekil 27: Soru Cevap ve Öğrenci Cevap Verileri Üzerinden Vektör Oluşturulması

Şekil 28'de oluşturulan d1 ve d2 doğrularının elemanlarına Kod 14'deki gibi 0 değeri atanır. Kod 14'deki değerler atanmazsa eleman sayısının farklı olmasından dolayı yanlış hesaplama yapar ve program hata verir. Anlatılan bu bilgileri akış diyagramı şeklinde ifade etmek için Şekil 28'deki diyagram oluşturulmuştur.

Şekil 28'de gösterilen diyagramda "d_{cevap}" ve "$d_{öğrencicevap}$" dizilerinden d1 ve d2 doğrularının elde edilmesi gösterilmiştir. Kod 14 ve Kod 15'de programlama dilinde yazılan kodları, akış diyagramıyla ifade etmektedir.



Şekil 28: "d_{cevap}" ve "$d_{öğrencicevap}$" Dizilerinden q, d1, d2 Vektörlerinin Oluşturulması

```
        for (int i = 0; i < q.Count; i++)
{
            d1.Add(0);
            d2.Add(0);
}
```

Kod 14: “q” Doğrusunun Elemanı Kadar “d1” ve “d2” Vektörünü Oluşturma ve Değer Atama

```
for (int i = 0; i < q.Count; i++){
        for (int j = 0; j <aranan.Count; j++){
        if (q[i].ToString() == aranacak[j].ToString()){
        d1[i] =  1;
                break;
        }
        }
}
```

Kod 15: d_{cevap} ve q Verilerinden d1 Doğrusunun Oluşturulması

Karşılaştırılacak olan soru cevap ve öğrenci cevap verileri, yarı yapısal bir halden sayısal bir veri dönüştürülmüştür. Sayısal olarak ifade edilebilen iki doğruya matematiksel işlemleri uygulayabilir. İki verinin birbirine ne kadar benzediğinin analizi, “Kosinüs Benzerliği Teoremini” uygulanarak yapılabilir.

Formül 5’deanlatılan kosinüs benzerliği formülünü Kod16’daki gibi uyguladık. “sonuc” değişkeninin değeri bizim benzerlik oranımızı verecektir. “sonuc” değişkeninin değeri, herhangi bir benzerlik yoksa IsNaN(null) değer alacağı için boş değer döndürmesini engellemek amacıyla “0”’a eşitlenmiştir. Bu hesaplamada d_{cevap} ve $d_{öğrencicevap}$ doğruları arasında benzerlik varsa 0 ile 1 arasında bir değer hesaplayacaktır. Hesaplanan bu değer 0,56 gibi bir değerse d_{cevap} ve $d_{öğrencicevap}$ doğruları %56 oranında benzer denilebilir.

```
   double sim = 0;
    int N = 0;
    N = ((d2.Count < d1.Count) ? d2.Count : d1.Count);
    double dot = 0;
    double mag1 = 0;
double mag2 = 0;
for (int n = 0; n < N; n++){
dot += Convert.ToDouble(d1[n]) * Convert.ToDouble(d2[n]);
     mag1 += Math.Pow(Convert.ToDouble(d1[n]), 2);
     mag2 += Math.Pow(Convert.ToDouble(d2[n]), 2);
}
Double sonuc= dot / (Math.Sqrt(mag1) * Math.Sqrt(mag2));
if (double.IsNaN(sonuc)){
sonuc = 0;
}
```

Kod 16: Kosinüs Benzerliğinin Uygulanması.

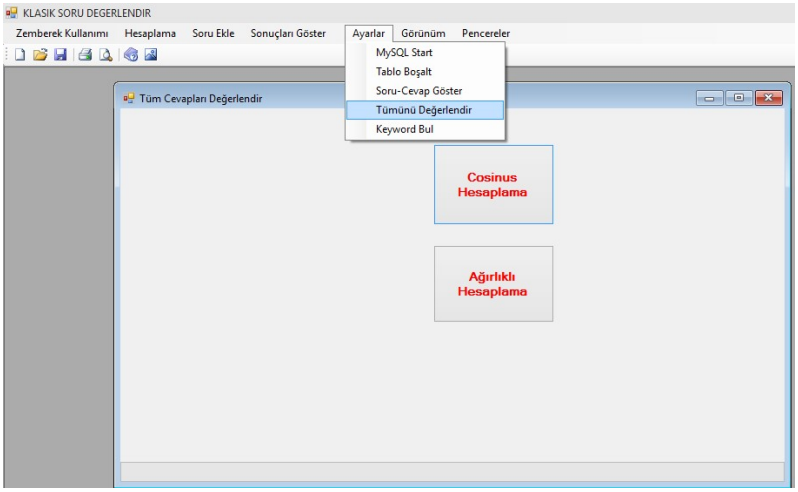
3.6 Ağırlıklı Hesaplama

Şekil 26'da gösterilen hesaplama formunda, uygulanan diğer bir yöntemde ağırlıklı hesaplama yöntemidir. Bu yöntem, Formül 6'da gösterildiği gibi kullanılmıştır. Bu formülde sorunun puanı, cevapta geçen kelime sayısına bölünür ve tekrar sayısı ile çarpılır. Öğrencinin verdiği cevapta bu kelimeler arandı, bulunan her kelimenin puanları toplanarak alabileceği not hesaplanmaya çalışıldı.

$$ti = \frac{sPuan}{\sum sorucevap} \quad \longrightarrow \quad \text{cevapPuan} = \sum_{1}^{öğrencicevap.count} ti$$

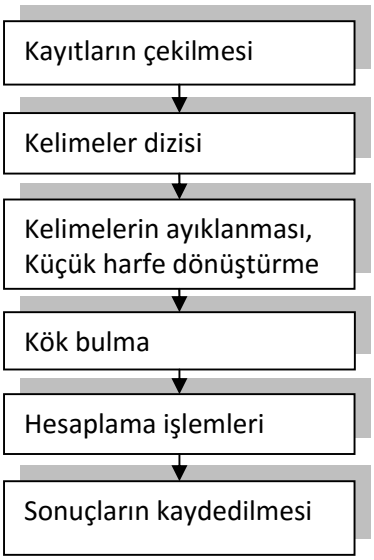
Formül 6: Ağırlıklı Yöntemle Hesaplanması.

