**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# ARİTMETİK PROBLEMLERİN ÇÖZÜMLENMESİ VE SİMGELENMESİ

**YÜKSEK LİSANS TEZİ**

**Müh. Ömer BAHADIR**

**Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**

**Programı : BİLGİSAYAR MÜHENDİSLİĞİ**

**HAZİRAN 2007**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# ARİTMETİK PROBLEMLERİN ÇÖZÜMLENMESİ VE SİMGELENMESİ

**YÜKSEK LİSANS TEZİ**

**Müh. Ömer BAHADIR**
**(504041529)**

**Tezin Enstitüye Verildiği Tarih : 4 Mayıs 2007**
**Tezin Savunulduğu Tarih : 13 Haziran 2007**

**Tez Danışmanı : Prof.Dr. Coşkun SÖNMEZ**
**Diğer Jüri Üyeleri Prof.Dr. Eşref ADALI (İ.T.Ü.)**
**Dr. Yaşar ERENLER (İ.T.Ü.)**

**HAZİRAN 2007**

## ÖNSÖZ

Tezin konusu, aritmetik matematik problemlerinin doğal dil işleme aşamalarından faydalanılarak çözümlenmesi ve simgelenmesidir. Geçmişten günümüze kadar problem çözme tabanlı sistemler üzerinde matematik, fizik, mekanik gibi çeşitli alanlarda uygulamalar yapılmaktadır. Ancak yapılan bu çalışmaların sayısı ve çeşitliliği yeterli değildir.

Problem çözme tabanlı sistemler, doğal dil işleme konusunun kapsamı altında incelenmektedir. Geliştirilen uygulama ve sonuçlarından önce doğal dil işleme çalışmaları ve bu alanda yapılan uygulamalar tanıtılmaktadır. Problem çözme tabanlı sistemlerin tarihsel gelişimleri hakkında bilgi verildikten sonra bu sistemler içinde öne çıkan bazı uygulamalar incelenmektedir.

Tezin konusuna yönelik geliştirilen uygulamada, bilgisayara girilmiş olan aritmetik problem çözümlenip, ilköğretim öğrencilerinin anlayacakları şekilde simgelenmektedir. Yapılan bu çalışmadaki amaç; Türkçe doğal dil işleme uygulamalarının sayısını ve çeşitliliğini artırmak, yeni fikirlere öncülük etmek, ilköğretim öğrencilerine aritmetik problemlerinin çözümü konusunda yardımcı olmak ve test imkanı sunmak, ilköğretim öğrencileri arasında bilgisayar kullanımını teşvik etmek, insanı anlayan makine kavramına katkıda bulunmaktır.

Bu tez çalışmam süresince her türlü yardımlarını esirgemeyen hocam Sayın Prof. Dr. Coşkun SÖNMEZ'e ve manevi desteklerini benden esirgemeyen aileme ve arkadaşlarıma teşekkür ederim.

**Haziran 2007** **Ömer BAHADIR**

**İÇİNDEKİLER**


**KISALTMALAR** **v**
**TABLO LİSTESİ** **vi**
**ŞEKİL LİSTESİ** **vii**
**SEMBOL LİSTESİ** **viii**
**ÖZET** **ix**
**SUMMARY** **xi**

**1. GİRİŞ** **1**

**2. DOĞAL DİL İŞLEME – DDİ** **3**
2.1. Doğal Dil İşleme Nedir ? 3
2.2. Yapay Zeka – Doğal Dil İşleme İlişkisi 3
2.3. Doğal Dil İşleme Uygulamaları 4
2.4. Doğal Dil İşleme Alanları 5
2.4.1. Doğal Dil Anlama 5
2.4.1.1. Biçimbirimsel Analiz (Morfolojik Analiz) 6
2.4.1.2. Anlam Belirginleştirme (POS Etiketlemesi) 7
2.4.1.3. Sözdizimsel Analiz (Sintaks Analizi) 11
2.4.1.4. Anlambilimsel Analiz (Semantik Analiz) 12
2.4.1.5. Söylem Analizi 12
2.4.2. Doğal Dil Üretme 12

**3. PROBLEM ÇÖZME TABANLI SİSTEMLER** **14**
3.1. STUDENT 14
3.2. CARPS 17
3.3. MECHO 18
3.4. ALİ 19
3.4.1. Morfolojik Analiz Seviyesi 20
3.4.2. Sözdizimsel Analiz Seviyesi 20
3.4.3. Anlamsal Analiz Seviyesi 22
3.4.4. Örnek Problemler 22

**4. UYGULAMAYA YÖNELİK TEMEL BİLGİLER** **24**
4.1. Türkçe Dilbilgisi Temel Kuralları 24
4.1.1. İsim Tamlamaları 24
4.1.1.1. Belirtili İsim Tamlaması 25
4.1.1.2. Belirtisiz İsim Tamlaması 25
4.1.1.3. Takısız İsim Tamlaması 25
4.1.1.4. Zincirleme İsim Tamlaması 26
4.1.2. Türkçe Ortografik Kurallar 26

4.1.2.1. Büyük Ünlü Uyumu 26
4.1.2.2. Küçük Ünlü Uyumu 27
4.1.2.3. Ünlü Düşmesi 27

**5. ARİTMETİK PROBLEMLERİN ÇÖZÜMLENMESİ VE SİMGELENMESİ 28**
5.1. Temel Adımlar 28
5.2. Biçimbirimsel Analiz (Morfolojik Analiz) Modülü 28
5.2.1. Zemberek 29
5.2.1.1. Zemberek Nedir ? 29
5.2.1.2. Zemberek Nasıl Çalışır ? 30
5.2.2. Tez Kapsamında Zemberek Kullanımı 32
5.3. Eklerin Belirlenmesi Modülü 34
5.4. Soru Kalıbının Belirlenmesi Modülü 36
5.4.1. Karar Ağaçları 36
5.4.2. Uygulama Kapsamında Karar Ağaçlarının Kullanılması 40
5.4.2.1. Öğrenme Kümesinin Oluşturulması 40
5.4.2.2. Karar Ağacının Oluşturulması 44
5.4.2.3. Karar Ağaçları ile Soru Kalıbının Belirlenmesi 48
5.5. Sorunun Çözülmesi Modülü 48
5.6. Çözümün denklemler şeklinde ifade edilmesi ve gösterilmesi modülü 49
5.7. Sorunun ve Çözümün Simgelenmesi Modülü 50

**6. EKRAN GÖRÜNTÜLERİ VE ÖRNEK UYGULAMALAR 52**
6.1. Örnek Uygulama I 52
6.2. Örnek Uygulama II 56

**7. SONUÇLAR VE TARTIŞMA 62**

**KAYNAKLAR 64**

**ÖZGEÇMİŞ 66**

**KISALTMALAR**

| | |
|---|---|
| **DDİ** | : Doğal Dil İşleme |
| **NLP** | : Natural Language Processing |
| **POS** | : Part of Speech |
| **CARPS** | : CAlculus Rate Problem Solver |
| **ALİ** | : Aritmetikçi Lisan İşleyici |
| **Ö** | : Özne |
| **D** | : Dolaylı Tümleç |
| **Z** | : Zarf Tümleci |
| **N** | : Nesne |
| **Y** | : Yüklem |
| **İG** | : İsim Grubu |
| **SG** | : Sıfat Grubu |
| **İN** | : İsim Nesnesi |
| **SN** | : Sıfat Nesnesi |
| **DZ** | : Diğer Zarflar |
| **S** | : Sıfat |
| **İ** | : İsim |
| **ZB** | : Zaman Belirteçleri |
| **T** | : Tamlayan |
| **TN** | : Tamlanan |
| **ZM** | : Zamir |
| **NE** | : Nesne Eki |
| **TE** | : Tamlayan Eki |
| **TNE** | : Tamlanan Eki |
| **KE** | : Kip Eki |
| **ZE** | : Zaman Eki |
| **DE** | : Dolaylı Tümleç Eki |
| **EF** | : Ek Fiil |
| **Cl** | : Class (Sınıf) |
| **Nit** | : Nitelik |

**TABLO LİSTESİ**


| | | Sayfa No |
|---|---|---|
| **Tablo 2.1** | Eklemeli morfoloji örnekleri …….. | 7 |
| **Tablo 2.2** | Türetmeli morfoloji örnekleri ……..… | 7 |
| **Tablo 2.3** | "yüz" kelimesinin isim anlamları -1…….…… | 9 |
| **Tablo 2.4** | "yüz" kelimesinin isim anlamları -2…….…… | 9 |
| **Tablo 2.5** | "yüzmek" kelimesinin anlamları - 1…..…….…… | 10 |
| **Tablo 2.6** | "yüzmek" kelimesinin anlamları - 2 …….……... | 10 |
| **Tablo 4.1** | Zincirleme isim tamlaması örnekleri ……..…….. | 26 |
| **Tablo 4.2** | Küçük ünlü uyumuna örnek kelimeler …… | 27 |
| **Tablo 5.1** | Nitelikler ve 10 tane örnek problem için belirlenmiş nitelik değerleri -1 …….. | 42 |
| **Tablo 5.2** | Nitelikler ve 10 tane örnek problem için belirlenmiş nitelik değerleri -2 …….. | 43 |

## ŞEKİL LİSTESİ


| | | Sayfa No |
|---|---|---|
| **Şekil 2.1** | **:** Doğal Dil Anlama'nın Aşamaları..... | 6 |
| **Şekil 2.2** | **:** Türkçe'deki kelime sınıfları..... | 8 |
| **Şekil 2.3** | **:** Örnek cümlenin sözdizimsel analizi..... | 11 |
| **Şekil 2.4** | **:** Doğal Dil Üretme'nin Aşamaları..... | 13 |
| **Şekil 3.1** | **:** MECHO Modülleri..... | 19 |
| **Şekil 3.2** | **:** Örnek problemin morfolojik analizi ..... | 20 |
| **Şekil 3.3** | **:** Örnek problem için gramer kuralları..... | 21 |
| **Şekil 3.4** | **:** Örnek problem için aday diziler..... | 21 |
| **Şekil 3.5** | **:** Örnek problemin sözdizimsel analiz sonucu..... | 22 |
| **Şekil 5.1** | **:** Uygulamanın modülleri ve bu modüllerin işleyiş sırası..... | 29 |
| **Şekil 5.2** | **:** Örnek kelime için kök adaylarının bulunması..... | 32 |
| **Şekil 5.3** | **:** Yumuşama örneği..... | 32 |
| **Şekil 5.4** | **:** Örnek problem için morfolojik analiz sonucu..... | 33 |
| **Şekil 5.5** | **:** "yirminin" kelimesinin morfolojik analizi..... | 34 |
| **Şekil 5.6** | **:** Karar ağacı yapısı..... | 38 |
| **Şekil 5.7** | **:** Öğrenme kümesinin seçilmesi..... | 44 |
| **Şekil 5.8** | **:** Kök ve çocuk düğümler..... | 46 |
| **Şekil 5.9** | **:** Karar ağacı oluşturulması..... | 47 |
| **Şekil 5.10** | **:** Örnek problem..... | 51 |
| **Şekil 5.11** | **:** Örnek problemin simgelenmesi..... | 51 |
| **Şekil 6.1** | **:** Öğrenme işlemi sonucu karar ağacının oluşturulması..... | 52 |
| **Şekil 6.2** | **:** Örnek problem..... | 53 |
| **Şekil 6.3** | **:** Örnek problemin çözümü..... | 54 |
| **Şekil 6.4** | **:** Örnek problem için biçimbirimsel çözümleme..... | 55 |
| **Şekil 6.5** | **:** Örnek problemin simgesel gösterimi..... | 56 |
| **Şekil 6.6** | **:** Örnek Sorular Ekranı..... | 57 |
| **Şekil 6.7** | **:** Örnek Soru Eklenmesi..... | 58 |
| **Şekil 6.8** | **:** Örnek Problemin Çözümü..... | 59 |
| **Şekil 6.9** | **:** Örnek problem için biçimbirimsel çözümleme..... | 60 |
| **Şekil 6.10** | **:** Örnek problemin simgesel gösterimi..... | 61 |

**SEMBOL LİSTESİ**

| | |
|---|---|
| **A** | **:** Niteliğe göre dallanma |
| **S** | **:** Veri kümesi |
| **v** | **:** A niteliğinin mevcut değerleri |

## ARİTMETİK PROBLEMLERİN ÇÖZÜMLENMESİ VE SİMGELENMESİ

## ÖZET

Doğal dil işleme çalışmaları önceleri yapay zeka biliminin alt kategorisi olarak değerlendirilirken oldukça kısa bir sürede araştırma konuları genişleyerek tek başına incelenen bir disipline dönüşmüştür. Bu hızlı değişimin nedeni son yıllarda bilginin ön plana çıkarak bilgi toplumu kavramını doğuran bilişim alanındaki önemli gelişmelerdir. İnternet üzerinden bilgiye ulaşmanın ve iletişimin kolaylaşması ile birlikte, erişilecek bilgi miktarı da aynı oranda artmıştır.

Doğal dil işleme alanında birçok uygulamalar yapılmış ve yapılmaya devam edilmektedir. Çeviri programları, ses tanıma sistemleri, bilgisayar destekli öğretim bu alanda yapılan çalışmalardan bazılarıdır. Bu çalışmalardan bazıları da problem çözme tabanlı sistemlerdir. Problem çözme tabanlı sistemler içinde matematik, mekanik, fizik problemlerini anlayan ve çözen sistemler üzerinde çalışılmaktadır.

Problem çözme tabanlı sistemler, bu tez çalışmasının ana konusunu oluşturmaktadır. Bu tez çalışmasında, ilköğretim öğrencileri seviyesindeki aritmetik matematik problemlerinin çözümlenmesi ve simgelenmesi gerçekleştirilmektedir. Geliştirilen uygulamanın modülleri; kelimelerin biçimbirimsel analizinin yapılması, kelimelerin aldıkları eklerin belirlenmesi, soru kalıbının belirlenmesi, sorunun çözülmesi, çözümün denklemler şeklinde ifade edilmesi, sorunun ve çözümün simgelenmesi olarak sıralanabilir.

Kelimelerin biçimbirimsel analizinin yapılması modülünde, kullanıcı tarafından girilen aritmetik problemdeki tüm kelimeler için biçimbirimsel analiz yapılmaktadır. Bu modül ile, tüm kelimelerin kökleri ve aldıkları ekler belirlenmektedir.

Kelimelerin eklerinin belirlenmesi modülünde, tüm kelimelerin aldıkları ekler kontrol edilerek cümle içindeki isim tamlamaları belirlenmektedir.

Soru kalıbının belirlenmesi modülünde, girilen problemin hangi tip bir aritmetik problem olduğu anlaşılmaktadır. Uygulama kapsamında, soru kalıbının belirlenmesi için veri madenciliği sınıflandırma yöntemlerinden karar ağaçları kullanılmıştır. İlk önce model oluşturulması için 75 tane örnek aritmetik problem ve bu problemlerin çözümlerini içeren bir öğrenme kümesi oluşturulmuş ve bu öğrenme kümesi ile sonlanma koşulları dikkate alınarak karar ağacı oluşturulmuştur. Sonra, oluşturulan karar ağacı ile girilen problemin hangi tip bir aritmetik problem olduğu belirlenmektedir.

Sorunun çözülmesi modülünde, hangi kalıba ait olduğu bilinen aritmetik problem çözülmektedir.

Çözümün denklemler şeklinde ifade edilmesi modülünde, girilen aritmetik problemin çözümü denklemler ve dört işlem sembolleriyle gösterilmektedir.

Sorunun ve çözümün simgelenmesi modülünde ise çözüm, ilköğretim 1., 2., 3. sınıf öğrencilerine kesirli sayıların anlatılmasında kullanılan simgelerle (şekillerle) ifade edilmektedir.

Bu tez çalışmasının amacı; Türkçe doğal dil işleme uygulamalarının sayısını ve çeşitliliğini artırmak, yeni fikirlere öncülük etmek, ilköğretim öğrencilerine aritmetik problemlerinin çözümü konusunda yardımcı olmak ve test imkanı sunmak, ilköğretim öğrencileri arasında bilgisayar kullanımını teşvik etmek, insanı anlayan makine kavramına katkıda bulunmaktır.

## SOLVING AND SYMBOLIZING ARITHMETIC PROBLEMS IN TURKISH

## SUMMARY

Formerly, it has appreciating that natural language processing applications are subfield of artificial intelligence, they have been changed to alone discipline with expanding research subjects quitely. The main reason of this quick changing is, the important improvement of informatics that have arised from concept of information society recently. With getting easy of accessing of information and communication over internet, a number of information to access has increased at the same time.

There have been a lot of applications at natural language processing. Translation programs, speech recognition systems, computer aided instruction are some studies of this subject. Some of these studies are problem solving systems. It has been working that the systems which are understanding and solving mathematics, mechanics and physics problems.

Problem solving systems constitute the main subject of this work. In this work, primary school-level arithmetic mathematics problems are solved and symbolized. The modules of this application are morphological analysis of every word, definition of affixes of words, definition of problem types, solving problems, expressing answer by equations and symbolizing of problem and its answer.

In the module of morphological analysis of every word, it is realized that morphological analysis of every word entered by user. In this module, it is defined all words roots and their affixes.

In the module of definition of affixes of words, it is defined noun phrase by controlling every word affixes.

In the module of definition of problem types, it is realized that problem is which type of arithmetic problem. In this work, decision trees which are one of the data mining classification methods have used to determine the problem types. Firstly, it has constituted data training set that contains 75 arithmetic problems and their solutions to develop a model and then has constituted decision trees to take end conditions into consideration. Then, it is realized that input problem is which type of arithmetic problem by this decision tree.

In the module of solving problems module, the arithmetic problem that is known which type of arithmetic problem is solved.

In the module of expressing answer by equations, answer of the arithmetic problem is expressed by equations and four artihmetic operations.

In the module of symbolizing of problem and its answer, the answer is expressed by symbols which are used to teach fractional numbers to primary school students.

The major reason of this work is to increase the number and kind of Turkish natural language processing applications, to pioneer new thoughts, to help and test primary

school students about their arithmetic problems, to encourage using of computer between primary school students and to contribute to concept of machine that understands person.

# 1. GİRİŞ

Doğal Dil İşleme, ana işlevi bir doğal dili çözümleme, anlama, yorumlama ve üretme olan bilgisayar sistemlerinin tasarımını ve gerçekleştirilmesini konu alan bir bilim ve mühendislik alanıdır. [1] Doğal dil işleme, yapay zeka ve dilbilimi yanında matematik ve bilgisayar bilimleriyle de beraber çalışmaktadır. Doğal dil işleme uygulamalarından bazıları ise şöyledir : makine çevirisi, metin özetleme, metin kategorileme, metinden-sese çeviri sistemleri, sesten-metine çeviri sistemleri, rapor üretimi, soru-cevap makineleri, bilgi erişimi, bilgi çıkarımı, kelime-işlem programları vb... Doğal dil işleme aşamaları ise biçimbirimsel analiz, sözdizimsel analiz, anlamsal analiz ve söylem analizi olarak sıralanabilir.

Doğal dil işleme alanında birçok uygulamalar yapılmış ve yapılmaya devam edilmektedir. Çeviri programları, ses tanıma sistemleri, bilgisayar destekli öğretim bu alanda yapılan çalışmalardan bazılarıdır. Bu çalışmalardan bazıları da problem çözme tabanlı sistemlerdir. Matematik, mekanik, fizik problemlerini anlayan ve çözen sistemler üzerinde çalışılmaktadır.

Türkçe bağlantılı bir dil olduğundan köklere ekler getirilerek binlerce yeni sözcük elde edilebilir. Bu özelliği nedeniyle Türkçe, geniş anlatım olanaklarına ve yeni sözcük türetme gücüne sahip bir dildir. Türkçe Ural-Altay dil ailesinin Altay koluna bağlı ve sondan eklemeli bir dildir.

Türkçe için yapılan doğal dil işleme çalışmaları geçmiş yıllara göre belli bir noktaya gelmiştir fakat hâlâ doğal dil işleme alanında uygulama eksikliği bulunmaktadır. Bu tezde, Türkçe aritmetik matematik problemlerinin çözümlenmesi ve simgelenmesi gerçekleştirilmiştir. Yapılan bu çalışma ile Türkçe doğal dil işleme uygulamalarının sayısını ve çeşitliliğini artırmak, yeni fikirlere öncülük etmek, ilköğretim 1., 2., 3. sınıf öğrencilerine aritmetik problemlerinin çözümü konusunda yardımcı olmak ve test imkanı sunmak, ilköğretim öğrencileri arasında bilgisayar kullanımını teşvik etmek, insanı anlayan makine kavramına katkıda bulunmak gibi amaçlar hedeflenmiştir.

Tezin ikinci bölümünde, doğal dil işleme kavramına genel bir giriş yapılmaktadır. Doğal dil işleme-yapay zeka ilişkisi, doğal dil işleme uygulama alanları tanıtılmakta ve doğal dil işleme aşamalarından biçimbirimsel (morfolojik) analiz, kelime sınıfının belirlenmesi, sözdizimsel analiz, anlamsal (semantik) analiz ve söylem analizi anlatılmaktadır.

Tezin üçüncü bölümünde, daha önceden geliştirilen problem çözme tabanlı sistemler hakkında bilgiler verilmektedir. Bu sistemlerin tarihsel gelişimleri anlatılırken, STUDENT, CARPS, MECHO ve ALİ gibi problem çözme tabanlı sistemlerin çalışma prensipleri incelenmektedir.

Tezin dördüncü bölümünde Türkçe dilbilgisi temel kuralları hakkında kısa bilgiler verilmektedir. Tez kapsamında kullanılan örnek problemlerde bulunan isim tamlamalarının yapısı ve diğer Türkçe dilbilgisi kuralları anlatılmaktadır.

Tezin beşinci bölümünde geliştirilen uygulama ve uygulamanın modülleri hakkında detaylı bilgiler verilmektedir. Bu bölümde, kelimelerin biçimbirimsel analizinin yapılması, biçimbirimsel analiz için kullanılan Zemberek programının çalışma prensipleri, kelimelerin aldıkları eklerin belirlenmesi, soru kalıbının karar ağaçları kullanılarak belirlenmesi, karar ağaçlarının yapısı ve çalışma prensipleri, sorunun çözülmesi, çözümün denklemler şeklinde ifade edilmesi ve gösterilmesi, sorunun ve çözümün simgelenmesi anlatılmaktadır.

Tezin altıncı bölümünde geliştirilen uygulamanın ekran görüntüleri verilmekte ve iki örnek problemin çözülmesi gösterilmektedir.

Son olarak, tezin yedinci bölümünde tezin konusuna dair sonuç ve tartışma bölümü yer almaktadır. Geleceğe yönelik yapılabilecek iyileştirmeler ve geliştirmeler bu bölümde belirtilmektedir.

## 2. DOĞAL DİL İŞLEME – DDİ

### 2.1. Doğal Dil İşleme Nedir ?

Doğal dil işleme, ana işlevi bir doğal dili çözümleme, anlama, yorumlama ve üretme olan bilgisayar sistemlerinin tasarımını ve gerçekleştirilmesini konu alan bir bilim ve mühendislik alanıdır. [1] Başka bir tanıma göre ise doğal dil işleme (DDİ), insana özgü tüm dillerin işlenmesi ve kullanılması amacıyla araştırma yapan bilim dalıdır. [2] DDİ, yapay zeka ve dil biliminin alt kategorisi olarak anılmakta ve bu bilimlerin yanında bilgisayar ve matematik gibi bilimlerden de yardım almaktadır. [3] Bu tanımlardan da açıkca görüldüğü üzere doğal dil işleme (Natural Language Processing), insanlar tarafından doğal dilde konuşulan, anlatılan ve bu anlama sonrası uygulanan işlemlerin bilgisayarlar tarafından da anlaşılması ve uygulanabilmesidir. DDİ, bilgisayarlar tarafından dilin morfolojisinin tanınabilmesini ve diller arasında otomatik olarak dönüşümler yapılabilmesini sağlar.

Doğal dil işleme çalışmaları önceleri yapay zeka biliminin alt kategorisi olarak değerlendirilirken, oldukça kısa bir sürede araştırma konuları genişleyerek tek başına incelenen bir disipline dönüşmüştür. Bu hızlı değişimin nedeni son yıllarda bilginin ön plana çıkarak bilgi toplumu kavramını doğuran bilişim alanındaki önemli gelişmelerdir. İnternet üzerinden bilgiye ulaşmanın ve iletişimin kolaylaşması ile birlikte, erişilecek bilgi miktarı da aynı oranda artmıştır. [4]

### 2.2. Yapay Zeka – Doğal Dil İşleme İlişkisi

Teknolojik gelişmelerin hızlanması ile birlikte makine-insan etkileşimleri de hızla ilerlemektedir. Artık bilgisayarların, insanları ve insanların konuşmalarını anlamaları ve bu doğrultuda işlemler yapmaları beklenmektedir. Giderek gelişen ve insanı anlayan makinelerin daha zeki olması insanın yaşam kalitesini yükselteceğinden, vazgeçilmez olması kaçınılmazdır. Zeki makine kavramı, yapay zeka çalışmalarının hızlanmasına yol açmıştır. Geleceğin en önemli sektörlerinden biri olan yapay zeka ile insanın iletişim kuracağı tek araç dildir. [2]

## 2.3. Doğal Dil İşleme Uygulamaları

Doğal Dil İşleme'nin amacı doğal dillerin kurallı yapısının çözümlenerek anlaşılması ve yeniden üretilmesidir. [2] Bu amaç doğrultusunda gerçekleştirilen doğal dil işleme çalışmaları aşağıda özetlenmiştir:

**Makine Çevirisi**

Diller arası çevrimin bilgisayarlar tarafından başarılı bir şekilde yapılması hedeflenmektedir. Örneğin Türkçe'den İngilizce'ye, İngilizce'den Türkçe'ye çevirilerin yapılması.

**Metin Özetleme**

Verilen bir metnin özetinin bilgisayarlar tarafından çıkarılması hedeflenmektedir. Çeşitli diller için metin özetleme programları ve algoritmaları bulunmaktadır.

**Metin Kategorileme**

Doğal dille yazılmış bilgilerin bilgisayarlar tarafından otomatik olarak sınıflandırılması hedeflenmektedir. Bir metnin yazarının bulunması, web sayfalarının içeriklerine göre sınıflandırılması ve gazetelerin web sayfalarındaki haber metinlerinin otomatik sınıflandırılması [5] bu tip çalışmalara örnek olarak verilebilir.

**Metinden – Sese (Text-to-Speech) Çeviri Sistemleri**

Verilen bir metnin bilgisayarlar tarafından sese çevrilmesi hedeflenmektedir. Bu konuda birçok dilde yapılmış çalışmalar bulunmaktadır. İngilizce için "ReadPlease", Türkçe için ise "GVZ" ve "SpeakTrk" uygulamaları örnek olarak verilebilir.

**Sesten – Metine (Speech-to-Text) Çeviri Sistemleri**

Bu sistemler, sesten metine çeviri yapılmasını hedeflemektedir. Ses sinyallerinin analiz edilerek kelime dizilerine çevrilmesi gerçekleştirilir. Bu konuda Türkçe de dahil olmak üzere birçok dilde yapılmış çalışmalar bulunmaktadır.

**Rapor Üretimi**

Bu sistemler, verilen bir metinle ilgili bilgisayar tarafından otomatik olarak rapor üretilmesini hedeflemektedir.

**Soru – Cevap Makineleri**

İnsan ile bilgisayar arasında karşılıklı dialog kurulması ve sorulan sorulara bilgisayar tarafından doğru cevaplar verilmesi hedeflenmektedir.

**Bilgi Erişimi (Information Retrieval)**

Belli bir kullanıcının istediği bilgiye (metin, resim gibi) ulaşmasını sağlayan sistemler tasarlanması hedeflenmektedir.

**Bilgi Çıkarımı (Information Extraction)**

Bu sistemler, önceden belirlenmiş bir formata uygun bilgilerin büyük metin dosyaları arasından bulunup çıkartılmasını hedeflemektedir. [6]

**Kelime – İşlem Programları**

Yanlış yazılan kelimeler için kelimeler önerilmesi gibi kelimeler ile ilgili işlemlerin yapılması hedeflenmektedir.

## 2.4. Doğal Dil İşleme Alanları

Doğal dil işleme çalışmaları, doğal dil anlama ve doğal dil üretme olmak üzere iki alt alana ayrılmaktadır.

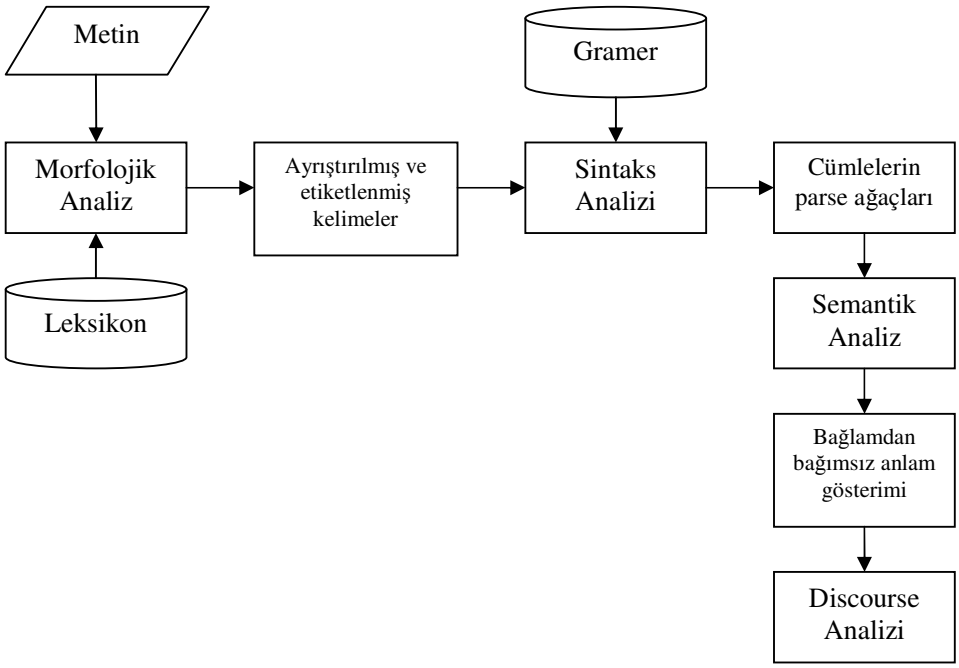
### 2.4.1. Doğal Dil Anlama

Doğal Dil Anlama, insanların belirli amaçlarla söyledikleri kelimelerin veya cümlelerin bilgisayarlar tarafından anlaşılmasını ifade etmektedir.

Doğal Dil Anlama'nın aşamaları aşağıda gösterilmektedir :

- Biçimbirimsel Analiz (Morfolojik Analiz)
- Anlam Belirginleştirme (POS Etiketlemesi)
- Sözdizimsel Analiz (Sintaks Analizi)
- Anlambilimsel Analiz (Semantik Analiz)
- Söylem Analizi

Doğal Dil Anlama'nın aşamaları Şekil 2.1'de gösterilmektedir.




**Şekil 2.1:** Doğal Dil Anlama'nın Aşamaları [7]

#### 2.4.1.1. Biçimbirimsel Analiz (Morfolojik Analiz)

Biçimbirimsel analiz, DDİ çalışmalarının ön aşamasını oluşturmaktadır. Kelimelerin, **morfem** denilen en küçük anlamlı birimlere ayrıştırılmasına biçimbirimsel analiz veya morfolojik analiz denilmektedir. [7] En küçük anlamlı birimden kasıt, kelimenin kökü ve ekleridir. Örnek olarak Türkçe "**masalarımdaki**" kelimesine bakılacak olunursa biçimbirimsel analizinin aşağıdaki gibi olduğu görülür :

masa + lar + ım + da + ki

masa : Kök

lar : Çoğul eki

ım : 1. tekil şahıs iyelik eki

da : İsmin -de hali (bulunma durumu)

ki : İlgi eki

Türkçe'de morfoloji eklemeli veya türetmeli olabilir. [7] Eklemeli ve türetmeli morfolojiye örnekler Tablo 2.1 ve Tablo 2.2'de gösterilmektedir.

**Tablo 2.1:** Eklemeli Morfoloji Örnekleri [7]

| | |
|---|---|
| **İsim** | Çoğul ekleri : ev/evler<br>İyelik ekleri : evim / evin / evi / evimiz / eviniz / evleri<br>İsmin –e hali : eve<br>İsmin –i hali : evi<br>İsmin -de hali : evde<br>İsmin -den hali : evden |
| **Fiil** | Zaman ekleri : gel / geldi / geliyor / gelmiş / gelecek<br>Hikaye ekleri : geliyordum / gelmişti / gelecekti<br>Zamir ekleri : geldim / geldin / geldi / geldik / geldiniz / geldiler |

**Tablo 2.2:** Türetmeli Morfoloji Örnekleri [7]

| |
|---|
| -cı : kapı / kapıcı |
| -laş : uygar / uygarlaş |
| -mek : gel / gelmek |
| -cik : mini / minicik |
| -li : Gebze / Gebzeli |

#### 2.4.1.2. Anlam Belirginleştirme (POS Etiketlemesi)

Bir kelimenin ait olduğu kelime sınıfına POS (part-of-speech) denilmektedir. [7] Metin içerisindeki tüm kelimelerin sınıflarının belirlenmesi işlemine de anlam belirginleştirme veya POS etiketlemesi denilmektedir. POS etiketlemesi, biçimbirimsel (morfolojik) analizden sonra ve sözdizimsel (sintaks) analizinden önce (gerekirse) yapılan bir işlemdir. Amacı, cümle içindeki tüm kelimelerin doğru POS'larının belirlenmesidir. [7] Türkçe'de çoğu kelimelerin birden fazla anlamları olabildiği için birden fazla da POS'ları olabilmektedir. Yukarıda da ifade edildiği gibi POS etiketlemesinin amacı, birden fazla POS'u olan kelimeler için geçersiz

POS'ların elenerek doğru POS'un tespit edilmesidir. Birden fazla POS'u olan kelimelere örnek olarak "yüz" kelimesinin anlamları [8]'den alınarak Tablo 2.3, 2.4, 2.5 ve 2.6'da gösterilmektedir.