Şekli 29'da gösterilen değerlendirme formu ağırlıklı yöntem ve kosinüs hesaplama yöntemiyle bulduğu sonuçları, "degerlendirme" tablosunda "Cid" alanında soru puanı, verilen puan ve hesaplanan puan olarak otomatik olarak kaydeder.



Şekil 29: Toplu Değerlendirme Ekranı

Şekil 29'da "Ayarlar" menüsünden "Tümünü Değerlendir" seçilirse, veritabanımızda kayıtlı tüm öğrenci cevaplarını değerlendirir ve "degerlendirme" tablosuna ekler. Bu ekleme sırasında işlemlerin ilerleyişi, sayfanın alt kısmında progressBar nesnesi ile kullanıcılara gösterilir. Veritabanımıza eklenmiş tüm kayıtların değerlendirilmesi; Şekil 30'daki diyagramda gösterildiği gibi veritabanından kayıtlar çekilir, kelimelere ayrıştırılarak kelime dizisine aktarılır, ayrıştırılan bu kelimeler noktalama işaretlerinden temizlenir ve küçük harfe dönüştürülür, kelimeler köklerine ayrılır, hesaplama işlemi uygulanır, sonuçlar tabloya kaydedilir.

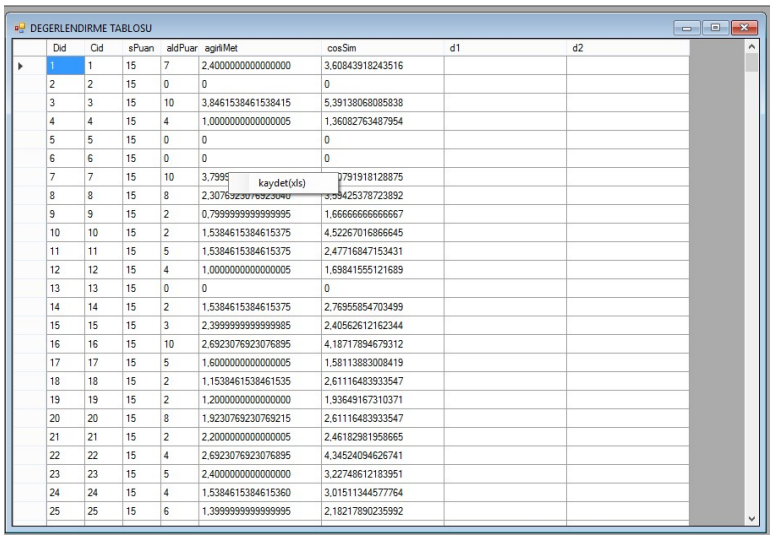


Şekil 30: Toplu Değerlendirme İşleyiş Basamakları

Sonuçları kaydetmek için "degerlendirme" tablosu kullanıldı. "degerlendirme" tablosunu görmek için Şekil 31'de gösterilen, toplu değerlendirme formu oluşturuldu. Şekil 31'deki formda bir adet dataGridView nesnesi yer almaktadır. Veritabanındaki "degerlendirme" tablosundan alınan veriler, DataTable nesnesine çekilir ve dataGridView nesnesinin DataSource'una yüklenir. dataGridView nesnesinde gösterilen verileri test etmek için bu veriler bir Excel dosyasına yazılabilir. Bu Excel dosyasına yazma işlemi için "saveFileDialog" nesnesi ile Kod 17'deki kod kullanılarak yapılmıştır. Daha sonra bu Excel belgesine kaydedilen veriler SPSS[5] programında kullanılacaktır.

"degerlendirme" tablosunda kayıtları, "Did" tabloya eklenen her kayıt için bir numara verilir ve otomatik olarak artan bir tamsayı olarak ayarlanmıştır. "Cid" öğrencilerin verdiği her cevabın tutulduğu kaydın alandır. Bu tabloda "sPuan" alanında sorunun puanı eklenmektedir. "aldPuan" alanı öğrencini cevabının öğretmen tarafından değerlendirilen, sorudan aldığı puanı göstermektedir. Bu çalışmanın hesaplama sonuçları "agirilMet" ve "cosSim" alanlarında tutulmaktadır.

[5] IBM Firmasının SPSS Statistics Programı(versiyon19)



Did	Cid	sPuan	aldPuar	agirliMet	cosSim	d1	d2
1	1	15	7	2,4000000000000000	3,60843918243516		
2	2	15	0	0	0		
3	3	15	10	3,8461538461538415	5,39138068085838		
4	4	15	4	1,0000000000000005	1,36082763487954		
5	5	15	0	0	0		
6	6	15	0	0	0		
7	7	15	10	3,7995	0791918128875		
8	8	15	8	2,3076923076923040	3,59425378723892		
9	9	15	2	0,7999999999999995	1,66666666666667		
10	10	15	2	1,5384615384615375	4,52267016866545		
11	11	15	5	1,5384615384615375	2,47716847153431		
12	12	15	4	1,0000000000000005	1,69841555121689		
13	13	15	0	0	0		
14	14	15	2	1,5384615384615375	2,76955854703499		
15	15	15	3	2,3999999999999985	2,40562612162344		
16	16	15	10	2,6923076923076895	4,18717894679312		
17	17	15	5	1,6000000000000005	1,58113883008419		
18	18	15	2	1,1538461538461535	2,61116483933547		
19	19	15	2	1,2000000000000000	1,93649167310371		
20	20	15	8	1,9230769230769215	2,61116483933547		
21	21	15	2	2,2000000000000005	2,46182981958665		
22	22	15	4	2,6923076923076895	4,34524094626741		
23	23	15	5	2,4000000000000000	3,22748612183951		
24	24	15	4	1,5384615384615360	3,01511344577764		
25	25	15	6	1,3999999999999995	2,18217890235992		

Şekil 31: Toplu Değerlendirme Sonuçlarının Gösterilmesi

```
saveFileDialog1.ShowDialog();
string cikti = "";
string baslik = "";
for (int j = 0; j < dataGridView1.Columns.Count; j++){
baslik = baslik.ToString() + Convert.ToString(dataGridView1.Columns[j].HeaderText) +
"\t";
cikti += baslik + "\r\n";
for (int i = 0; i < dataGridView1.RowCount - 1; i++){
string stLine = "";
for (int j = 0; j < dataGridView1.Rows[i].Cells.Count; j++)
stLine = stLine.ToString() + Convert.ToString(dataGridView1.Rows[i].Cells[j].Value) +
"\t";
cikti += stLine + "\r\n";
}
Encoding utf16 = Encoding.GetEncoding(1254);
byte[] uret = utf16.GetBytes(cikti);
FileStream fs = new FileStream(saveFileDialog1.FileName, FileMode.Create);
BinaryWriter bw = new BinaryWriter(fs);
bw.Write(uret, 0, uret.Length);
```

```
bw.Flush();
bw.Close();
fs.Close();
```

Kod 17: DataGridView Nesnesinden Excel’e Veri Aktarımı

4 DEĞERLENDİRME

4.1 Veri Seti

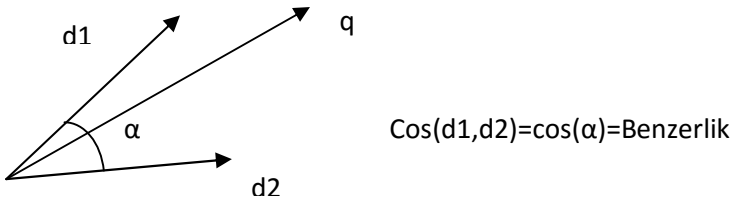
Bu tez kapsamında geliştirilen sistem ile klasik soruları değerlendirmek için Mesleki Teknik Anadolu Lisesinin 10.sınıf, Bilişim Teknolojilerinin Temelleri Dersinin klasik soruları ve cevapları kullanılmıştır. Bu veriler içerisinden değerlendirmek üzere toplam 1555 öğrenci cevabı seçilmiştir. Şekil 32'de örnek soru, cevap, soru puan ve öğrenci cevap kayıtları gösterilmektedir.

Bölüm2'de anlatıldığı gibi cevap önce kelimelere ayrılıp Zemberek DDİ Kütüphanesi ile kökleri bulunmuştur. Sorunun cevabı "d_{cevap}" adında bir dizide tutulmaktadır. Öğrencinin bu soruya verdiği cevap da "$d_{öğrencicevap}$" dizisinde tutulmaktadır. "d_{cevap}" dizisi d1 doğrusuna ve "$d_{öğrencicevap}$" dizisi de d2 doğrusuna, Vektör Uzayı Modeline göre sayısal bir yapıya çevrilerek, yerleştirilmiştir.

Cid	adSoyad	soru	puan	cevap	CEVABI	Puani
16	ÖMER KAYA	İşletim sisteminin işlevlerini yazın. (Yönetim Görevleri)	15	1. Bilgisayarı Başlatma 2. Kullanıcı Arayüzü Sunma 3. Programları Çalıştırma ve İşlem Yönetimi 4. Bellek Yönetimi 5. Aygıt, Sürücü ve I/O Yönetimi 6. İnternet Bağlantısı 7. Performans Kontrolü 8. Dosya Sistemi ve Depolama Yönetimi 9. Ağ Yönetimi ve Denetimi 10. Kullanıcı Yönetimi	Bilgisayarı başlatır. Kullanıcılara arayüzü sunar. Kullanıcıya donanım birimlerini tanıtır. İnternete bağlanmamızı sağlar. Tıkladığımızda dosyalarımızı açar.	10
17	ÖMER FARUK ALTUNTAŞ	NTFS ve FAT arasındaki temel farklılıkları yazın.	15	NTFSde cluster büyüklüğü daha küçük olduğu gibi dosya ve klasörleri sıkıştırabilme özelliği vardır. NTFS daha büyük sabit diskleri destekliyor olup iki dosya sistemi arasında dosya adlandırma kurallarında da farklılık vardır. NTFSde de FAT32de olduğu gibi ana dizin içinde istenildiği kadar dosya-klasör oluşturulabilir. NTFS klasör ve dosyalar için izinler düzenleyebilir. Güvenlik ve daha geniş bir kullanım için sağlanan bazı destekler NTFSnin bir diğer üstünlüğüdür. NTFSnin bir dezavantajı (aynı zamanda bu bir avantajdır) ise Windows9x ve MSDOS işletim sistemlerinden ulaşılamamaktadır.	NTFS'nin dosya büyüklüğü FAT'dan fazladır. FAT NTFS'ye göre bilgileri en ufak parçalara bölerek tutar.	5
18	ÖMER FARUK ALTUNTAŞ	İşletim sisteminin işlevlerini yazın. (Yönetim Görevleri)	15	1. Bilgisayarı Başlatma 2. Kullanıcı Arayüzü Sunma 3. Programları Çalıştırma ve İşlem Yönetimi 4. Bellek Yönetimi 5. Aygıt, Sürücü ve I/O Yönetimi 6. İnternet Bağlantısı 7. Performans Kontrolü 8. Dosya Sistemi ve Depolama Yönetimi 9. Ağ Yönetimi ve Denetimi 10. Kullanıcı Yönetimi	Bilgisayardaki donanımların kullanımı sağlar.	2
19	MUHAMMED MAZLUM ORHAN	NTFS ve FAT arasındaki temel farklılıkları yazın.	15	NTFSde cluster büyüklüğü daha küçük olduğu gibi dosya ve klasörleri sıkıştırabilme özelliği vardır. NTFS daha büyük sabit diskleri destekliyor olup iki dosya sistemi arasında dosya adlandırma kurallarında da farklılık vardır. NTFSde de FAT32de olduğu gibi ana dizin içinde istenildiği kadar dosya-klasör oluşturulabilir.	NTFS ,FAT'a gore daha cok depolamma alınına izin verir	2