Türkçe'deki kelime sınıfları Şekil 2.2'de gösterilmektedir.

| | Etiket | | Etiket |
|---|---|---|---|
| Özne | Ö | Zaman Belirteci | ZB |
| Dolaylı Tümleç | D/DT | Tamlayan | T |
| Zarf Tümleci | Z | Tamlanan | TN |
| Nesne | N | Zamir | ZM |
| Yüklem | Y | Nesne Eki | NE |
| İsim Grubu | İG | Tamlayan Eki | TE |
| Sıfat Grubu | SG | Tamlanan Eki | TNE |
| İsim Nesnesi | İN | Kip Eki | KE |
| Sıfat Nesnesi | SN | Zaman Eki | ZE |
| Diğer Zarflar | DZ | Dolaylı Tümleç Eki | DE |
| Sıfat | S | Ek Fiil | EF |
| İsim | İ | | |

**Şekil 2.2:** Türkçe'deki kelime sınıfları

**Tablo 2.3: "**yüz" kelimesinin isim anlamları -1

| |
|---|
| **1 .** Doksan dokuzdan sonra gelen sayının adı. |
| **2 .** Bu sayıyı gösteren 100, C rakamlarının adı. |
| **3 .** *sıfat* On kere on, doksan dokuzdan bir artık. |
| **4 .** Kere, kat vb. kelimeler ile birlikte kullanılarak yapılan işin çokluğunu abartılı bir biçimde anlatan söz:<br>*"Hikmet Bey'in kurum ve edası, her zamankinden belki yüz kat üstündü."*- S. M. Alus. |

**Tablo 2.4:** "yüz" kelimesinin isim anlamları -2

| |
|---|
| **1 .** Başta, alın, göz, burun, ağız, yanak ve çenenin bulunduğu ön bölüm, sima, çehre, surat. |
| **2 .** Yüzey, satıh:<br>*"Suyun yüzünde."*- . |
| **3 .** Kesici araçlarda ağız:<br>*"Bıçağın keskin yüzü."*- . |
| **4 .** Bir kumaşın dikiş sırasında dışa getirilen gösterişli bölümü. |
| **5 .** Yorgana ve yastığa geçirilen kılıf. |
| **6 .** Bir şeyin görünen bölümünde kullanılan kumaş:<br>*"Yorgan yüzü. Kanepenin yüzü."*- . |
| **7 .** Birinin görülegelen veya umulan hoşgörürlüğüne güvenilerek gösterilen cüret:<br>*"Ne yüzle? Yüzü olmamak."*- . |
| **8 .** Nedeniyle, sebebiyle. |
| **9 .** Yan, taraf. |
| **10 .** Bir yapının dışa bakan düşey yüzeylerinin her biri. |
| **11 .** *mecaz* Utanma. |

**Tablo 2.5: "**yüzmek" kelimesinin anlamları - 1

| |
|---|
| **1 .** Kol, bacak, yüzgeç vb. organların özel hareketleriyle su yüzeyinde veya su içinde ilerlemek, durmak:<br>*"Yüzmek bilmediği için on dakika içinde boğulmuştu."*- S. F. Abasıyanık. |
| **2 .** Yüzme sporu yapmak. |
| **3 .** Bir sıvının yüzeyinde batmadan durmak:<br>*"Tahta suda yüzer."*- . |
| **4 .** Herhangi bir durumun en aşırı derecesinde olmak:<br>*"Hiçbir kaygının gölgelemediği bir saadet içinde yüzmektedir."*- H. Taner. |
| **5 .** *mecaz* Dalgalanmak:<br>*"Korkma! Sönmez bu şafaklarda yüzen al sancak, / Sönmeden yurdumun üstünde tüten en son ocak."*- M. A. Ersoy. |
| **6 .** *mecaz* Herhangi bir şeyle üzeri kaplanmak, bir şeye bulanmak:<br>*"Kitaplar toz içinde yüzüyor. Ev pislik içinde yüzüyor."*- . |

**Tablo 2.6: "**yüzmek" kelimesinin anlamları - 2

| |
|---|
| **1 .** Derisini çıkarmak, derisini soymak. |
| **2 .** *halk ağzında* Çok para istemek. |

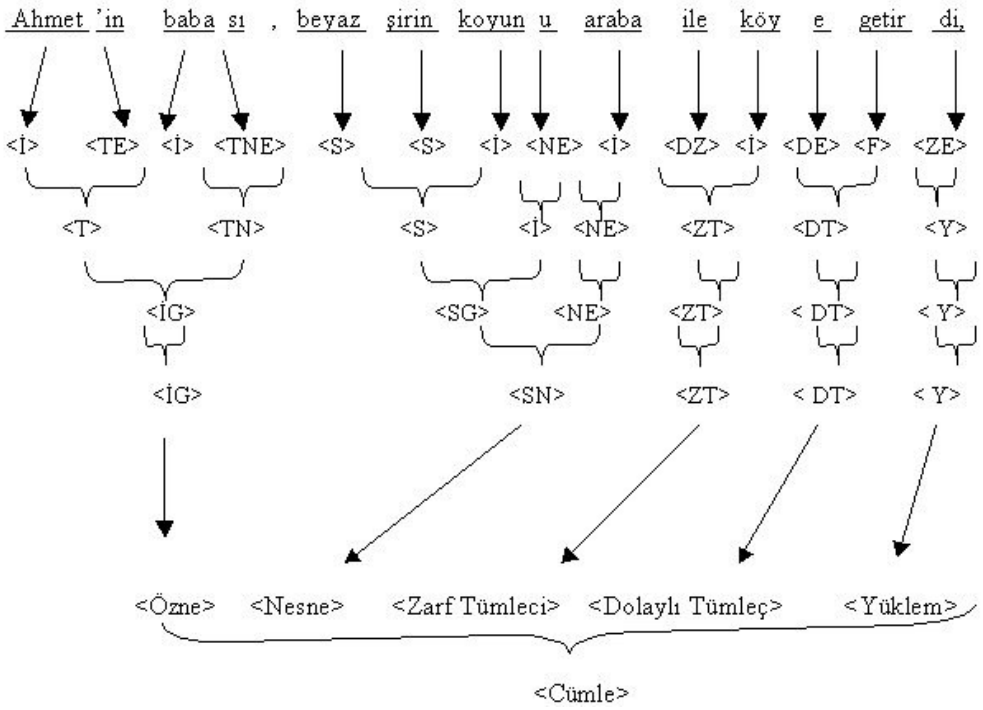
### 2.4.1.3. Sözdizimsel Analiz (Sintaks Analizi)

Sözdizimsel analiz, sözdizimini (sintaks) veya cümleyi oluşturan morfolojik öğelerin hiyerarşik kurallara uyumunu karşılaştırarak ölçümlemektir. [2]

Sözdizimsel analizin amacı bir giriş cümlesini alarak bu cümleyi temsil eden ayrıştırma (parse) ağaçlarını oluşturmaktır. [7]

Sözdizimsel analizin yapılabilmesi için leksikona, gramere ve ayrıştırma algoritmasına gerek vardır. Leksikon, kelimelerin köklerini ve POS bilgilerini içeren sözlük manasına gelmektedir. Bir dildeki cümle türleri ve cümledeki öğelerin diziliş sırası kurallarını içeren formel tanımlamalara gramer adı verilmektedir. Ayrıştırma algoritmaları, giriş cümlesini alarak dilbilgisi kurallarına uygun biçimde ayrıştırma ağaçlarını oluşturma amacını taşımaktadır. Bir cümlenin birden çok ayrıştırma ağacı olabilmektedir. [7]

"Ahmet'in babası, beyaz şirin koyunu araba ile köye getirdi." [9] cümlesinin hiyerarşik yapısı Şekil 2.3'de görülmektedir.




**Şekil 2.3:** Örnek cümlenin sözdizimsel analizi

#### 2.4.1.4. Anlambilimsel Analiz (Semantik Analiz)

Anlambilimsel analiz, sözdizimini oluşturan morfolojik öğelerin ayrılması yani, sözdizimsel analiz ile anlam taşıyan kelimelerin sınıflandırılması işleminden sonra gelen anlamlandırma veya anlama sürecidir. Bu süreçte anlam taşıyan kelimelerin, ekler ve cümle hiyerarşisi içindeki konumlarının saptanması sayesinde birbirleri ile ilişkileri kurulabilir. Bu ilişkiler anlam çıkarma, fikir yürütme gibi ileri seviye bilişsel fonksiyonların oluşturulmasında ham bilgi olarak kullanılmaktadır. [2]

Anlambilimsel analizin amacı, sözdizimsel analiz sonucu oluşan ayrıştırma ağaçlarından gramatik olarak doğru, ancak anlam olarak yanlış ağaçların elenmesidir. Bir anlamsal analiz sistemi bir metni analiz ettikten sonra o metinle ilgili soruları cevaplayabilmektedir. [7]

#### 2.4.1.5. Söylem Analizi

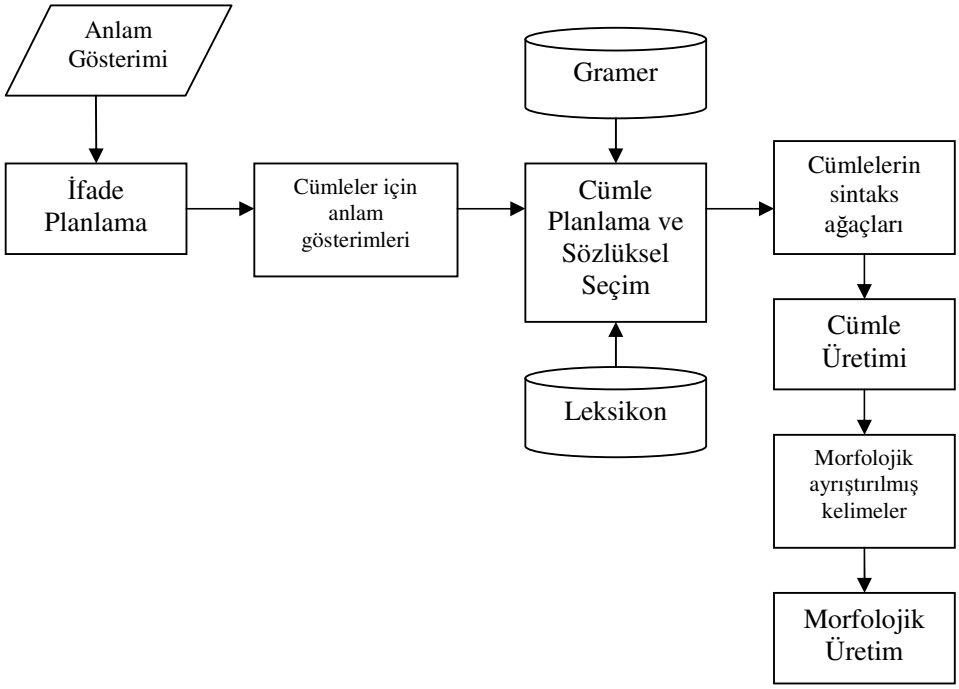
Arka arkaya yazılan ve birbiri ile konu ve anlam bütünlüğü olan cümleler içeren metinlere söylem (discourse) denilmektedir. [7] Söylem, dilbiliminde konuşmanın (speech) dengi olarak ta kabul edilmektedir.

Söylem analizi, metinlerin anlambilimsel ve sözdizimsel yapılarını inceler. Amacı bir metin verildiğinde o metinle ilgili sorulara cevap verebilmek, metin içinde geçen dolaylı konular hakkında çıkarım yapmaktır. [7]

### 2.4.2. Doğal Dil Üretme

Doğal dil üretme, doğal dil anlama'nın tam tersi olacak şekilde söylenmek istenen kelimelerin veya cümlelerin Türkçe gibi bir doğal dille temsil edilmesini ifade etmektedir.

Doğal Dil Üretme'nin aşamaları Şekil 2.4'de gösterilmektedir.




**Şekil 2.4:** Doğal Dil Üretme'nin Aşamaları [7]

## 3. PROBLEM ÇÖZME TABANLI SİSTEMLER

Bu bölümde, Türkçe ve yabancı diller için geliştirilen "problem çözme" sistemleri hakkında bilgi verilmektedir.

### 3.1. STUDENT

1964 yılında Daniel Bobrow tarafından geliştirilen STUDENT [10] isimli program cebir problemlerini analiz eden ve çözen ilk programdır. İlköğretim aritmetik problemlerini İngilizce doğal dilde anlayıp, denklemler kuran bir programdır.

STUDENT, kelime eşleştirme (pattern matching) tekniğine göre çalışmaktadır. Program her cümlenin bir denklem olduğunu varsaymaktadır. Programa önceden bazı denklemleri oluşturmasına yardımcı olacak belirli kelimeler hakkında bilgi yüklenmiştir. STUDENT, verilen aritmetik problem içindeki "eşittir", "fazladır", "eksiktir" gibi kelimeleri operator olarak tayin edip, bu kelimeler etrafındaki diğer kelimeleri değişkenler olarak atamaktadır. Program bu istenilen anlamlar ile sözdizimini (syntax) çözümlemektedir.

Aşağıda örnek problemler görülmektedir. [10]

(THE PROBLEM TO BE SOLVED IS)

(TOM HAS TWICE AS MANY FISH AS MARY HAS GUPPIES . IF MARY HAS 3 GUPPIES , WHAT IS THE NUMBER OF FISH TOM HAS Q.)

(THE EQUATIONS TO BE SOLVED ARE)

(EQUAL G02520 (NUMBER OF FISH TOM (HAS / VERB)))

(EQUAL (NUMBER OF GUPPIES (MARY / PERSON) (HAS / VERB)) 3)

(EQUAL (NUMBER OF FISH TOM (HAS / VERB)) (TIMES 2 (NUMBER OF GUPPIES (MARY / PERSON) (HAS /VERB))))

(THE NUMBER OF FISH TOM HAS IS 6)

Yukarıdaki örnek için STUDENT, Tom'un insan olduğunun ve guppy (ufak balık) kelimesinin ne anlama geldiğinin farkında değildir. Tamamen sözdizimsel bilgilere dayanarak doğru sonucu bulmaya çalışmaktadır. Aşağıdaki örnek için STUDENT programının çalışma mantığı şöyledir:

(THE GAS CONSUMPTION OF MY CAR IS 15
MILES PER GALLON. IF THE DISTANCE
BETWEEN BOSTON AND NEW YORK IS 250
MILES, WHAT IS THE NUMBER OF GALLONS
OF GAS USED ON A TRIP BETWEEN NEW YORK
AND BOSTON Q.)

İlk önce aritmetik problem içinde geçen tüm kelimelerin etiketleri (kelime sınıfları) bulunur. Etiketlenmiş problem aşağıdaki gibi gösterilmektedir.

(THE GAS CONSUMPTION (OF/OP)(MY/PRON)
CAR (IS/VERB) 15 MILES PER GALLON
(PERIOD/DIW) IF THE DISTANCE BETWEEN
BOSTON AND NEW YORK (IS/VERB) 250
MILES, (WHAT/QWORD) (IS/VERB) THE NUMBER
(OF/OP) GALLONS (OF/OP) GAS USED ON A
TRIP BETWEEN NEW YORK AND BOSTON (Q/DIM))

Bir sonraki aşamada program, cümleleri kendi istediği şekilde cümleler halinde tanımlar. İlk cümle değişmezken, ikinci cümlede geçen "IF" kelimesi tanınır ve ikinci cümle iki küçük cümleye bölünür.

(THE DISTANCE BETWEEN BOSTON AND NEW

YORK IS 250 MILES. WHAT IS THE NUMBER

OF GALLONS OF GAS USED ON A TRIP BETWEEN

NEW YORK AND BOSTON Q.)

Program daha sonra cümleleri denklemlere dönüştürür. Buradaki temel mantık, kelimelerin cebirsel karşılıklarına çevrilmesidir. Örneğin "is" kelimesinin "equals" (eşittir) işaretine, "times" kelimesinin çarpma işaretine çevrilmesi gibi.

(EQUAL X00001 (NUMBER OF GALLONS OF

GAS USED ON TRIP BETWEEN NEW YORK AND BOSTON))

(EQUAL (DISTANCE BETWEEN BOSTON AND NEW YORK) (TIMES 250(MILES)))

(EQUAL (GAS CONSUMPTION OF MY CAR)

(QUOTIENT(TIMES 15(MILES)) (TIMES l(GALLON))))

Yukarıdaki şekildeki yeni üretilen X00001 değişkeni İngilizce'deki "a" veya "the" kelimelerinin kabul edilmemesinden kaynaklanmaktadır. Program son olarak denklemleri çözmeye çalışır ve sonucu ekrana yazar.

(THE NUMBER OF GALLONS OF GAS USED ON A

TRIP BETWEEN NEW YORK AND BOSTON IS

16.66 GALLONS)

STUDENT, sözdizimsel ve anlamsal analiz olarak yeterli kapasiteye sahip değildir. İngilizcenin çok küçük bir kelime hazinesini kullanarak problemleri çözmektedir.

Fakat bu alanda yapılan ilk çalışma olması nedeniyle problem çözme sistemleri arasında önemli bir yer tutmaktadır.

### 3.2. CARPS

Aritmetik matematik problemlerini çözmek için geliştirilen diğer bir program da Eugene Charniak tarafından geliştirilen CARPS (CAlculus Rate Problem Solver) isimli programdır. [11] Bobrow'un STUDENT isimli programına çok benzemekle beraber, aralarındaki temel fark CARPS'ta kullanılan yapıların farklı olmasıdır. CARPS, sadece oran-orantı problemlerini çözebilmektedir ve LISP [12] ve CONVERT [13] dilleri ile geliştirilmiştir. Kelime eşleştirme işlemleri için CONVERT dili kullanılmıştır.

CARPS'ın çalışma mantığı STUDENT ile benzer olduğu için burada birkez daha tekrarlanmamıştır. CARPS, [11]'e göre aşağıdaki yönleriyle Bobrow'un STUDENT isimli programından daha güçlü bir yapıya sahiptir.

1. CARPS, aynı nesneyi STUDENT'e göre daha fazla kelimelerle tanımlayabilmektedir.
2. CARPS, bir nesne hakkında bilgileri parça parça toplayabilmektedir. Aritmetik problemlerde bir denklem için iki veya daha fazla cümleden çıkarımlar yapmak çok yaygın değildir.
3. CARPS, giriş cümlelerini ayrıştırmak için kendi bilgilerini kullanmakta sınırlı bir öğrenme düzeyine sahiptir.

Aşağıdaki CARPS'ın çözdüğü örnek bir problem [11] bazı ara aşamalar atlanarak gösterilmektedir.

(A SHIP IS 30.0 MILES SOUTH OF POINT 0

AND TRAVELING WEST AT 2 5 .0 MILES PER

HOUR /. HOW FAST IS THE DISTANCE FROM

THE SHIP TO 0 INCREASING ?)

(THE EQUATION SET IS)

((EQUAL (GOO 12) (EXPT (PLUS

(TIMES 899.99998 (EXPT MILE 2 . ) )

(TIMES 624.99998 (EXPT TIM 2 . ) (EXPT MILE 2 .)