Şekil 32: Soru, Cevap, Soru Puan ve Öğrenci Cevap Kayıtlarının Görünümü

Değerlendirme hesaplamalarında temel alınacak yöntem, Şekil 33'de gösterildiği gibi iki doğrunun arasındaki açının kosinüsü benzerlik oranını vermesidir.



Şekil 33: Kosinüs Benzerlik Teoremi

Şekil 34'de sorunun cevabı ile öğrencinin verdiği cevap hesaplanmasının sonuçları gösterilmektedir. Bu hesaplamada iki ayrı sonuç bulunmuştur. Bu tabloda gösterilen "Cid" cevap Id bilgisini, "Soru Puanı" alanı, sorunun kaç puan üzerinden değerlendirileceğini, "Aldığı Puan" alanı, öğretmen bu cevaba kaç puan vermiş, "agirliMet" alanı, ağırlıklı yöntemle hesaplanan puanı, "cosSim" alanı da kosinüs benzerliğine göre hesaplamayı gösterir.

Cid	Soru Puanı	Aldığı Puan	agirliMet	cosSim
…	…	…..	…….	…….
15	15	3	2,40	2,40
16	15	10	2,69	4,18
17	15	5	1,60	1,58
18	15	2	1,15	2,61
19	15	2	1,20	1,93
20	15	8	1,923	2,61
21	15	2	2,20	2,46
22	15	4	2,692	4,34
23	15	5	2,40	3,22
24	15	4	1,53	3,01
25	15	6	1,40	2,18
26	15	8	2,69	4,04

27	15	8	5,00	8,37
28	15	6	2,00	2,22
…	…	….	….	……

Şekil 34: Sorunun Cevabı ile Öğrenci Cevabının Benzerlik Hesabıyla Bulunan Sonuçları

Şekil 34'de gösterildiği gibi, Cid'si 15 olan cevap örnek olarak alınırsa, 15 puanlık bir soruya öğrenci cevabının hesaplanması sonucunda sistem bu soruya 2,405 puan vermiştir. Aynı soru öğretmen tarafından değerlendirildiğinde 3 puan verildiği görülmektedir. Ayrıca öğrencinin cevap vermediği sorularda sistem 0 puan vermiştir.

Cid'si 20 ve 28'nolu sorularda, ilk bakışta sistemin hesapladığı notlar ile öğretmenin verdiği notlar arasında gözle görünür bir yakınlık olmadığı görülmektedir. Ancak unutulmamalıdır ki eğitimde her değerlendirme kendi içerisinde bir tutarlılık göstermelidir. Yani bir öğretmenin aynı sınıfın sınav sonuçlarını farklı bir zamanda değerlendirmesi durumunda sonuçlarında farklılıklar görülebilir. Bundan dolayı değerlendirme sisteminde "çan eğrisi" yönteminin kullanılması önerilir.

Bu çalışmada sistemin bulduğu sonuçlar ile öğretmenin verdiği sonuçların bilimsel olarak analiz edilmesi planlanmış ve veri setleri arasındaki benzerlik Pearson's Korelasyon katsayısı analizi ile test edilmiştir.

Pearson's Korelasyon katsayısı analizi, Formül 8'de gösterildiği gibi, iki sürekli veri setinin doğrusal ilişkisinin derecesinin ölçümünde kullanılır. Bu iki veri seti arasında "Bilimsel olarak anlamlı bir ilişki var mıdır?" sorusunun cevabını arar. İki değişkenin kovaryansının yine bu değişkenlerin standart sapmalarının çarpımına bölünmesiyle elde edilir.

$$P(x,y) = \frac{Cov(x,y)}{\sigma x * \sigma y}$$

$$P\left(d_{cevap}, d_{\text{öğ}rencicevap}\right) = \frac{Cov\left(d_{cevap}, d_{\text{öğ}rencicevap}\right)}{\sigma d_{cevap} * \sigma d_{\text{öğ}rencicevap}}$$

Formül 7: Pearson's Korelasyon Katsayısı Hesaplanması.

Bu sayede öğrencinin, öğretmenin verdiği not ile sistemin hesapladığı notun aralarındaki farkı değil de, bu notların ortalamalarından ne kadar standart sapma uzaklıkta olduklarının farkı kıyaslanmaktadır.

Bu analiz için IBM Firmasının SPSS Statistics [6] Programı (versiyon19) kullanılmıştır. Programımızın "Ayarlar" menüsü içersindeki "Toplu Değerlendir" seçeneği yardımıyla elde edilen sonuçlar(Şekil 34), SPSS programına aktarılıp, Pearson's Korelasyon katsayısı analizi uygulanmıştır.

4.2 Analiz Sonuçları

Tablo 1:Sonuçların Ortalaması ve Standart Sapma Değerleri

	Ortalama	Standart Sapma	N
aldgPuan	3,27	3,552	1555
cosSim	2,17932193	3,116730207	1555

Tablo 1'de ilk yapılan analizin veri setlerinin açıklayıcı bilgileri sunulmuştur. Burada "aldgPuan" değişkeni öğretmenin değerlendirmesi sonucu verdiği verilen notları gösterir. Bu değişkenlerin veri sayısı 1555'dir (N sütunu), ortalama değeri 3.27 ve standart sapması 3,552'dir. "cosSim" değişkeninde sistemin hesaplama sonucu bulduğu notu gösterir. Bu değişkenlerin veri sayısı da 1555'dir (N sütunu), ortalama değeri 2,179 ve standart sapması 3,116'dir.