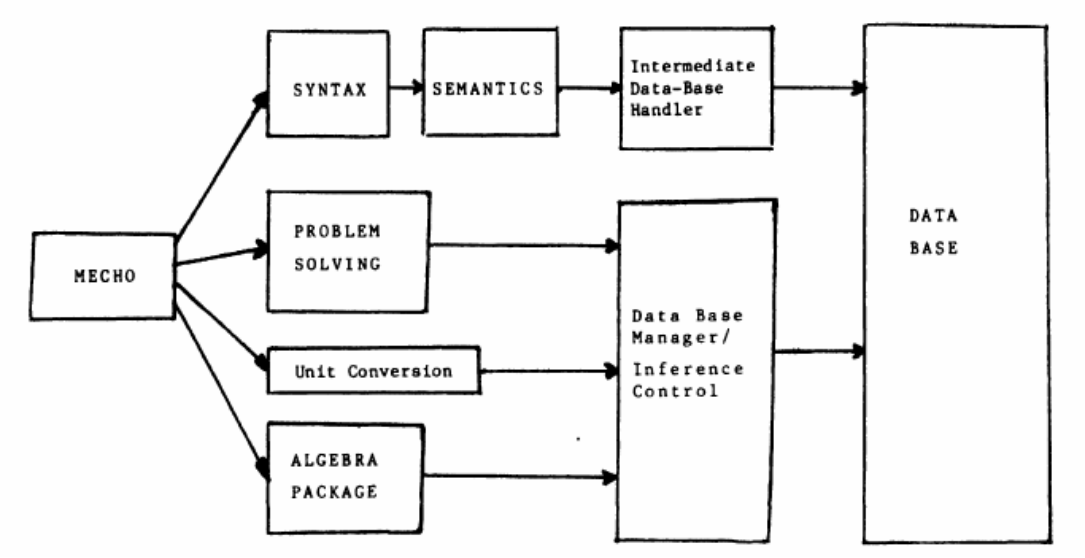
(EXPT HOUR - 2 . ) ) ) 0 . 5 ) ))

THE ANSWER IS)

0.

### 3.3. MECHO

MECHO, mekanik problemlerini çözen, PROLOG dili ile geliştirilmiş bir programdır. [14] MECHO, mekanikle ilgili özellikle makara ve hareket problemlerini başarıyla çözmektedir. Verilen problem içindeki cümleler sözdizimsel ayrıştırıcı (syntactic parser) tarafından PROLOG fact'lerine (gerçek) dönüştürülmektedir. Daha sonra, sözdizimsel analiz sonucu oluşturulan ayrıştırma ağaçlarından anlamsal olarak doğru ayrıştırma ağacının tespit edilmesi için anlamsal analiz yapılmaktadır. Verilen problem içindeki mekanik ile ilgili kelimeler (örneğin "makara") bulunmakta ve bu bulunan kelimeler ipucu kabul edilerek bu kelimelerle ilgili diğer kelimeler de sözlükten bulunmaktadır. Cümleler denklemler halinde ifade edilmekte ve bu denklemler "algebra package" adı verilen modül ile çözümlenmektedir. Şekil 3.1'de MECHO'nun modülleri ve bu modüller arasındaki ilişkiler gösterilmektedir.




**Şekil 3.1:** MECHO Modülleri [14]

### 3.4. ALİ

ALİ (Aritmetikçi Lisan İşleyici) Cem Say tarafından geliştirilen Türkçe aritmetik problemleri çözen ilk programdır. [15] 23 tane yazı içeren Türkçe aritmetik problem, ilköğretim 3.sınıf Türkçe matematik kitaplarından seçilerek, üzerlerinde biraz değişiklik yapılmıştır. Yapılan değişiklikler, gramer olarak yanlış olan cümlelerin grameri doğru cümleler haline çevrilmesidir. ALİ, 4 işlem (toplama, çıkarma, bölme, çarpma) ve üçgen, kare gibi geometrik şekillerle ilgili matematik problemlerini çözme kapasitesine sahiptir. [15]

ALİ, Turbo Prolog dili kullanılarak geliştirilmiştir. Program çalışmaya başladığında kullanıcı soru cümlesini programın problem alanına ortografik kurallara (imla kurallarına) uyarak girer ve ALİ, problemin ortografik kurallara uyup uymadığı kontrolünü yaparak problem cümlesini biçimbirimsel (morfolojik) analiz seviyesine gönderir. [15]

ALİ'nin çalışma mantığı aşağıdaki basit örnek üzerinde açıklanacaktır.

"Bir yılda kaç hafta bulunur?"

### 3.4.1. Morfolojik Analiz Seviyesi

Türkçe, zengin bir morfolojik yapıya sahip olduğu için, ALİ verilen problemin her kelimesi için morfolojik analiz yapmaktadır. Kelimenin ait olduğu sınıfın tanımlanması (POS Etiketlemesi) için Türkçe- Azerice Makine Çevirisi Projesi [16], üzerinde küçük değişiklikler yapılarak kullanılmıştır. POS etiketleyicisinin bazı eksikleri bulunduğundan morfolojik analizin doğru şekilde yapılabilmesi için "morfolojik son çözücü" (morphological postprocessor) tanımlanmıştır. Örneğin "15'ten" kelimesinin sahip olduğu eklerin tespit edilebilmesi için "15" kelimesi yazıya ("on beş" şeklinde) çevrilmiş ve aldığı ekin "ismin –den hali" olduğu belirlenmiştir. [15]

```
[[<"bir",[<adjective;root(adjective("bir"))>]>,
<"yılda",[<noun;root(noun("yıl")),case(locative)>]>,
<"kaç",[<adjective;root(adjective("kaç"))>,
<verb;root(verb("kaç"))>]>,
<"hafta",[<noun;root(noun("hafta"))>]>,
<"bulunur",[<verb;root(verb("bul")),voice(passive),tense(aorist)>]>
]]
```

**Şekil 3.2**: Örnek problemin morfolojik analizi

Şekil 3.2'de örnek problem için morfolojik analiz sonucu gösterilmektedir. Şekil 3.2'de görüldüğü gibi tüm kelimelerin morfolojik analizleri yapılmış ve her kelime için kelimenin türü, kökü ve (varsa) ekleri sırayla belirtilmiştir. "kaç" kelimesi için 2 tane tür bulunmuş ve bunlarda alt alta sıralanmıştır. Yukarıdaki örnek için bulunan liste bir sonraki aşama olan sözdizimsel (sintaks) seviyeye gönderilmiştir. [15]

### 3.4.2. Sözdizimsel Analiz Seviyesi

Sözdizimsel analiz seviyesi, morfolojik analiz seviyesi sonucu üretilen listeleri alır ve her cümle için gramere uygunluğunu tespit etmek amacıyla ayrıştırma ağaçlarını hesaplar. MECHO programının tam aksine ALİ, sözdizimsel analiz seviyesinde anlamsal bilgilerin kullanılmasını minimum seviyede tutmuştur. Anlamsal bilgilerin kullanılmasına örnek olarak "5 metre ip" cümlesindeki ikinci kelimenin bir birim olup olmadığının kontrolü verilebilir. Kontrol sonucu ikinci kelimenin bir birim

olduğu saptanarak, bu tip kelimeler için "Ölçü + Birim + İsim Öbeği" (Measurement + Unit + Noun Phrase) yapısı belirlenmiştir. [15]

23 problem kümesi için Şekil 3.3'deki gramer yapısı ve ayrıştırma algoritması için de PROLOG dilinde gerçeklemesi kolay olan yukarıdan-aşağıya, soldan-sağa ayrıştırma algoritmaları kullanılmıştır. [15]

$$
\begin{aligned}
&S \rightarrow NP\ V\\
&S \rightarrow NP\ NP\ V\\
&S \rightarrow NP\ NP\ NP\ V\\
&S \rightarrow AdvP\ AdvP\ V\\
&S \rightarrow AdvP\ AdvP\ AdvP\ V\\
&S \rightarrow NP\ NP\ AdvP\ V\\
&S \rightarrow AdvP\ NP\ AdvP\ V\\
&S \rightarrow S_{ise}\ S\ \{S_{ise}\ \text{is of the form "S + ise", i.e. "If S"}\}
\end{aligned}
$$

**Şekil 3.3:** Örnek problem için gramer kuralları [15]

Örnek problem tek bir cümle içerdiği için morfolojik analiz sonucu bulunan liste tüm kelimelerin analiz sonuçlarını içermektedir. Cümle içindeki "kaç" kelimesi 2 anlam içerdiği için bu cümle için Şekil 3.4'de gösterildiği gibi 2 tane aday dizisi bulunmaktadır. [15]

```
[<"bir",<adjective;root(adjective("bir"))>>,
  <"yılda",<noun;root(noun("yıl")),case(locative)>>,
  <"kaç",<adjective;root(adjective("kaç"))>>,
  <"hafta",<noun;root(noun("hafta"))>>,
  <"bulunur",<verb;root(verb("bul")),voice(passive),tense(aorist)>>
  ]
[<"bir",<adjective;root(adjective("bir"))>>,
  <"yılda",<noun;root(noun("yıl")),case(locative)>>,
  <"kaç",<verb;root(verb("kaç"))>>,
  <"hafta",<noun;root(noun("hafta"))>>,
  <"bulunur",<verb;root(verb("bul")),voice(passive),tense(aorist)>>
  ]
```

**Şekil 3.4:** Örnek problem için aday diziler [15]

Aday dizilerinin gramere uygunlukları kontrol edilerek sözdizimsel analiz sonuçlanır ve elde edilen ağaç (Şekil 3.5) anlamsal analiz seviyesine gönderilir .

```
pa(1,s([np([adj([adj("bir")]),np([noun([noun("yıl")])],loc,sg3,
  poss(frpos))],loc,sg3,poss(frpos)),np([adj([adj("kaç")]),
  np([noun([noun("hafta")])],nom,sg3,poss(frpos))],nom,sg3,
  poss(frpos)),vp([verb([verb("bul")],frcs,frpos,frmo,aorist,
  frct,positive,sg3,passive)],aorist,frct,positive,sg3,passive)])).
```

**Şekil 3.5:** Örnek problemin sözdizimsel analiz sonucu [15]

### 3.4.3. Anlamsal Analiz Seviyesi

Anlamsal analiz seviyesi, problemin metin halini ve sözdizimsel analiz sonucunu giriş olarak alır. Bu seviye bütün cümleleri S olarak kabul eder ve eğer S bir ayrıştırmaya sahipse S cümlesini kod içinde tanımlı anlamsal şablonlardan biriyle eşleştirmeye çalışır. Ayrıştırma dizisi eğer herhangi bir şablonla eşleşmişşe, problemin çözümü bulunur veya problem program içinde tanımlı bazı değişikliklere uğratılır. Örneğin "97 simitin 68 tanesi satılırsa kaç simit kalır?" cümlesi şart içermeyen iki cümle haline dönüştürülür. Dönüştürme sonucu oluşan cümleler; "97 simitin 68 tanesi satılır. Kaç simit kalır ?" şeklindedir. [15]

Örnek problem sonucunda oluşan ayrıştırma ağacı anlamsal analiz şablonlarından 5. şablon ile eşleşmekte ve bu şablona göre anlamsal çözücü cümle içinde yalın halde ve ismin – de halini içeren iki tane isim öbeği aramaktadır. Bu doğrultuda "yılda" ve "hafta" kelimeleri bulunmakta ve anlamsal analiz seviyesi tamamlanmaktadır. [15]

### 3.4.4. Örnek Problemler

Aşağıda ALİ'nin çözebildiği örnek problemlerden birkaçı verilmektedir. [15]

1) 36 kg elmanın dörtte biri kaç kilogram eder ?

{Semantic template used: Type 4}

36 kg elmanın dörtte biri 9 kilogram eder.

2) Bir fabrikada 217 işçi vardı. 15 işçi çıktı. 7 işçi emekli oldu. Fabrikada kaç işçi kalır?

{Semantic template used: Type 1, Type 2, Type 2, Type 4}

Fabrikada 195 işçi kalır.

3) Çevre uzunluğu 40 cm olan bir karenin bir kenarının uzunluğu kaç cm olur ?

{Semantic template used: Type 4}

Çevre uzunluğu 40 cm olan bir karenin bir kenarının uzunluğu 10 cm olur.

# 4. UYGULAMAYA YÖNELİK TEMEL BİLGİLER

## 4.1. Türkçe Dilbilgisi Temel Kuralları

Yeryüzünde ortalama 4000 dil konuşulmaktadır. Bu diller arasında en çok ve en yaygın konuşulan ilk yedi dil Çince, Hintçe, İspanyolca, İngilizce, Fransızca, Rusça ve Türkçedir. Bu sıralama, Türkçenin dünya dilleri arasında ne kadar büyük bir yeri olduğunu göstermesi bakımından son derece önemlidir. [17]

Türkçenin diğer önemli özelliklerinden biri de çok eski yıllardan beri devlet dili oluşudur. Yukarıda sözü edilen 4000 dilden, yalnızca 118'i devlet dili olabilmiştir. Oysa M.Ö. III. yüzyılda, Hunların tarih sahnesine çıktıkları dönemde Hun Türkçesi devlet dili olarak kullanılmaktaydı. Bu olay da Türkçenin kullanılış tarihinin en az 3000 yıl kadar gerilere gittiğini göstermekte ve Türkçenin yüzyıllar boyu kullanılan hem çok eski ve köklü, hem de çok konuşulan yaygın bir dil olduğunu göstermektedir. [17]

Türkçe bağlantılı bir dil olduğundan köklere ekler getirilerek binlerce yeni sözcük elde edilebilir. Bağlantılı dil, çekim ve türetim sırasında sözcük köklerinin değişmediği dillere denilmektedir. Bu özellikleri nedeniyle Türkçe, geniş anlatım olanaklarına ve yeni sözcük türetme gücüne sahip bir dildir. [17]

Türkçe Ural-Altay dil ailesinin Altay koluna bağlı bir dildir.

### 4.1.1. İsim Tamlamaları

İki ya da daha çok isimden kurulmuş tamlamalara isim tamlamaları denilmektedir. Bir başka ifadeyle isim tamlamaları; birden fazla ismin iyelik, tür, nitelik vb. ilgilerle oluşturduğu kelime grublarıdır. İsim tamlamalarına, "Akdeniz akşamları", "benim ödevim", "defne yaprağı", "gül kokusu", "hesap makinesi", "insanlık hâli", "masa örtüsü", "öğrencilerin sınavı", "öğretmenin kalemi" gibi örnekler verilebilir. Grubu oluşturan kelimelerden birincisi tamlayan (belirten), ikincisi tamlanan (belirtilen)dır. [18]

İsim tamlamaları dörde ayrılır :

1) Belirtili isim tamlaması
2) Belirtisiz isim tamlaması
3) Takısız isim tamlaması
4) Zincirleme isim tamlaması

#### 4.1.1.1. Belirtili İsim Tamlaması

Tamlayanın ilgi hali ekini (-ın, -in, -un, -ün), tamlananın ise iyelik ekini (-ı, -i, -u, -ü; -sı, -si, -su, sü) aldığı isim tamlamalarına belirtili isim tamlaması denilmektedir.

Aşağıda belirtili isim tamlamasına örnekler verilmiştir. [17]

kız + çanta → kız**ın** çanta**sı**

ev + kapı → ev**in** kapı**sı**

okul + müdür → okul**un** müdür**ü**

Türkçede iki ünlü yanyana gelmediğinden dolayı ünlü ile biten kelimelerden sonra isim tamlaması varsa araya **n** ünsüzü eklenir.

öğrenci + kalem → öğrenci**nin** kalemi

#### 4.1.1.2. Belirtisiz İsim Tamlaması

Tamlayanın ek almadığı, tamlayanın ise iyelik eki aldığı tamlamalara belirtisiz isim tamlaması denilmektedir.

“araba lastiği”, “otobüs şoförü”, “pencere camı”, “okul bahçesi”, “müze müdürü” belirtisiz isim tamlamalarına örnek olarak verilebilir. [17]

#### 4.1.1.3. Takısız İsim Tamlaması

Ek almadan kurulan isim tamlamalarına takısız isim tamlaması denilmektedir. Takısız isim tamlamalarında tamlayan ve tamlanan isim ek almaz. Takısız isim tamlaması anlam yönüyle ikiye ayrılabilir:

- Tamlayan, tamlananın neden yapıldığını bildirir :

ipek gömlek, altın yüzük, bakır kazan, keten pantolon

- Tamlayan, tamlananın özelliklerini, neye benzediğini, unvanını, sayısını belirtir :

aslan çocuk, kurt adam, Doktor Mehmet Gürsel, beş kilo elma [17]

#### 4.1.1.4. Zincirleme İsim Tamlaması

İsim tamlamasında tamlayanlar ve tamlananlar birden fazla olursa bu tamlamalara zincirleme isim tamlaması denilmektedir. [17]

Tablo 4.1'de zincirleme isim tamlamalarına örnekler verilmektedir. [17]

**Tablo 4.1:** Zincirleme isim tamlaması örnekleri

| **İsim Tamlaması** | **Zincirleme İsim Tamlaması** |
|---|---|
| elbise dolabı | elbise dolabının kapısı |
| müze müdürü | müze müdürünün şapkası |
| Orhan'ın evi | Orhan'ın evinin kapısının kolu |
| Türk Hava Yolları | Türk Hava Yollarının uçakları |

### 4.1.2. Türkçe Ortografik Kurallar

İki morfem birleştirilirken uyulması gereken imla kurallarına ortografik kurallar denilmektedir. [7]

Türkçede sesler belli bazı kurallara göre birleşerek heceleri, ekleri ve kelimeleri oluşturmaktadır. Eklerin kelimelerle birleşmeleri de ses kurallarına göredir. Türkçede bütün ünlülerde ve ünsüzlerde görülen bu ses kurallarının en önemlileri aşağıda sıralanmaktadır. [17]

#### 4.1.2.1. Büyük Ünlü Uyumu

Türkçe bir sözcükte kalın ünlülerden (a, ı, o, u) sonra kalın ünlülerin, ince ünlülerden (e, i, ö, ü) sonra ince ünlülerin gelmesine büyük ünlü uyumu denilmektedir. [17]

| | | |
|---|---|---|
| ayak | boyun | burun |
| gelincik | gözlük | yüzük |

#### 4.1.2.2. Küçük Ünlü Uyumu

Türkçe'de bir kelimede düz ünlüden (a, e, ı, i) sonra düz ünlünün, yuvarlak ünlüden (o, ö, u, ü) sonra dar yuvarlak (u, ü) veya geniş düz (a, e) ünlünün gelmesine küçük ünlü uyumu denilmektedir. [17]

Tablo 4.2'de örnek kelimeler gösterilmektedir.