[6] http://www-03.ibm.com/software/products/tr/spss-stats-standard

2: Pearson Korelasyon Katsayısı Analizi

		aldgPuan	cosSim
aldgPuan	Pearson Correlation	1	,526**
	Sig. (2-tailed)		,000
	N	1555	1555
cosSim	Pearson Correlation	,526**	1
	Sig. (2-tailed)	,000	
	N	1555	1555

**. Correlation is significant at the 0.01 level (2-tailed).

Tablo 2'de Pearson's Korelasyon katsayısı analizi sonuçları görülmektedir. Sonuçlarda ilk olarak "aldgPuan" ile "cosSim" değişkenlerinin arasındaki korelasyonun bilimsel geçerlilik (significance) değerinin 0,0000 olarak bulunduğu görülmektedir. Bu iki değişken arasında bilimsel olarak yüksek Güvenilirlik derecesinde (Confidence Level) bir gerçek korelasyon olduğunu göstermektedir.

Yapılan korelasyon analizinde bilimsel gerçekliği sağladıktan sonra iki değişken arasındaki düzeyi Tablo 3'deki Pearson's Korelasyon katsayısı lineer ilişki gücü verileri ile kıyaslanabilir.

Tablo 3: Pearson Korelasyon Katsayısı Lineer İlişki Gücü Seviyeleri[47]

0.8 üzeri	Cok güçlü(Very strong)
0.6 - 0.8 arası	Güçlü (Moderately strong)
0.3 -0.5 arası	Uygun (Fair)
0.3'den az	Zayıf (Poor)

Tablo 3'deki verilerin referansına dayanarak iki değişken arasındaki 0,526 korelasyon katsayısı değeri (r) ile "uygun (fair)" düzeyde bir ilişki gücünü sağlamıştır.

Hesaplama sonucu bulunan notların ilişkisini bulmak için kullandığımız diğer bir test de ikili t-test ile ikili karşılaştırmadır (Paired-Samples t-test). SPSS programında daha önce Pearson's Korelasyon katsayısı testini uyguladığımız veri setini t-test'e tabi tutarak Tablo 4'deki sonuçlar elde edilmiştir. Bu analiz sonucunda Pearson's Korelasyon katsayısı analizi ile t-test sonuçlarının aynı olduğu görülmüştür.

Tablo 4: T-Test Sonuçlarının Gösterimi

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	aldgPuan	3,270	1555	3,552	,090
	cosSim	2,179	1555	3,116	,079

**Paired Samples Statistics

		N	Correlation	Sig.
Pair 1	aldgPuan & cosSim	1555	,526	,000

**Paired Samples Correlations

Buradaki ilişki gücünü artırmak için, tek başlarına bir anlam ifade etmeyen kelimeler (ve, ile, daha, gibi, veya, daha, vb) hesaplamaya alınmamıştır. Bu kelimelerin haricinde soru cevaplarının içerisindeki kelimeler de düzenlenmiştir. Yani aranan kelimeler tüm cevabı değil de sadece özellikle aranan kelimelere (keyword) indirgenmiştir. Bu şekilde sistemin başarısının arttığı görülmüştür. Bu teste tabi tutulan veri sayısı 632'e indirilmiştir. Bunun sebebi, tamamen boş veya cevap verilmeyen soruların korelasyon sayısını artırdığının görülmesi ve veri setinden bu verilerin çıkarılmasıdır. Yapılan diğer bir işlemde ise, programda kullanılan hesaplama işlemine uygun olmayan soruların analizden çıkarılmasıdır. Bu tür sorular çok fazla sayısal ifade içermesi, 1-2-3 gibi sorulara öğrencilerin bir-iki-üç gibi cevaplar vermesi, "… olması gereken nedenlerden 3 tanesi yazınız" benzeri soruların doğru olan 10 cevaptan 3 tanesini istemekte olması, fakat 3 tane doğru cevap yazılsa bile 7 tane cevabın eksik verilmesinden dolayı eksik puanlama yapılmasıyla da sorunun cevabında geçmeyen diğer 7 tanesinden herhangi birinin yazılmasının yanlış cevapmış gibi işlem yapılmasına neden olması gibi sorunlar içermektedir.

Tablo 5: Tekrar Düzenlenmiş Veri Seti Üzerinde Analiz Sonuçları.