**Tablo 4.2:** Küçük ünlü uyumuna örnek kelimeler

| | | | |
|---|---|---|---|
| a → a, ı | anl**a**m, aç**ı**k | o → a, u | oyn**a**m**a**k, ok**u**l |
| e → e,i | el**e**m**e**k, eğ**i**t**i**m | ö → e, ü | öğ**e**tm**e**n, öğ**ü**t |
| ı → a, ı | ısm**a**rl**a**m**a**k, ıs**ı**nmak | u → a, u | uy**a**nm**a**k, uyk**u** |
| i → e, i | izl**e**m**e**k, iz**i**n | ü → e, ü | ür**e**tmek, üt**ü** |

#### 4.1.2.3. Ünlü Düşmesi

Sözcüklere birtakım çekim ve yapım eklerinin gelmesi veya birleşik sözcüklerin yapımı sırasında bir ünlünün kaybolmasına ünlü düşmesi denilmektedir. [17]

| | |
|---|---|
| ağız → ağzım | omuz → omzum |
| alın → alnın | beyin → beynin |
| burun → burnu | göğüs → göğsüm |

# 5. ARİTMETİK PROBLEMLERİN ÇÖZÜMLENMESİ VE SİMGELENMESİ

## 5.1. Temel Adımlar

Türkçe aritmetik problemlerin çözümlenmesi ve simgelenmesi için geliştirilen uygulamada izlenen temel adımlar şöyledir:

I. Kullanıcının girdiği soru içindeki tüm kelimeler için biçimbirimsel analiz yapılması

II. Tüm kelimelerin aldıkları eklerin belirlenmesi

III. Soru kalıbının belirlenmesi

IV. Sorunun çözülmesi

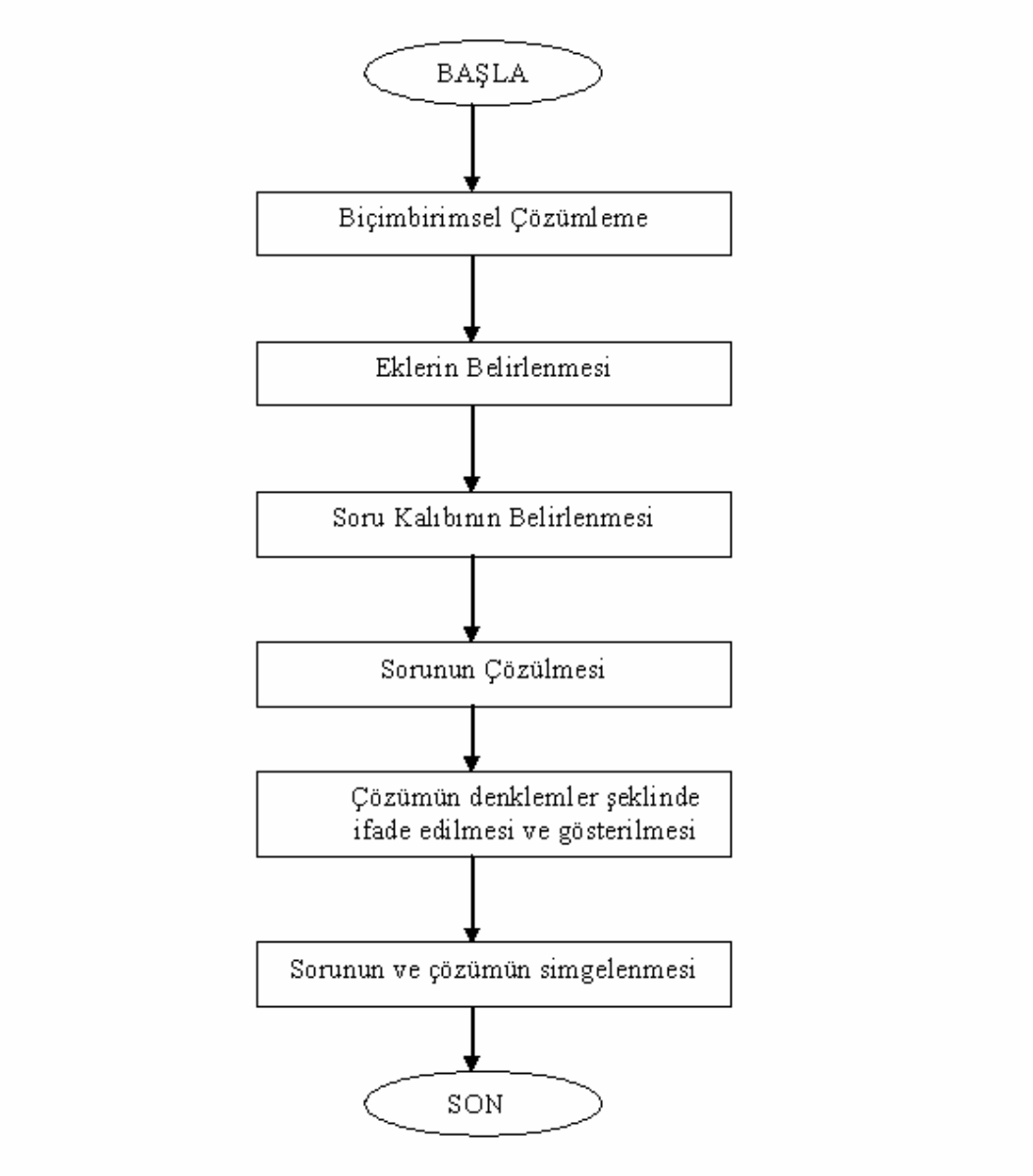
V. Çözümün denklemler şeklinde ifade edilmesi ve gösterilmesi

VI. Sorunun ve çözümün simgelenmesi

Şekil 5.1'de geliştirilen sistemin genel akış diyagramı verilmektedir.

## 5.2. Biçimbirimsel Analiz (Morfolojik Analiz) Modülü

Geliştirilen uygulamanın ilk adımı olan biçimbirimsel analiz modülünde, kullanıcı tarafından girilen problem içindeki tüm kelimelerin biçimbirimsel analizi yapılmaktadır. Biçimbirimsel analizden maksat, tüm kelimelerin köklerinin ve aldıkları eklerin belirlenmesidir. Biçimbirimsel analiz için Zemberek projesinden faydalanılmıştır.




**Şekil 5.1:** Uygulamanın modülleri ve bu modüllerin işleyiş sırası

### 5.2.1. Zemberek

#### 5.2.1.1. Zemberek Nedir ?

Zemberek, Türkçe diline ilişkin çeşitli bilgi işlem problemlerinin çözümlenmesi için açık kodlu, platform bağımsız bir kütüphane oluşturulması amacı ile geliştirilen bir projedir. [19] Zemberek diğer bir ifadeyle Türkçe doğal dil işleme kütüphanesidir. Zemberek ile yapılabilen doğal dil işleme işlemleri aşağıda sıralanmaktadır:

- Kelime denetleme
- Kelime çözümleme
- Kelime önerme
- Türkçe karakter kullanılmadan yazılan yazıların dönüştürülmesi (ASCII → Türkçe dönüşümü),
- Heceleme
- Kodlama hatası yüzünden bozulmuş metinlerin temizlenmesi
- Türkçe metin tesipiti
- Kelime ayrıştırma
- Türkçe → ASCII dönüşümü

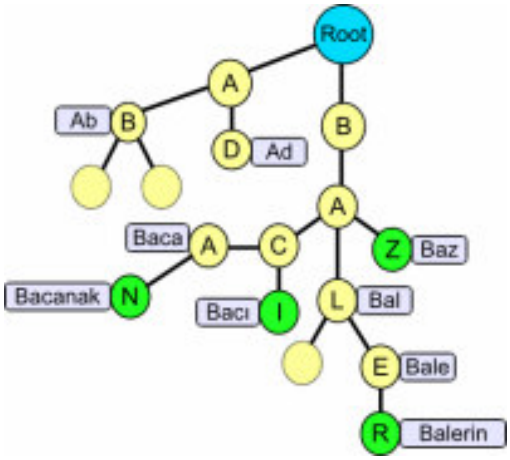
#### 5.2.1.2. Zemberek Nasıl Çalışır ?

Zemberek çok basit bir sistemle verilen bir kelimeyi morfolojik olarak incelemektedir. Önce verilen kelimenin kökü olabilecek adaylar belirlenmekte, sonra da olabilecek ekler uygun sırayla bu köke eklenmeye çalışılmaktadır. Eğer bu işlem sırasında girişteki kelimenin aynısını elde edilebilmişse, o zaman uygun kök ve ekler de bulunmuş demektir ve kelime Türkçedir, eğer kök adaylarının hiçbirinden sonuç elde edilememişse o zaman kelime Türkçe değildir. [20] Aşağıda bu işlemin ilk adımı olan kök adaylarının bulunması işlemi detaylı olarak açıklanmaktadır.

Kök adaylarının bulunabilmesi için Türkçe'deki tüm kök kelimelerinin bulunması gerekmektedir. Zemberek Türkiye Türkçesi için yaklaşık 30.000 kök içeren bir kılavuzu da beraberinde taşımaktadır. Bu kılavuz her kök tipine ve özel durumlarına göre etiketlenmiş şekilde bulunur. Diğer Türk dilleri için yapılan gerçeklemeler de benzeri bir kök sözlüğünü taşımalıdırlar. Burada bahsi geçen özel durumlar sondaki sert ünsüz harfin yumuşaması (sağlık → sağlığa) veya bazı durumlarda sondan bir önceki sesli harfin düşmesi (burun → burnu) gibi halleri içermektedir. Kökler için kullanılan etiketler de dile göre değişiklik gösterebilir. Aşağıda Azeri Türkçesi için yazılmış kök sözlüğünün küçük bir bölümü görülmektedir.

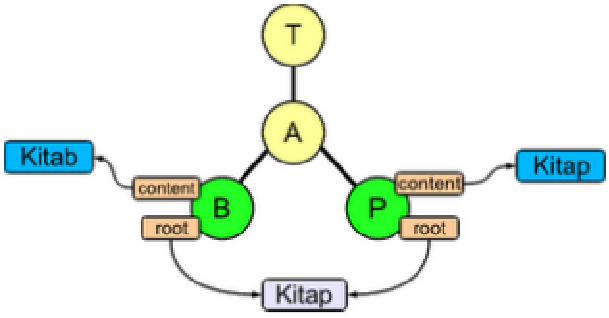
su AD
sağlıq AD YUM
al EY
gel EY
tuz AD
istiot AD
et AD
balıg AD
bir RA
iki RA
dörd RA
dünen ZAMAN
sabah ZAMAN

Verilen bir kelimenin kök adaylarının bulunabilmesi için Zemberek bu kökleri özel bir ağaca yerleştirir. Bu özel ağaç sayesinde adayların belirlenmesi son derece hızlı şekilde yapılabilmektedir. Bu ağaçta kökler içeriklerine göre yerleşmektedirler. Örnek olarak Şekil 5.2'de "baz" kökü sırasıyla B-A-Z ile etiketlenmiş düğümlerin en sonuncusuna bağlanmış şekilde durmaktadır. Burada dikkati çeken bir nokta da uzun köklerin gereksiz fazladan düğüm oluşturmayacak şekilde ağaca bağlanması ve bellekten tasarruf edilmesidir. Yani "Balerin" kökü B-A-L-E düğümlerinden sonra gelen R düğümüne bağlanmıştır. [20]




**Şekil 5.2:**Örnek kelime için kök adaylarının bulunması

Kök ağacı kök düğümleri ve bu düğümlerin bağlantılarından oluşmaktadır. "Kitap" kelimesi gibi bir ek geldiği zaman yumuşama olasılığı olan köklerin değişmiş halleri de kök ağacına eklenmektedir. Ancak bu değişmiş haller de orijinal kökü işaret etmektedirler. [20] Şekil 5.3'de yumuşama örneği gösterilmektedir.

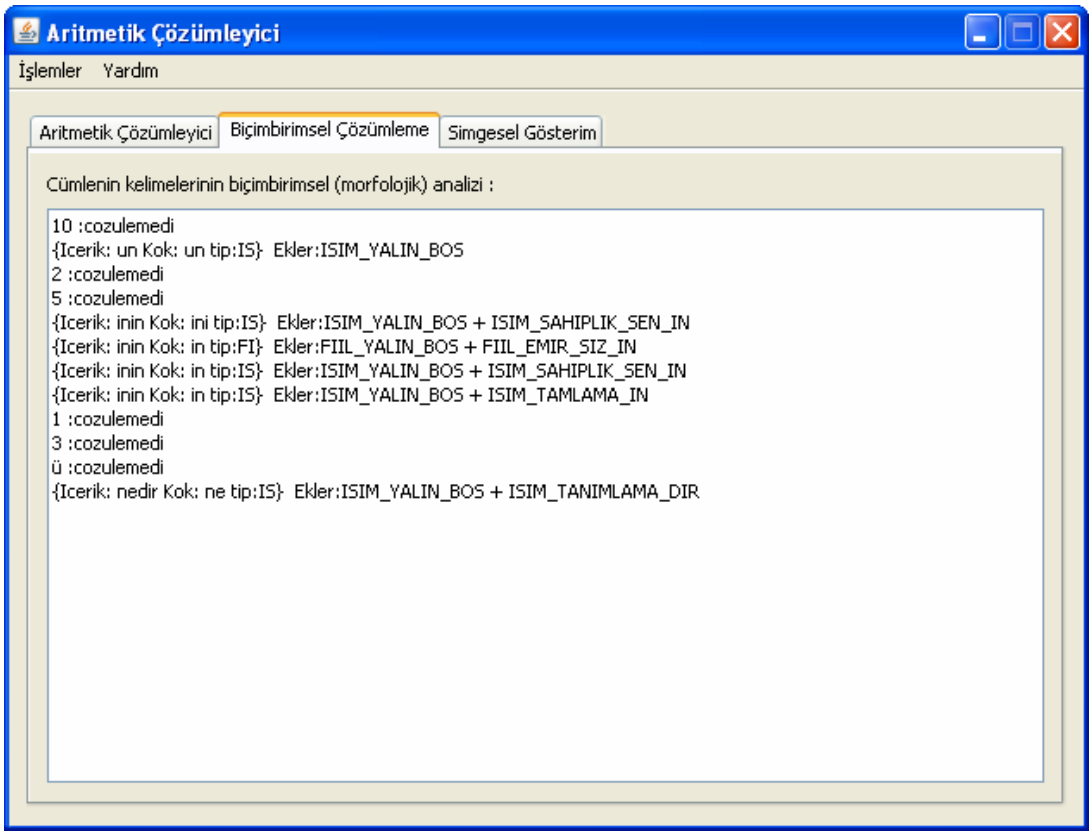



**Şekil 5.3:**Yumuşama örneği

### 5.2.2. Tez Kapsamında Zemberek Kullanımı

Geliştirilen uygulamada Zemberek kütüphanesinin kelime çözümleme metotlarından faydalanılmıştır. Cümle halinde verilen aritmetik problemin bütün kelimeleri sırasıyla Zemberek kütüphanesine gönderilmiş, Zemberek'in kelime çözümlemesi yapması sağlanmış ve kelime çözümlemeleri Zemberek'ten alınmıştır. Aşağıdaki örnekler aritmetik problemler ve bu problemlerde geçen kelimelerin Zemberek sonucu oluşan çözümlemelerini göstermektedir.

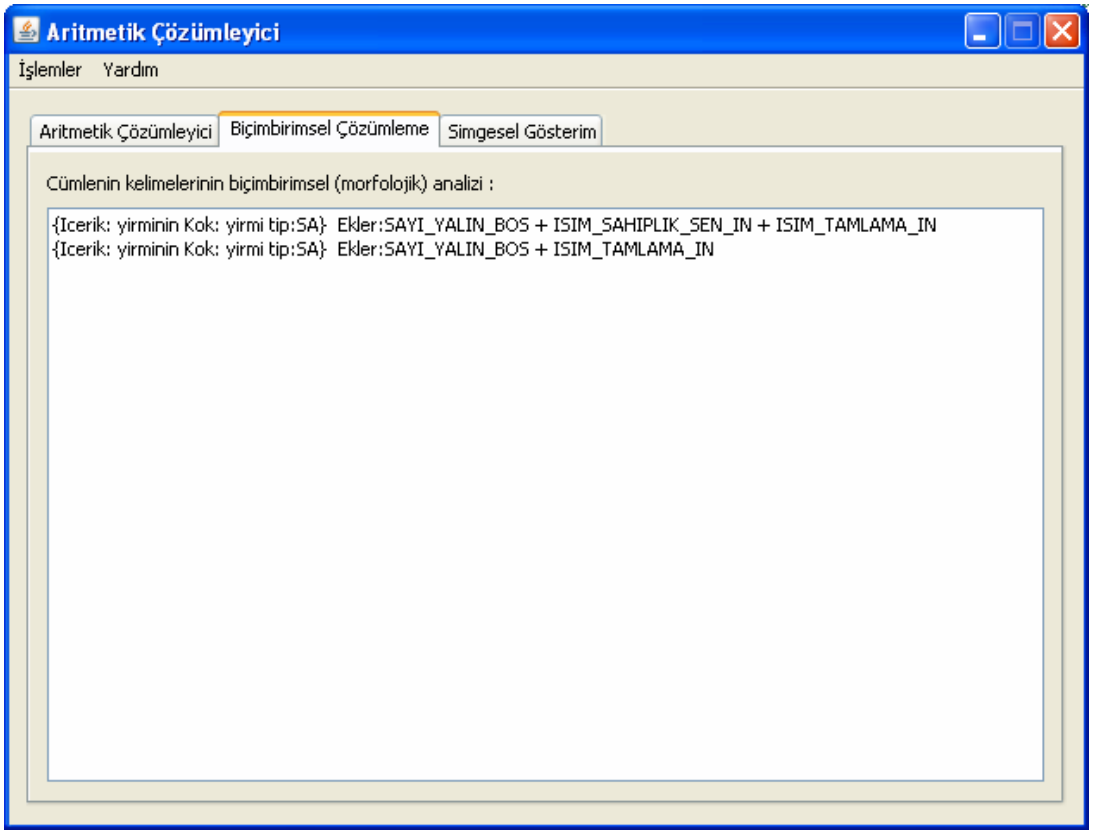
“10’un 2/5’inin 1/3’ü nedir?” örnek problemi için gerçekleştirilen biçimbirimsel analiz Şekil 5.4’de gösterilmektedir.