	Mean	Std. Deviation	N
aldgPuan	4,310	4,369	632
cosSim	1,810	2,100	632

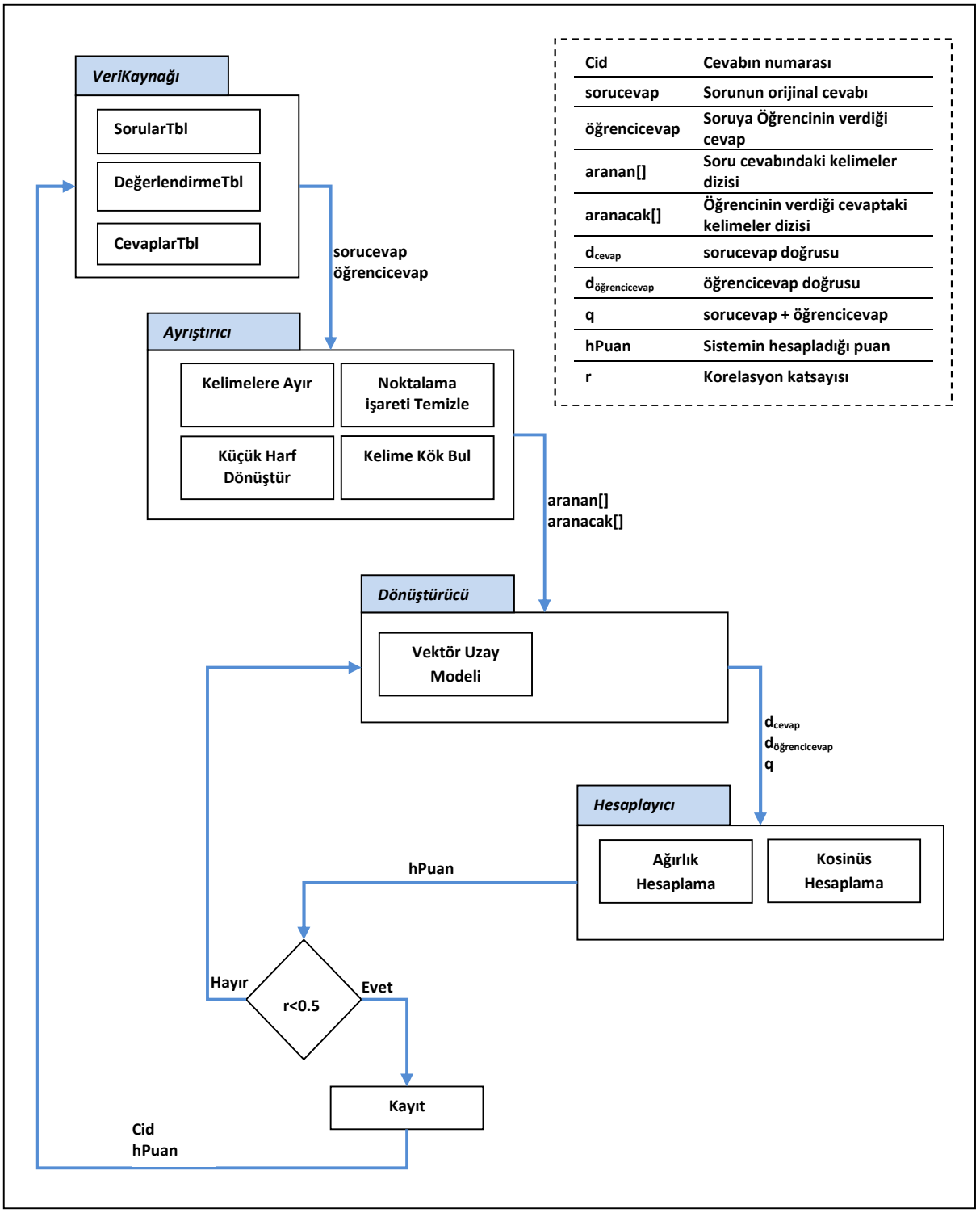
		aldgPuan	cosSim
aldgPuan	Pearson Correlation	1	,677**
	Sig. (2-tailed)		,000
	N	632	632
cosSim	Pearson Correlation	,677**	1
	Sig. (2-tailed)	,000	
	N	632	632

**. Correlation is significant at the 0.01 level (2-tailed).

Tablo 5'de düzenlemeden geçmiş verilerin sonuçlarını görmekteyiz. Burada "aldgPuan" değişkeni öğretmenin değerlendirmesi sonucu verdiği söylemiştik. Bu değişkenlerin veri sayısı 632'dir (N sütünü), ortalama değeri (Mean sütünü) 4,31 ve standart sapması (Std. Deviation) 4,369'dir. "cosSim" değişkeninde sistemin hesaplama sonucu bulduğu notu gösterir. Bu değişkenlerin veri sayısı da 632'dir (N sütünü), ortalama değeri (Mean sütünü) 1,810 ve standart sapması (Std. Deviation) 2.100'dir. Tablo 5'deki verilerin referansına dayanarak iki değişken arasındaki 0,677 korelasyon katsayısı değeri (r) ile "Güçlü (Moderately strong)" düzeyde bir ilişki olduğu gözlemlenmiştir.

Geliştirilen "Klasik Soru Değerlendirme Karar Destek Sisteminin" en doğru sonuçları veren (bilimsel olarak geçerli ve en yüksek korelasyon katsayısına sahip) algoritması Şekil 35'de verilmiştir.

Bu yapılan optimizasyonlar sonucunda geliştirilen “klasik soru değerlendirme karar destek sisteminin” en doğru sonuçları veren (bilimsel olarak geçerli ve en yüksek korelasyon katsayısına sahip) algoritması Şekil 35’de verilmiştir.



Şekil 35: Klasik Soru Değerlendirme Sistemi Algoritması

5. SONUÇ

"Klasik Soru Değerlendirme Karar Destek Sistemi" çalışmasında ADDIE (Analiz – Tasarım – Geliştirme – Uygulama - Değerlendirme) sistem geliştirme modeli uygulanmıştır. Bu tez çalışmasında tasarlanan klasik soru değerlendirme sisteminde sorunun doğru cevabı ile öğrencinin verdiği cevap karşılaştırıldı.

İlk olarak, çalışmanın analiz bölümünde veri ve metin madenciliği alanında literatürde bulunan çalışmalar analiz edilip değerlendirildi. Bu değerlendirme sonucunda geliştirilecek olan sistem için Vektör Uzay Modeli, Kosinüs Teoremi, Zemberek Kütüphanesi ve Pearson's Korelasyonu kullanılmasına karar verildi.

Çalışmanın ikinci bölümünde "Veri Madenciliği" ve "Metin Madenciliği" ile ilgili kavramlar, teoremlerin ışığında sistemin veri analizi kısmı tasarlandı.

Geliştirme aşamasında "Klasik Soru Değerlendirme Karar Destek Sistemi'nin" Visual Studio ortamında C# dili ile kodlanması yapıldı. Bu sistemde, öğrencinin verdiği cevap ve sorunun cevabı noktalama işaretleri ve özel sembollerden temizlendi. Ardından, Zemberek Kelime işlemcisi ile kelime kökleri bulundu ve dizilere aktarıldı.