**Şekil 5.4:** Örnek problem için morfolojik analiz sonucu

Şekil 5.4’te de gösterildiği gibi çözülemeyen kelimeler için “çözülemedi” ifadesi yazılmıştır. Sayılar ve rakamlar Zemberek kütüphanesi tarafından sayı olarak girildiği taktirde çözülemediği için tez kapsamında aritmetik problemin içindeki sayılar ve rakamlar (yukarıdaki örnekteki 10 ve 2/5 sayıları gibi) yazıya çevrilerek Zemberek’e gönderilmiş ve yazı olarak sayının ekleri bulunarak kelime tipi tanımlanmıştır. Örnek olarak,

“10’un” kelimesi içindeki 10 sayısı yazı ile “on” kelimesine çevrilmiş ve “10’un” kelimesi “onun” şeklinde ifade edilerek Zemberek’e gönderilmiştir. Zemberek yazı ile gelen bu sayının kökünü ve eklerini bulur ve elde edilen kök ve ekler ile kelimenin sınıfı (isim tamlamasındaki tamlayan veya tamlanan olup olmadığı) belirlenir. Şekil 5.5’de Zemberek’in yazı ile yazılan sayıları çözümlemesine örnek gösterilmiştir.




**Şekil 5.5: "**yirminin" kelimesinin morfolojik analizi

Şekil 5.5'te de görüldüğü gibi "yirminin" kelimesi tamlayan eki içermektedir. Buradan bu sayının işlem yapılacak sayı olduğu sonucu çıkarılacaktır.

Kesirli sayı içeren aritmetik problemler için kesirin tamamı yazıya çevrilmez, sadece payda değeri yazıya çevrilir, ekleri eklenir ve Zemberek'e gönderilir. Zemberek sonucu alınan çözümleme o kesirin kelime sınıfını tanımlama da yeterli olacaktır. Örnek olarak "10/7'nin" kelimesinin paydadaki 7 değeri yazı ile "yedi" haline dönüştürülür, sahip olduğu ekler kelimenin sonuna eklenir, yani "yedinin" kelimesi oluşturulur ve bu kelimenin Zemberek'ten çözümlemesi alınır.

### 5.3. Eklerin Belirlenmesi Modülü

İlk modül olan biçimbirimsel analiz modülünden sonra, tüm kelimelerin kökleri ve aldıkları ekler belirlenmiş olur. Eklerin belirlenmesi modülünde, tüm kelimelerin aldıkları ekler kontrol edilerek cümle içindeki isim tamlamaları belirlenir. Bu işlemden sonra aşağıdaki gibi kelimeleri tanımlama işlemi yapılır:

- Eğer bir kelime tamlayan ekine sahipse, o kelime tamlayan kelimelerin tutulduğu diziye eklenir.
- Eğer bir kelime tamlanan ekine sahipse, o kelime tamlanan kelimelerin tutulduğu diziye eklenir.
- Eğer bir kelime tanımlama ekine sahipse, o kelime tanımlama eki içeren kelimelerin tutulduğu diziye eklenir.
- Eğer bir kelime tamlanan ve tanımlama eklerine sahipse, o kelime tamlanan ve tanımlama eki içeren kelimelerin tutulduğu diziye eklenir.

Bu kelimeler ve bu kelimelerin aldıkları eklerin tutulduğu diziler soru kalıbının belirlenmesi ve sorunun çözümlenmesi aşamalarında kullanılmak üzere saklanır.

Aşağıda bazı örnek cümleler ve bu cümle içindeki kelimelerin aldıkları eklerin sınıflandırılmasına örnekler verilmiştir.

**Örnek 1 :** 10/7'nin 2/5'inin 1/3'ü nedir?

Bu cümlenin biçimbirimsel analiz ve eklerin belirlenmesi modülü sonucu aldıkları ekler şu şekildedir.

Tamlayan dizisi : 10/7'nin, 2/5'inin

Tamlanan dizisi: 1/3'ü

**Örnek 2 :** 9/2 hangi sayının 3/2'sinin 4/5'idir?

Bu cümlenin biçimbirimsel analiz ve eklerin belirlenmesi modülü sonucu aldıkları ekler şu şekildedir.

Tamlayan dizisi : sayının, 3/2'sinin

Tamlanan ve tanımlama dizisi : 4/5'idir

**Örnek 3 :** Hangi sayının 3/2'sinin 5/8'i 9'dur?

Bu cümlenin biçimbirimsel analiz ve eklerin belirlenmesi modülü sonucu aldıkları ekler şu şekildedir.

Tamlayan dizisi : sayının, 3/2'sinin

Tamlanan dizisi : 5/8'i

Tanımlama dizisi : 9'dur

**Örnek 4 :** 10'un 3/5'inin 5 eksiğinin 2/3'ünün 6 fazlası nedir?

Bu cümlenin biçimbirimsel analiz ve eklerin belirlenmesi modülü sonucu aldıkları ekler şu şekildedir.

Tamlayan dizisi : 10'un, 3/5'inin, eksiğinin, 2/3'ünün

Tamlanan dizisi : fazlası

### 5.4. Soru Kalıbının Belirlenmesi Modülü

Soru kalıbının belirlenmesi modülünde girilen problemin hangi tip bir aritmetik problem olduğu anlaşılmaktadır. Bu modül, biçimbirimsel analiz ve eklerin belirlenmesi modüllerinden sonra oluşan bilgilere bağlı olarak gerçekleştirilmektedir. Geliştirilen uygulama kapsamında soru kalıplarının belirlenmesi için veri madenciliği sınıflandırma yöntemlerinden karar ağaçları (decision trees) kullanılmasına karar verilmiştir. Bundan sonraki bölümde karar ağaçlarının yapısı, niçin karar ağaçları kullanıldığı ve uygulama kapsamında geliştirilen karar ağaçlarından bahsedilecektir.

#### 5.4.1. Karar Ağaçları

Günümüzde bilgisayar sistemleri her geçen gün ucuzlamakta ve güçleri de artmaktadır. Bilgisayar sistemlerinin bu gelişmeyle birlikte kullanımı da bu ölçüde yaygınlaşmaktadır. Bu yaygınlaşmayla birlikte işletmelerde üretilen sayısal bilgi miktarının arttığı buna paralel veri tabanlarının daha fazla veriyi saklayabilecek boyutlara ulaştığı ve bilgisayar sistemlerindeki gelişme ile veriye ulaşmanın kolaylaştığı görülmektedir. Bu sayede doğru ve daha detaylı bilgiye ulaşmak mümkün hale gelmiş fakat bu büyük sayısal veri yığınlarının yönetilmesi ve anlamlı hale getirilmesi sorun olarak ortaya çıkmıştır. [21]

Veri madenciliği (data mining), büyük miktarda veri içinden gelecekle ilgili tahmin yapılmasını sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır. [22]

Veri madenciliğinde kullanılan yaklaşımlardan biri de sınıflandırmadır. Sınıflandırmanın amacı bir niteliğin değerini diğer nitelikleri kullanarak belirlemektir. Verinin dağılımına göre bir model bulunup, bu bulunan model ile başarımı belirlendikten sonra niteliğin gelecekteki ya da bilinmeyen değeri tahmin edilmektedir. Sınıflandırma gözetimli (supervised) ve gözetimsiz (unsupervised)

olmak üzere ikiye ayrılmaktadır. Gözetimli sınıflandırmada sınıfların sayısı ve hangi nesnenin hangi sınıfta olduğu bilinmekte iken gözetimsiz sınıflandırmada hangi nesnenin hangi sınıfta olduğu bilinmemektedir. [23]

"Genç kadınlar küçük araba satın alır, yaşlı, zengin erkekler büyük, lüks araba satın alır" cümlesi sınıflandırmaya örnek olarak verilebilir. Buradaki amaç bir malın özellikleri ile müşteri özelliklerini eşlemektir. Böylece bir müşteri için ideal ürün veya bir ürün için ideal müşteri profili çıkarılabilir. Örneğin bir otomobil satıcısı şirket geçmiş müşteri hareketlerinin analizi ile yukarıdaki gibi iki kural bulursa genç kadınların okuduğu bir dergiye reklâm verirken küçük modelinin reklâmını verir. [22]

Kredi başvurusunu değerlendirme, kredi kartı harcamasının sahtekarlık olup olmadığına karar verme, hastalık teşhisi, ses tanıma, karakter tanıma, gazete haberlerini konulara göre ayırma ve kullanıcı davranışlarını belirleme sınıflandırma uygulamalarına örnek olarak verilebilir. [23]

Sınıflandırma işlemi üç aşamadan oluşmaktadır [23] :

1. Model oluşturma
2. Model değerlendirme
3. Modeli kullanma

Model oluşturma aşamasında her nesnenin sınıf etiketi olarak tanımlanan niteliğinin belirlediği bir sınıfta olduğu varsayılmaktadır. Model oluşturmak için kullanılan nesnelerin oluşturduğu veri kümesine öğrenme kümesi denilmektedir. [23]

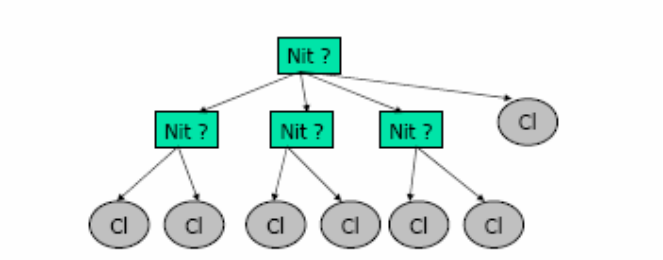
Model değerlendirme aşamasında modelin doğruluğu, sınama (test) kümesi örnekleri kullanılarak belirlenmektedir. Sınıf etiketi bilinen bir sınama kümesi örneği model kullanılarak belirlenen sınıf etiketiyle karşılaştırılmaktadır. Modelin doğruluğu, doğru sınıflandırılmış sınama kümesi örneklerinin toplam sınama kümesi örneklerine oranı olarak belirlenmektedir. [23]

Modeli kullanma aşamasında, model daha önce görülmemiş örnekleri sınıflandırmak için kullanılmaktadır. Örneklerin sınıf etiketleri ve bir niteliğin değeri tahmin edilmektedir. [23]

Sınıflandırmada kullanılan başlıca yöntemler;

- Karar Ağaçları (decision trees)
- Bayes sınıflandırıcılar (bayes classifier)
- Yapay sinir ağları (artificial neural networks )
- k-en yakın komşu yöntemi (k – nearest neighbor method)
- Genetik algoritmalar (genetic algorithms)
- Destek vektör makineleri (support vector machines) olarak sıralanabilir.

Geliştirilen uygulama kapsamında karar ağaçları yöntemi uygulanmıştır. Karar ağaçları genellikle sınıflama amacıyla kullanılan bir veri madenciliği tekniğidir. Karar ağacı, akış diyagramına benzer bir ağaç yapısında olup, her bir dal bir testin sonucunu, yaprak düğümleri ise sınıfları temsil etmektedir. Şekil 5.6'da bir karar ağacı yapısı gösterilmektedir.




**Şekil 5.6:** Karar ağacı yapısı [23]

Karar ağacı kolay anlaşılabilen kurallar yaratabilen, bilgi teknolojileri işlemleri ile kolay entegre olabilen en popüler sınıflama tekniğidir.

Karar ağacı karar düğümleri, dallar ve yapraklardan oluşur. Karar düğümü, gerçekleştirilecek testi belirtir. Bu testin sonucu ağacın veri kaybetmeden dallara ayrılmasına neden olur. Her düğümde test ve dallara ayrılma işlemleri ardışık olarak gerçekleşir ve bu ayrılma işlemi üst seviyedeki ayrımlara bağımlıdır. Ağacın her bir dalı sınıflama işlemini tamamlamaya adaydır. Eğer bir dalın ucunda sınıflama işlemi gerçekleşemiyorsa, bir karar düğümü oluşur. Ancak belirli bir sınıf oluşuyorsa, o dalın sonunda yaprak vardır. Bu yaprak, veri üzerinde belirlenmek istenen sınıflardan biridir. Karar ağacı işlemi kök düğümünden başlar ve yukarıdan aşağıya doğru yaprağa ulaşana dek ardışık düğümleri takip ederek gerçekleşir. [24]

Karar ağacı tekniğini kullanarak verinin sınıflanması iki basamaklı bir işlemdir. İlk basamak öğrenme basamağıdır. Öğrenme basamağında önceden bilinen bir eğitim verisi, model oluşturmak amacıyla sınıflama algoritması tarafından analiz edilir. Öğrenilen model, sınıflama kuralları veya karar ağacı olarak gösterilir. İkinci basamak ise sınıflama basamağıdır. Sınıflama basamağında test verisi, sınıflama kurallarının veya karar ağacının doğruluğunu belirlemek amacıyla kullanılır. Eğer doğruluk kabul edilebilir oranda ise, kurallar yeni verilerin sınıflanması amacıyla kullanılır. [24]

Test verisine uygulanan bir modelin doğruluğu, yaptığı doğru sınıflamanın test verisindeki tüm sınıflara oranıdır. Her test örneğinde bilinen sınıf, model tarafından tahmin edilen sınıf ile karşılaştırılır. Eğer modelin doğruluğu kabul edilebilir bir değer ise model, sınıfı bilinmeyen yeni verileri sınıflama amacıyla kullanılabilir. [24]

Örneğin, bir eğitim verisi incelenerek kredi durumu sınıfını tahmin edecek bir model oluşturulsun. Bu modeli oluşturan bir sınıflama kuralı

IF yas = "41...50" AND gelir = yüksek THEN kredidurumu = mükemmel

şeklinde olsun. Bu kural gereğince yaşı "41...50" kategorisinde olan (yaşı 41 ile 50 arasında olan) ve gelir düzeyi yüksek bir kişinin kredi durumunun mükemmel olduğu görülmektedir. [24]

Belirli bir sınıfın muhtemel üyesi olacak elemanların belirlenmesi, çesitli durumların yüksek, orta, düşük risk grupları gibi çeşitli kategorilere ayrılması, gelecekteki olayların tahmin edilebilmesi için kurallar oluşturulması, sadece belirli alt gruplara özgü olan ilişkilerin tanımlanması, kategorilerin birleştirilmesi gibi alanlarda karar ağaçları kullanılmaktadır. [24]

Karar ağaçları, oluşturulmalarının ucuz, yorumlanmalarının kolay olmaları, anlaşılabilir kurallar oluşturmaya elverişli olmaları ve güvenilirliklerinin iyi olması nedenleriyle sıflandırma yöntemleri içinde yaygın kullanılmaktadır. Bu gibi nedenlerinden dolayı da uygulama kapsamında karar ağaçları kullanılmasına karar verilmiştir.

### 5.4.2. Uygulama Kapsamında Karar Ağaçlarının Kullanılması

Uygulama kapsamında, girilen sorunun hangi kalıba ait olduğunu tespit etmek ve yeni tipteki soruları anlayabilmek amacıyla veri madenciliği sınıflandırma yöntemlerinden karar ağaçları kullanılmıştır.

Uygulama kapsamında geliştirilen karar ağaçları üç aşamadan oluşmaktadır :

1. Öğrenme kümesinin oluşturulması
2. Karar ağacının oluşturulması
3. Karar ağaçları ile soru kalıbının belirlenmesi

#### 5.4.2.1. Öğrenme Kümesinin Oluşturulması

Uygulama kapsamında, soru kalıbının belirlenmesinde model oluşturulması için öğrenme kümesi oluşturulmuştur. Öğrenme kümesinde, sorular içinde geçen bazı ayrıştırıcı özellikler nitelik olarak belirlenmekte ve örnek problemler için bu nitelik değerleri doldurulmaktadır. Aşağıda soru kalıbının belirlenmesi için belirlenen nitelik değerleri sıralanmaktadır.

1. İlk kelime sayı mı? (Tamsayı veya kesirli sayı)
2. İlk kelime “hangi” kelimesi mi?
3. İkinci kelime “hangi” kelimesi mi?
4. Son kelime “soru” kelimesi mi?
5. Son kelime “tanımlama” eki içeriyor mu?
6. Son kelime “tamlanan” + “tanımlama” ekleri içeriyor mu?
7. Soru içindeki tamlayan sayısı > 1 mi?
8. Soru içindeki tamlanan sayısı > 1 mi?
9. Soru “fazla” veya “eksik” problemi mi?
10. Soru bağlaç içeriyor mu?
11. Soru “toplama” problemi mi?
12. Soru “fark” problemi mi?
13. Soru “çarpma” problemi mi?

Yukarıda ifade edilen nitelik değerlerinden, birinci nitelik sorunun ilk kelimesinin tamsayı veya kesirli sayı olup olmadığını; ikinci nitelik sorunun "hangi" kelimesi ile başlayıp başlamadığını; üçüncü nitelik sorunun ikinci kelimesinin "hangi" kelimesi olup olmadığını; dördüncü nitelik sorunun son kelimesinin "soru" kelimesi (nedir, kaçtır, ne olmalıdır gibi) olup olmadığını; beşinci nitelik sorunun son kelimesinin "tanımlama" eki (-dir eki gibi) içerip içermediğini; altıncı nitelik sorunun son kelimesinin "tamlanan" ve "tanımlama" ekleri ("3/2'sidir" kelimesindeki gibi) içerip içermediğini; yedinci nitelik soru içindeki tamlayan kelimelerin sayısının birden fazla olup olmadığını; sekizinci nitelik soru içindeki tamlanan kelimelerin sayısının birden fazla olup olmadığını; dokuzuncu nitelik sorunun fazla veya eksik problemi ("10'un 5 fazlasının 2/3'ü nedir?" örnek problemi gibi) olup olmadığını; onuncu nitelik sorunun bağlaç (ile kelimesi gibi) içerip içermediğini; onbirinci nitelik sorunun toplama (iki problemin toplamını ifade eden tipten) problemi olup olmadığını; onikinci nitelik sorunun fark (iki problemin farkını ifade eden tipten) problemi olup olmadığını; onüçüncü nitelik ise sorunun çarpma (iki problemin çarpımını ifade eden tipten) problemi olup olmadığını ifade etmektedir.

Öğrenme kümesinin oluşturulması için, 75 tane örnek problemin nitelik değerleri belirlenmiş ve "ogrenme.txt" isimli dosyaya yazılmıştır. Tablo 5.1 ve 5.2'de ilk 10 örnek problem ve bu problemler için belirlenen nitelik değerleri gösterilmektedir.

**Tablo 5.1 :** Nitelikler ve 10 tane örnek problem için belirlenmiş nitelik değerleri - 1

| **SORU** | İlk kelime **"sayı"** mı? | İlk kelime **"hangi"** mi? | İkinci kelime **"hangi"** mi? | Son kelime **"soru kelimesi"** mi? | Son kelime **"tanımlama"** eki içeriyor mu? | Son kelime **"tamlanan + tanımlama"** eki içeriyor mu? | Tamlayan sayısı >1 mi? | Tamlanan sayısı >1 mi? |
|---|---|---|---|---|---|---|---|---|
| **1** | H | H | H | E | E | H | E | H |
| **2** | E | H | H | E | E | H | E | H |
| **3** | E | H | E | H | E | E | E | H |
| **4** | H | E | H | H | E | H | E | H |
| **5** | E | H | H | E | E | H | E | E |
| **6** | H | H | H | E | E | H | E | E |
| **7** | H | E | H | H | E | H | E | E |
| **8** | H | H | H | E | E | H | E | H |
| **9** | E | H | H | E | E | H | E | H |
| **10** | E | H | H | E | E | H | E | E |

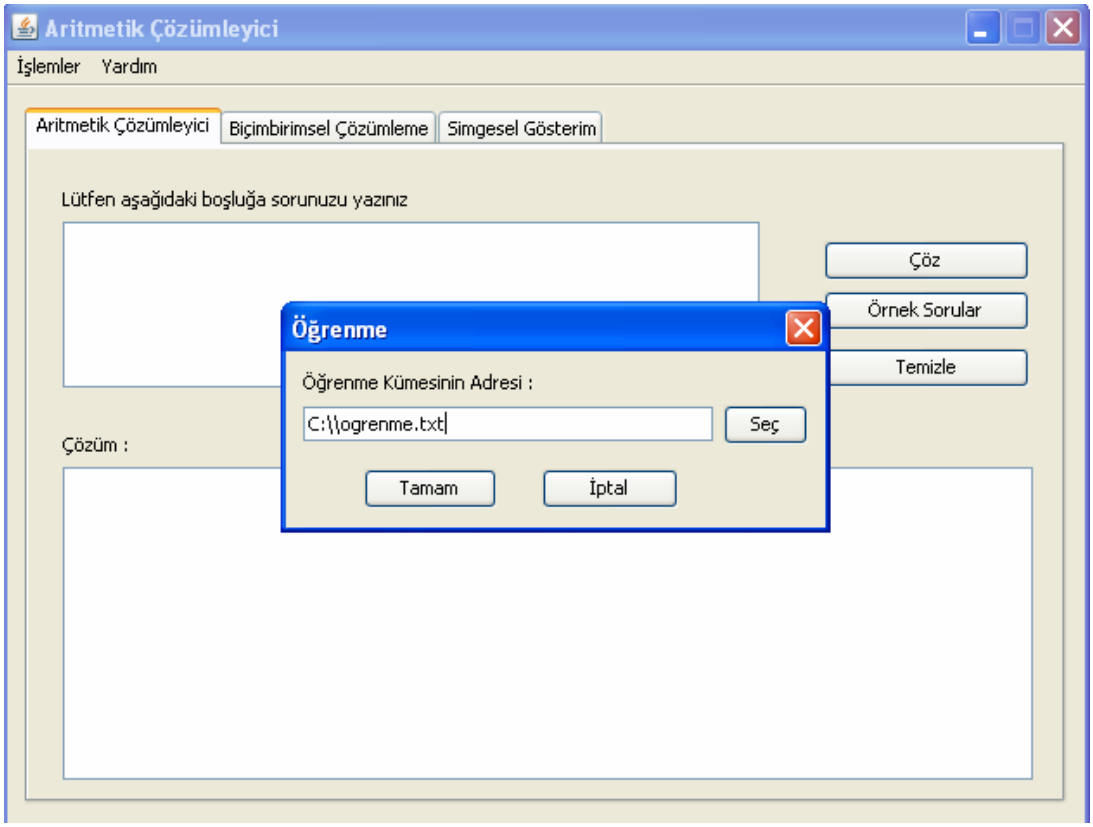
**Tablo 5.2 :** Nitelikler ve 10 tane örnek problem için belirlenmiş nitelik değerleri - 2

| **SORU** | "**Fazla-Eksik**" problemi mi? | "**ile**" kelimesi içeriyor mu? | "**toplama**" problemi mi? (Sondan bir önceki kelime) | "**fark**" problemi mi? (Sondan bir önceki kelime) | "**çarpma**" problemi mi? (Sondan bir önceki kelime) | Sonuç |
|---|---|---|---|---|---|---|
| **1** | H | H | H | H | H | KALIP 1 |
| **2** | H | H | H | H | H | KALIP 2 |
| **3** | H | H | H | H | H | KALIP 5 |
| **4** | H | H | H | H | H | KALIP 6 |
| **5** | H | E | E | H | H | KALIP 8 |
| **6** | H | E | H | E | H | KALIP 8 |
| **7** | H | E | E | H | H | KALIP 9 |
| **8** | E | H | H | H | H | KALIP 1 |
| **9** | E | H | H | H | H | KALIP 2 |
| **10** | E | E | E | H | H | KALIP 8 |

Nitelik değerleri belirlenen 10 tane örnek problem aşağıda sıralanmaktadır:

1) 10'un 2/5'i nedir?
2) 10 sayısının 2/5'i kaçtır?
3) 9 hangi sayının 3/2'sidir?
4) Hangi sayının 3/2'si 9'dur?
5) 20 sayısının 1/2'si ile 25 sayısının 4/5'inin toplamı nedir?
6) 20'nin 1/2'si ile 30'un 2/3'ünün farkı nedir?
7) Hangi sayının 1/2'si ile 4/5'inin toplamı 20'dir?
8) 10'un 5 fazlasının 2/3'ü nedir?
9) 10 sayısının 5 fazlasının 2/3'ü nedir?
10) 20 sayısının 1/2'sinin 3 eksiği ile 25 sayısının 4/5'inin 2 fazlasının toplamı nedir?

Uygulama çalışmaya başladığında kullanıcı, kendi oluşturduğu öğrenme kümesini uygulamaya tanıtıp kullanabilirken ayrıca uygulama için oluşturulan öğrenme kümesini de kullanabilmektedir. Kullanıcı "İşlemler" menüsünden "Öğrenme" menüsünü seçerek kendi öğrenme kümesini seçebilirken, yine "İşlemler" menüsünden "Otomatik Öğrenme" menüsünü seçerek uygulamanın kendi öğrenme kümesini de seçebilmektedir. Şekil 5.7'de şekilde bu durum gösterilmektedir.




**Şekil 5.7:** Öğrenme kümesinin seçilmesi

### 5.4.2.2. Karar Ağacının Oluşturulması

Karar ağacı oluşturma işlemi yinelemeli (rekürsif) olarak gerçeklenmektedir. Uygulamaya tanıtılan örnekler kümesi ile kök düğüm giriş olarak alınmakta ve sonlanma koşullarından biri sağlanıncaya kadar, örnekler bölünmeye ve düğümlere dal eklenmeye devam edilmektedir. Dallanma durunca, yaprak düğüme çoğunlukta olan sınıf etiketi atanmaktadır. Ağaçta sonlanma koşulu ise, şu koşullardan biri olabilir;

- Elde kalan tüm örneklerin aynı sınıftan olması

- Aşırı öğrenme koşulunun sağlanması
    - Ağaçta belirlenen maksimum derinliğe erişilmesi veya
    - Ağaçta belirlenen en az sayıda düğüme erişilmesi
- Elde kalan tüm örneklerin, tüm değerlerinin aynı olması ve böylece kazançlarının aynı olması

Karar ağacının oluşturulmasında ilk olarak, elde kalan örnek kümesine ilişkin tüm nitelikler için kazanç hesaplanmakta ve örneklerin hangi niteliğe göre bölünmesinin daha faydalı olacağı bulunmaya çalışılmaktadır.

Kazanç hesaplamak için önce elde kalan örnekler için genel entropi hesaplanmakta daha sonra ise kazanç hesaplanacak niteliğin tüm değerleri bulunmaktadır. Tüm bu değerler için de entropi hesaplanmakta ve niteliğin kazanç değeri bulunmaktadır. Entropi ve Kazanç formülleri aşağıda gösterilmektedir:

$$\text{Entropi (A)} = - \sum_{i=1}^{Etiketler} p_i \log(p_i) \qquad \textbf{(5.1)}$$

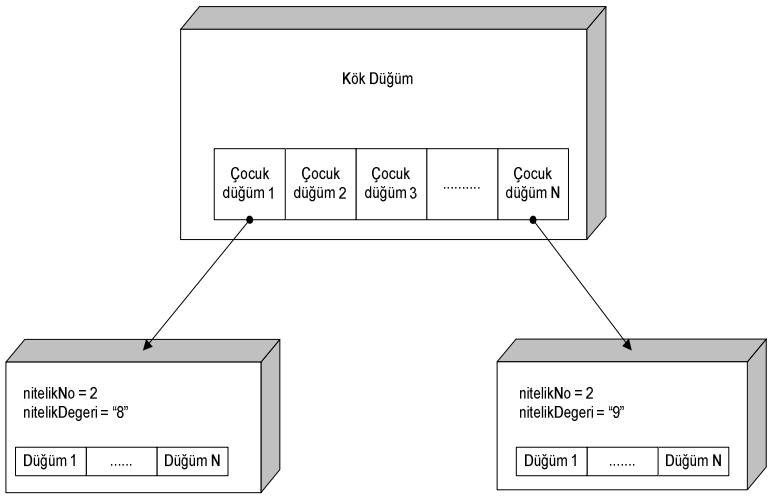
$$\text{Kazanç (S, A)} = \text{Entropi(S)} - \sum_{v \in Nitelik(A)} \frac{|S_v|}{|S|} Entropi(S_v) \qquad \textbf{(5.2)}$$

Burada, $S$ veri kümesini, $A$ niteliğe göre dallanmayı göstermektedir. $v$ ise $A$ niteliğinin mevcut değerlerini temsil etmektedir. Gerçeklenen karar ağacında her niteliğin iki tane etiket değeri (Evet : 1, Hayır :0) olabilmektedir.

Tüm nitelikler için kazanç hesaplandıktan sonra, dallanma için en yüksek kazancı sağlayan nitelik seçilmekte ve eldeki örnekler bu niteliğe göre dallara bölünmektedir. Her nitelik değerine ilişkin örnekler de ayrı bir dizide saklanmaktadır. Ağacın her dalı için (kullanılan bir niteliğin alt dallarda tekrar kullanılmaması kontrolü yapılarak) karar ağacı oluşturulması fonksiyonları tekrar çağrılmaktadır. Ayrıca, modeli kaydetmek amacıyla ağacı oluşturan nitelikler ve değerleri kök düğümüne eklenmektedir.

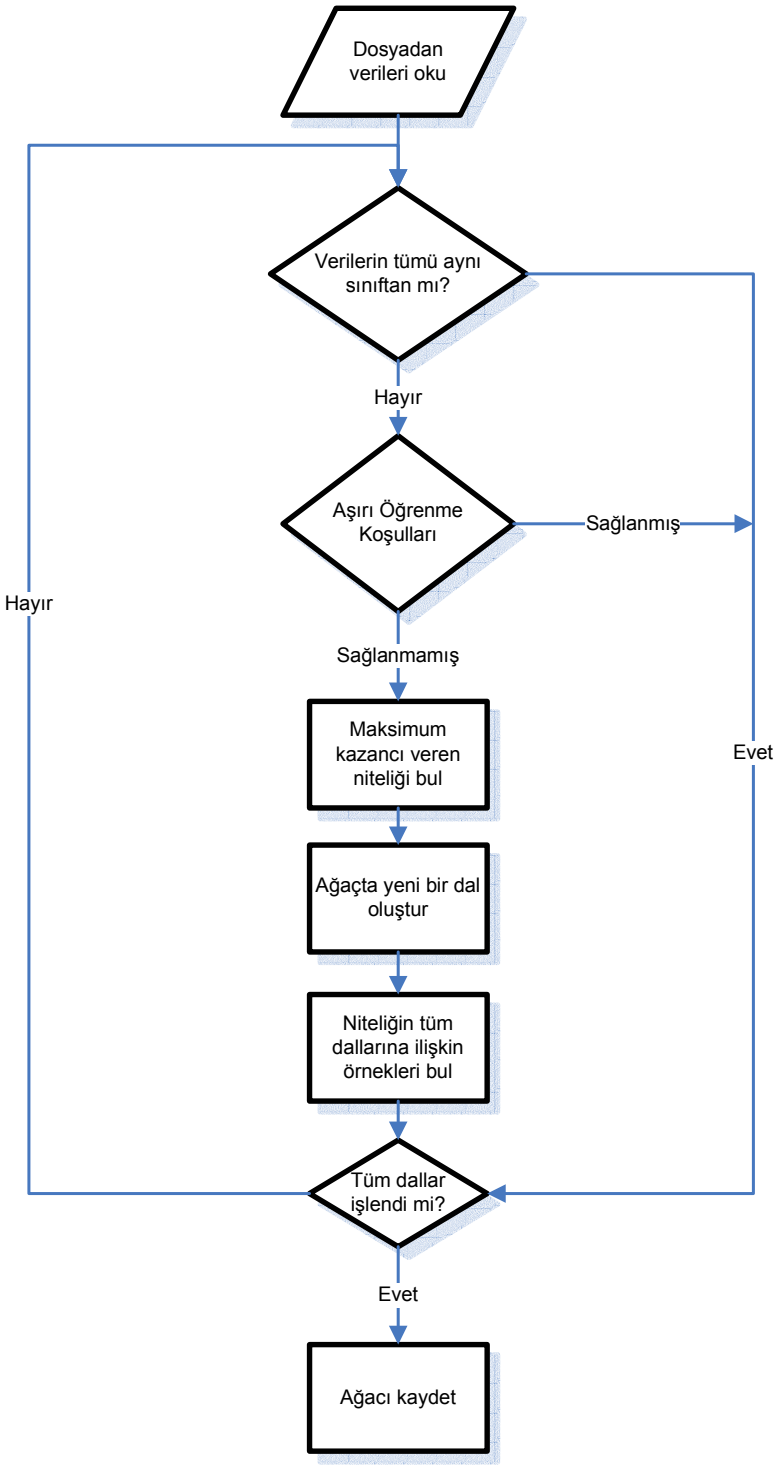
Uygulama kapsamında karar ağacı tabanlı yöntemlerin genel sorunu olan aşırı öğrenme sorunu da çözülmeye çalışılmaktadır. Bunun için de, ağaç belli bir seviyeye geldiğinde veya eldeki örneklerin sayısı belli bir minimum değere indiğinde ağacın daha fazla dallanması engellenmekte ve o anki örneklerin çoğunluğunu teşkil eden sınıf, o dalın etiketi olarak atanmaktadır.

Verileri sınıflandırmak için oluşturulan karar ağacı, daha sonra test edilebilmesi için yerelde bir dosyaya kaydedilmektedir. Oluşturulan modelde, her düğüm kendi çocuk düğümlerini içerdiğinden kök düğüm ve bu düğümün içerdiği düğüm dizisini kaydetmek yeterli olmaktadır. Uygulama kapsamında oluşturulan ağaç "agac.txt" ismiyle bir dosyaya kaydedilmektedir. Uygulama sonunda oluşan kök düğüm ve buna bağlı çocuk düğümlerin yapısı Şekil 5.8'de gösterilmektedir.




**Şekil 5.8:** Kök ve çocuk düğümler

Karar ağacı oluşturma algoritmasının akış diyagramı da Şekil 5.9'da gösterilmektedir.