Kelimelerin bulunduğu diziler üzerlerinde hesaplama yapılabilmesi için Vektör Uzayı Modeli kullanılarak sayısal bir yapıya dönüştürüldü.

Sorunun doğru cevabı ile öğrencinin verdiği cevabı karşılaştırmak için "Kosinüs Benzerliği Teoremi" ve Ağırlıklı Notlandırma" yöntemleri kullanıldı. Sistemin test edilebilmesi için örnek olarak alınan soru cevap ve öğrenci cevapları geliştirilen algoritma ile analiz edildi.

"Kosinüs Benzerliği Teoremi" ve "Ağırlıklı Notlandırma Yöntemi" ile karşılaştırılan metinlerden bulunan sonuçlar IBM Statistic firmasının SPSS programında "Pearson's Korelasyon" ve "T-test" analizleri ile test edildi. Her iki yöntemde de birbirine yakın sonuçlar hesaplandığı görüldü. İlk ölçme sonuçlarında öğreticinin değerlendirmesi ve sistem değerlendirmesi arasında Pearson's Korelasyon katsayısının0,532 olduğu gözlemlendi. İlk aşamada elde edilen test sonuçlarına göre oluşturulan algoritma üzerinde bazı optimizasyonlar yapıldı. Yapılan optimizasyonlardan sonra sistem yeniden değerlendirildi ve

öğretici değerlendirmeleri ve sistem değerlendirmeleri arasındaki Pearson's Korelasyon katsayısının 0,637 'ye çıktığı gözlemlenmiştir. Bu sonuçlar sistemin not değerlendirmeleri ve öğreticinin not değerlendirmeleri arasında güçlü bir korelasyon olduğunu göstermiştir.

5.1 Gelecekteki çalışmalar

Yapılan bu çalışmanın ilerdeki çalışmalarının geliştirilmesi aşamasında, siteme el yazısı tanıma modülü eklenerek, kağıt ortamında hazırlanmış sınavlar incelenip değerlendirilebilir ve sistemin çalışma kapsamı arttırılabilir.

Bununla beraber, sistemin değerlendirilmesi için kullanılan veri seti genişletilerek, hatta başka diller eklenerek, yabacı dil sınavları gibi çalışmanın kapsamı alanına alınabilir.

Sisteme eklenebilecek bir başka geliştirme ise, öğrenci cevapları kendi aralarında karşılaştırılarak, yazılacak yeni bir algoritma ile olası kopya teşebbüslerinin denetlenmesidir.

5.2 Kısıtlar

Geliştirilmiş olan klasik Soru değerlendirme çalışmasında bazı klasik soru yapıları, programın değerlendirmesinde kararlı bir sonuç sergilemeyebilir. Örneğin "Bilgisayar donanımlarından 5 tanesi yazınız." gibi sorularda, öğrencinin verdiği cevap, aradığımız cevapla aynı kelimeleri taşımayabilir. Buda değerlendirme sisteminin kapsamından çıkabilir. Bunun yanında klasik Soru Değerlendirme Karar Destek Sisteminde cevapların değerlendirilmesi ve yüksek başarı oranı aranan kelimelerin doğru seçilmesine bağlıdır.

Sitemin diğer bir kısıtlılığı da soruda geçen kelimelerin cevap cümlelerinde geçmesidir, bu da öğrenci hiçbir şey bilmediği halde soruyu yazarak puan almasına sebep olur. Bu nedenle soruda geçen kelimeler aranacak cevap içerisinden temizlenmelidir. "Türkiye'nin başkenti neresidir?" sorusunda aranılan asıl kelime "Türkiye, başkent" kelimeleri değil, "Ankara" kelimesidir.

REFERANSLAR

[1]-Yıldız, H. K., Gençtav, M., Usta, N., Diri, B., & Amasyalı, M. F. (2007). Metin Sınıflandırmada Yeni Özellik Çıkarımı. IEEE SIU 2007 15. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı.

[2]- İlhan, S., Duru, N., Karagöz, Ş., & Sağır, M. (2008). Metin Madenciliği ile Soru Cevaplama Sistemi.

[3]- KARACA, M. F., GÜNEL, M., & TAŞTAN, A. A. Metin Sınıflandırmada Benzerlik Hesaplama Tekniklerinin Değerlendirilmesi.

[4]- SEKER, S. E. Sosyal Ağlarda Akan Veri Madenciliği.

[5]- Tantuğ, A. C. (2012). Metin Sınıflandırma (Text Classification). TÜRKİYE BİLİŞİM VAKFI BİLGİSAYAR BİLİMLERİ ve MÜHENDİSLİĞİ DERGİSİ, 6(6).

[6]- KARADAĞ, A.,& TAKÇI, H. (2010). Metin Madenciliği ile Benzer Haber Tespiti. Akademik Bilişim.

[7]- AREAS, A. Veri Madenciliği Kavramı Ve Uygulama Alanları.

[8]- Huang, A. (2008, April). Similarity measures for text document clustering. InProceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand (pp. 49-56).

[9]- Ergün, K., Kubat, C., Çağıl, G., & Cesur, R. (2013). İnternet ortamındaki tüketici yorumlarından özet bilgi çıkarımı. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *17*(1).

[10]- ÖZEKES, S. (2005). Veri Madenciliği Modelleri ve Uygulama Alanları. İstanbul Ticaret Üniversitesi Dergisi..

[11] - TAŞGETİREN, S., YURAN, A. G. A. F., ÖZMEN, A. G. N., & ÖZKAN, A. G. N. BİYOMEDİKAL AR-GE 2015.

[12]- Savaş, S., Topaloğlu, N., & Yılmaz, M. (2012). Veri madenciliği ve Türkiye'deki uygulama örnekleri.

[13]- Argüden,Y. & Erşahin, B. (2008). Veri Madenciliği: Veriden Bilgiye, Masraftan Değere. *ARGE Danışmanlık Yayınları*, (10).

[14]- SEKER, S. E. Sosyal Ağlarda Akan Veri Madenciliği.

[15]- Sever, H.,& Oğuz, B. (2002). Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım: Kısım I: Eşleştirme Sorguları ve Algoritmalar. Bilgi Dünyası, 3(2), 173-204..

[16]- Özkan, Y. (2008). *Veri madenciliği yöntemleri*. Papatya Yayıncılık Eğitim.

[17]- Silahtaroğlu, G. (2008). Veri madenciliği. *İstanbul: Papatya yayıncılık*.

[18]-Chameleon: Hierarchical Clustering Using Dynamic Modeling, Karypis G. -Han E.-Kumar V, IEEE Computer, 1999 : 68-75

[19]- Baykasoğlu, A. (2005). Veri Madenciliği ve Çimento Sektöründe Bir Uygulama.*Akademik Bilişim*, 2-4.

[20]-Data Mining Concepts and Techniques, Han, J.-Kamber, M., Morgan Kaufmann Publishers, 1st Ed., San Francisco, USA, 2000.

[21]- Coşkun, (2013). *A Web Based Multi-Criteria Decision Support System for Department Selection Process of Vocational High School Students*. Boğaziçi Üniversitesi, İstanbul.

[22]- Zaki, M. J. (1999). Parallel and distributed association mining: A survey. *IEEE concurrency*, *7*(4), 14-25

[23]- Agrawal, ., Imielinski, T., & Swami, A. (1993). Database mining: A Performance perspective. Knowledge and Data Engineering, IEEE Transactions on, 5(6), 914-925

[24]- Dolgun, M. Ö., Özdemir, T. G., & Oğuz, D. (2009). Veri madenciliğinde yapısal olmayan verinin analizi: Metin ve web madenciliği. *İstatistikçiler Dergisi: İstatistik ve Aktüerya*, *2*(2).
[25]-ZOHAR, E.Y., Introduction to Text Mining, Supercomputing, Automated Learning Group National Center for Supercomputing Applications, University of Illinois,2002.
[26]- Zhang, X.,& Zhang, H. (2007). Comments on data mining static code attributes to learn defect predictors. *IEEE Transactions on Software Engineering*, *33*(9), 635-636.
[27]-Daş, R. (2008). Web Kullanıcı Erişim Kütüklerinden Bilgi Çıkarımı. Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği, Doktora Tezi, Elazığ.
[28]- Tan, A.H., Yu, P.S. (2004), Guest Editorial: Text and Web Mining, Applied Intelligence 18, 239-241, Kluwer Academic Publisher.
[29]- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620.
[30]- *Bagga, A*.,& Baldwin, B. (1998, August). Entity-based cross-document coreferencing using the vector space model. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1 (pp. 79-85). Association for Computational Linguistics.
[31]- İlhan, S., Duru, N., Karagöz, Ş., & Sağır, M. (2008). Metin Madenciliği ile Soru Cevaplama Sistemi.
[32]- Işık, M.,& Çamurcu, A. Y. (2008). Web belgeleri kümelemede benzerlik ve uzaklık ölçütleri başarılarının karşılaştırılması.
[33] Adalı, E. (2012) Doğal Dil İşlemenin Temelleri. URL: http://www.xn--adal-oza.net/wp-content/uploads/2012/10/Do%C4%9Fal-Dil-%C4%B0%C5%9Fleme.pdf
[34]- Derici, C., Celik, K., Ozgur, A., Gungor, T., Kutbay, E., Aydin, Y., & Kartal, G. (2014, April). Rule-based focus extraction in Turkish question answering systems. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd* (pp. 1604-1607). IEEE.
[35]- Amasyalı, M. F.,& Diri, B. (2005). Bir soru cevaplama sistemi: Baybilmiş.*TÜRKİYE BİLİŞİM VAKFI BİLGİSAYAR BİLİMLERİ ve MÜHENDİSLİĞİ DERGİSİ*, *1*(1).
[36]- Watson Projesi, (2015), URL: http://kobitek.com/ibm_watson_projesi
[37]- Altan, Z.,& Orhan, Z. (2005). Anlam Belirsizliği İçeren Türkçe Sözcüklerin Hesaplamalı Dilbilim Uygulamalarıyla Belirginleştirilmesi, XX. *Ulusal Dilbilim Kurultayı, İstanbul*.
[38]- İTÜ DDi Çalışması, (2015), URL: http://tools.nlp.itu.edu.tr/
[39]- YTÜ DDi Çalışması, (2015), URL: http://www.kemik.yildiz.edu.tr/
[40]- Akın, M. D. & Akın, A. A.(2007).Türk dilleri için açık kaynaklı doğal dil işleme kütüphanesi:ZEMBEREK. Elektrik Mühendisliği, 431, 38.
[41]- Libre Office Resmi Web sitesi, (2015), URL: http://extensions.libreoffice.org/
[42]- Java Dili Resmi Web Sitesi, (2015), URL: https://java.com/tr/about/
[43]- Akın, A. A.,& Akın, M. D. (2007). Zemberek, an open source nlp framework for turkic languages. *Structure*, *10*.
[44]- Pardus sayfası, (2015), URL: http://pardus.org.tr
[45]- Zemberek DDİ Kütüphanesi indirme sayfası, (2015), URL: http://zemberek.googlecode.com/svn/trunk/zemberek2/src/cekirdek/ net/ zemberek/ bilgi/ kokler/
[46]- Cöltekin, C. (2010, May). A Freely Available Morphological Analyzer for Turkish. In *LREC*.
[47]- Chan, Y. H. (2003). Biostatistics 104: correlational analysis. Singapore Med J,44(12), 614-9.