**Şekil 5.9:** Karar ağacı oluşturulması

#### 5.4.2.3. Karar Ağaçları ile Soru Kalıbının Belirlenmesi

Karar ağacı oluşturulduktan sonra, girilen sorunun hangi kalıba ait olduğunun belirlenmesi amacıyla bütün nitelikler için değerler belirlenmektedir. Çözülmesi istenen problem alınmakta ve bu problem için nitelik değerleri doldurulmakta ve bir dosyaya kaydedilmektedir. Örneğin,

20'nin 1/2'si ile 30'un 2/3'ünün farkı nedir?

problemi için nitelik değerleri,

0,0,0,1,1,0,1,1,0,1,0,1,0

olarak bulunmaktadır. 0 değeri problemin niteliği sağlamadığını, 1 değeri ise sağladığını ifade etmektedir. Örneğin, 2. nitelik değeri için belirlenen 0 değeri, problemin "hangi" kelimesi ile başlamadığını ifade etmektedir.

Oluşturulan dosya ve modeli ifade eden "agac.txt" dosyası giriş olarak alınarak, ağaç üzerinde dallanmalar yapılarak, problemin hangi kalıba ait olduğu belirlenmektedir. Örneğin, yukarıdaki örnek problem için KALIP-8 bilgisi bulunmaktadır.

### 5.5. Sorunun Çözülmesi Modülü

Bu modülde, girilen aritmetik problemin, hangi kalıba ait olduğu (soru kalıbının belirlenmesi modülü aracılığı ile) bulunduktan sonra çözümlenmesi yapılmaktadır. Soru kalıbı belli olan aritmetik problem bir önceki modülden alınarak, problem içindeki kelimelere göre sonuç üretilmektedir. Örneğin,

"10'un 2/5'inin 1/3'ü nedir?"

örnek probleminin KALIP-1'e ait bir örnek problem olduğu soru kalıbının belirlenmesi modülünden bu modüle verilir. Bu modül, KALIP-1 için uygulanacak işlemleri uygular ve sonucu üretir. KALIP-1 için uygulanacak işlemler, cümle içindeki tamlayan kelimelerin ve tamlanan kelimelerin çarpılmasını ve çarpım sonucunun problemin sonucunu oluşturacağı yönündedir. Bu tanımlamaya göre yukarıdaki örnek problem içindeki tamlayan ve tamlanan kelimeler bulunur ve bu kelimelerin sayı olup olmadıklarına bakılır. Eğer sayı iseler ekleri atılarak sayı formatlarına çevrilmesi sağlanır. Bütün sayı içeren kelimeler için bu işlemler

yapıldıktan sonra tamlayan kelimeler ve tamlanan kelimeler çarpılır. Örnek problem için

10 * 2/5 * 1/3 = 4/3 sonucu elde edilir.

Yukarıda anlatılan yöntem yalnızca KALIP-1 için geçerli bir yöntemdir. Diğer bütün soru kalıpları için benzer yöntemler uygulanarak girilen aritmetik problemin sonucu üretilir ve bir sonraki modül olan "çözümün denklemler şeklinde ifade edilmesi" modülüne gönderilir.

### 5.6. Çözümün denklemler şeklinde ifade edilmesi ve gösterilmesi modülü

Bu modülde girilen aritmetik problemin matematiksel ifadelerle (denklemlerle) gösterilmesi sağlanır. Problem içerisinde geçen sayılar bir önceki modül olan sorunun çözülmesi modülünden alınır ve problem içerisindeki işlemler belirlenerek, yazı şeklindeki aritmetik problem matematiksel ifadelerle (denklemler ve dört işlem sembolleriyle) gösterilir. Örneğin,

"20/9 sayısının 1/2'sinin 4/3'ü nedir?"

probleminin denklemler şeklinde ifadesi

((20/9 * 1/2) * 4/3) = 40/27

olarak gösterilmektedir.

Çözüm, denklemler şeklinde ifade edilirken işlem önceliklerine dikkat edilmektedir. Toplama, çıkarma, çarpma, bölme ve üs alma işlemlerinden bir kaçının birlikte bulunduğu rasyonel sayılarda işlemler, aşağıdaki sıraya göre yapılmaktadır.

1) Parantezler ve kesir çizgisi işleme yön verir.
2) Üslü işlemler varsa sonuçlandırılır.
3) Çarpma - bölme yapılır.
4) Toplama - çıkarma yapılır.

Toplama ile çıkarma ve çarpma ile bölme kendi arasında öncelik taşımamaktadır. Özellikle çarpma ile bölmede öncelik söz konusu ise hangi işlemin önce yapılacağı parantezle belirlenmektedir. Örneğin,

"20'nin 1/2'si ile 30'un 2/3'ünün farkı nedir?"

problemi için bu modül,

((20 * 1/2) - (30 * 2/3)) = -10

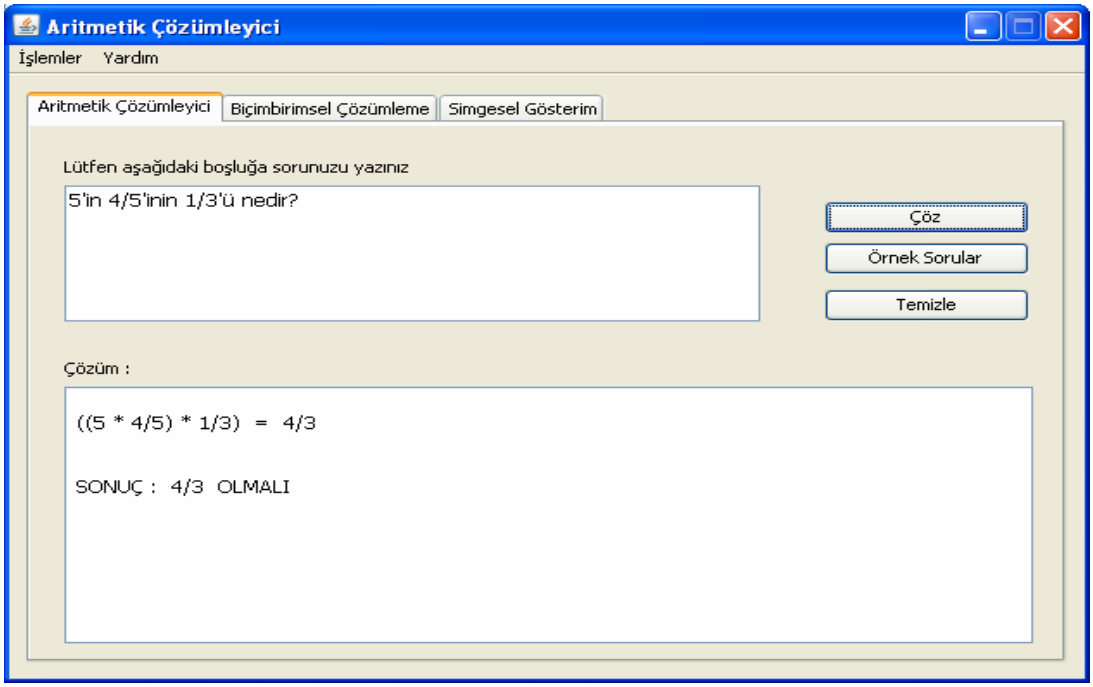
sonucunu üretmektedir. Örnekten de anlaşılacağı gibi, problem içindeki işlemler ve sayılar birbirlerinden parentezlerle ayrılmaktadır.

### 5.7. Sorunun ve Çözümün Simgelenmesi Modülü

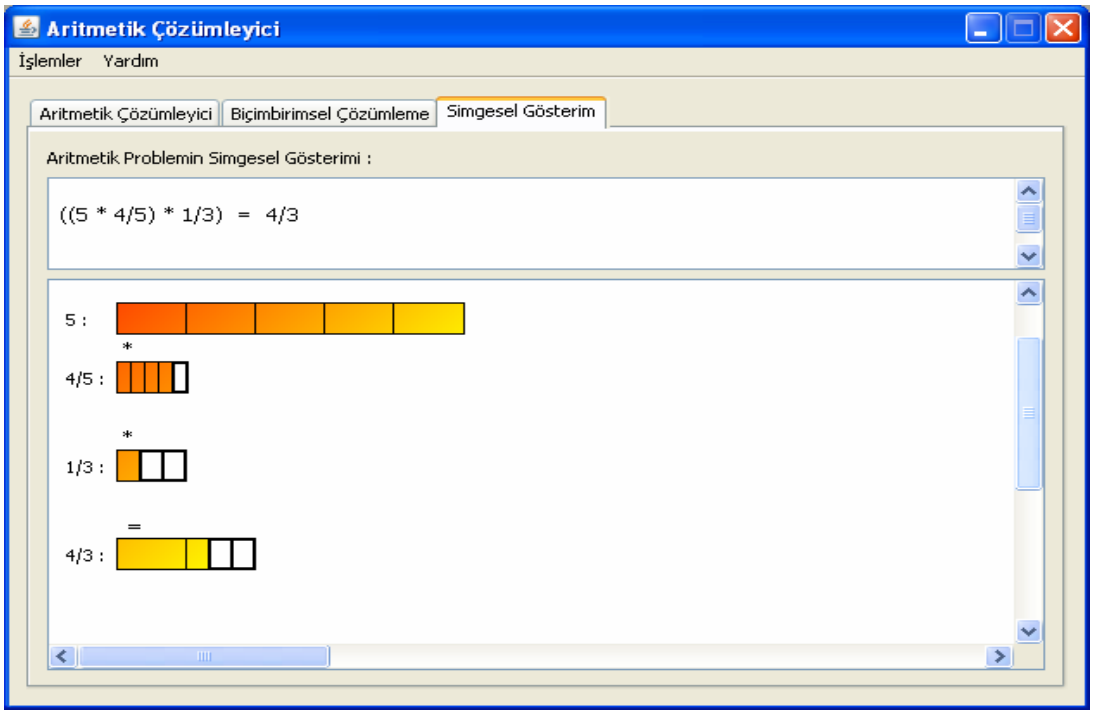
Gerçekleştirilen uygulamanın en son aşaması sorunun ve çözümün simgelenmesidir. Bu modülde, bir önceki modülden denklemler şeklinde ifade edilmiş soru ve çözüm alınmakta ve ilköğretim 1., 2., 3. sınıf öğrencilerine kesirli sayıların anlatılmasında kullanılan simgelerle (şekillerle) ifade edilmektedir.

Yapılan araştırmalar, öğrencilerin kesirli sayıları anlamasında geometriksel modellerin kullanılmasının daha iyi sonuçlar verdiği yönündedir. Öğrenciler kesirlerde toplama, çıkarma, bölme, çarpma işlemini her yıl rutin bir şekilde öğrenirler fakat daha sonraki yıllarda bu işlemlerin nasıl yapıldıklarını unuturlar. Öğrencilerin kesir işlemlerinde zorlanmalarının başlıca nedenlerinden birisi kesirleri anlamaları yerine formülleri ve algoritmayı ezberlemeleri bir diğeri de kesirlerin pay ve paydalarını farklı iki tam sayı olarak algılamalarıdır. İlköğretimin ilk kademesinde kesirleri yeni öğrenen öğrencilere kesirleri daha iyi öğretmek için kullanabilecek metot, kesirleri bir bütünün parçası olarak göstermektir. Bu yaklaşım kolay anlaşılmakta ve bunu anlatmak için de birçok model ve manipulatif araç kullanılabilmektedir. [25]

Geliştirilen uygulama kapsamında girilen aritmetik problemin ilköğretim öğrencilerinin anlayabileceği türden şekillerle gösterilmesi de gerçekleştirilmektedir. Şekil 5.10 ve 5.11'de örnek bir problem ve bu problemin simgelenmesi gösterilmektedir.




**Şekil 5.10:** Örnek problem



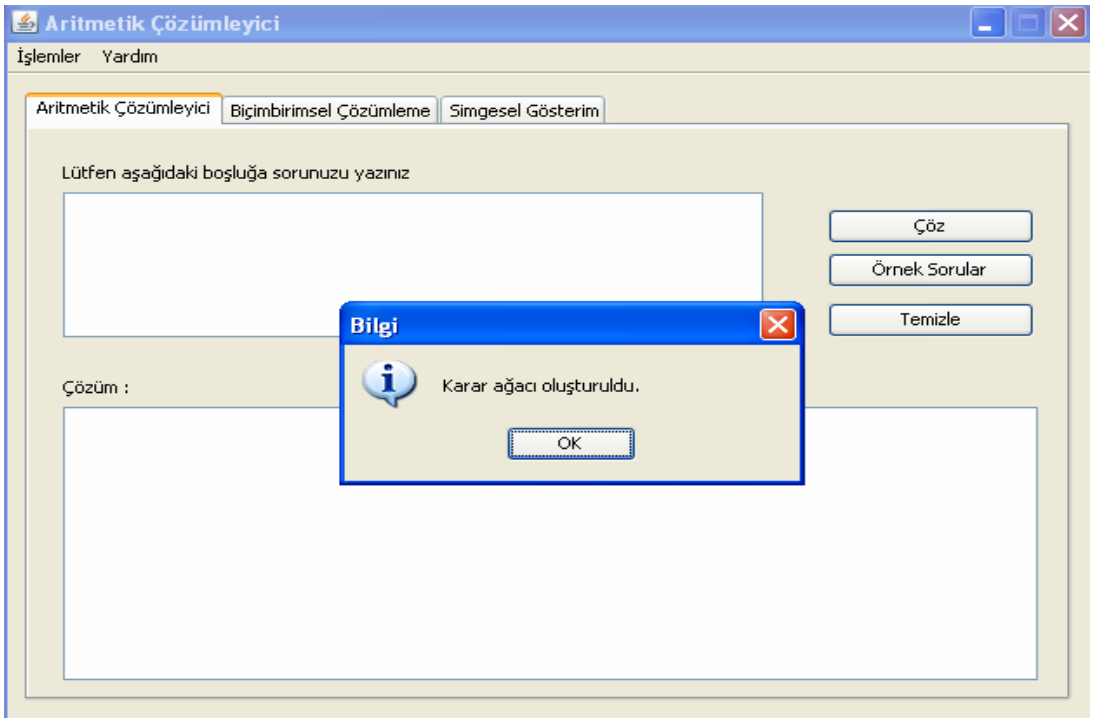

**Şekil 5.11:** Örnek problemin simgelenmesi

# 6. EKRAN GÖRÜNTÜLERİ VE ÖRNEK UYGULAMALAR

Bu bölümde geliştirilen uygulamanın arayüzü ile ilgili ekran görüntüleri iki örnek uygulama için gösterilmektedir. Uygulama Java dili kullanılarak gerçekleştirilmiştir.

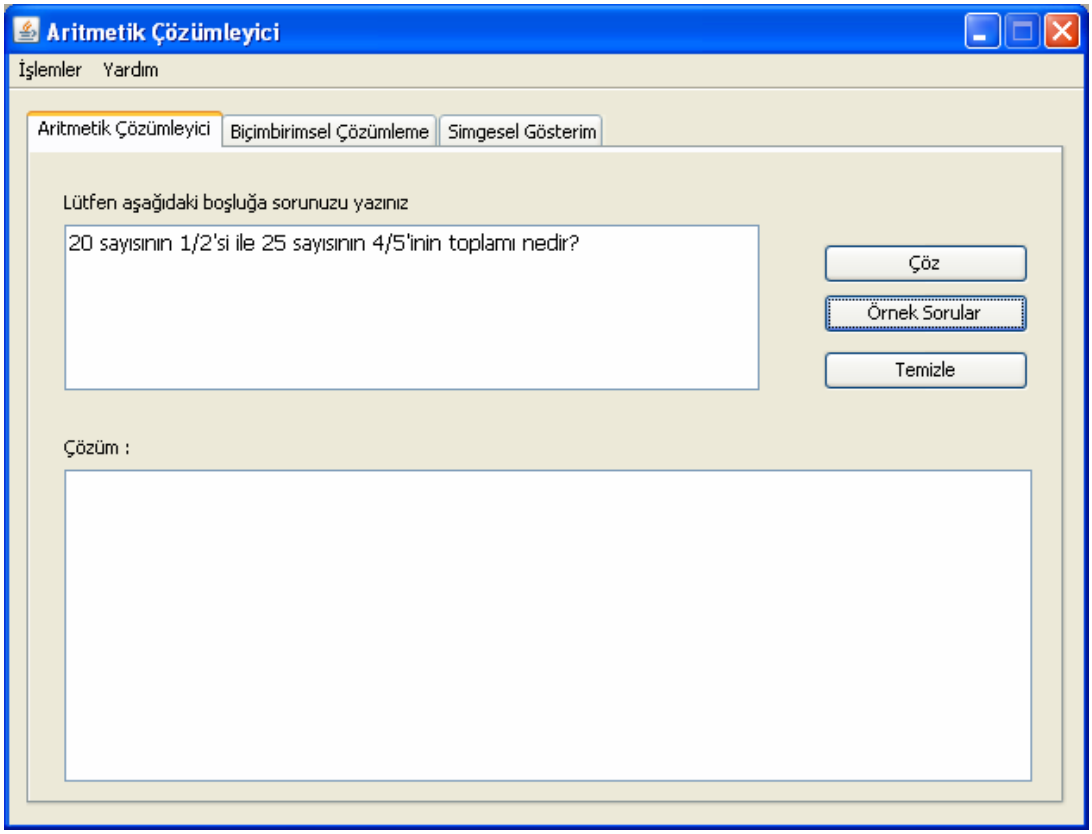
## 6.1. Örnek Uygulama I

Sorunun doğru olarak çözülebilmesi ve soru kalıbının belirlenebilmesi için öğrenme kümesinin uygulamaya tanıtılması gerekmektedir. Kullanıcı, öğrenme kümesini kendisi oluşturup uygulamayı bu öğrenme kümesi ile öğretebilirken, aynı zamanda uygulama kapsamında oluşturulan öğrenme kümesi ile de uygulamanın otomatik öğrenmesini sağlayabilir. Şekil 6.1'de "İşlemler" menüsünden "Otomatik Öğrenme" seçilerek, uygulamaya öğrenme kümesi tanıtılmaktadır.




**Şekil 6.1:** Öğrenme işlemi sonucu karar ağacının oluşturulması

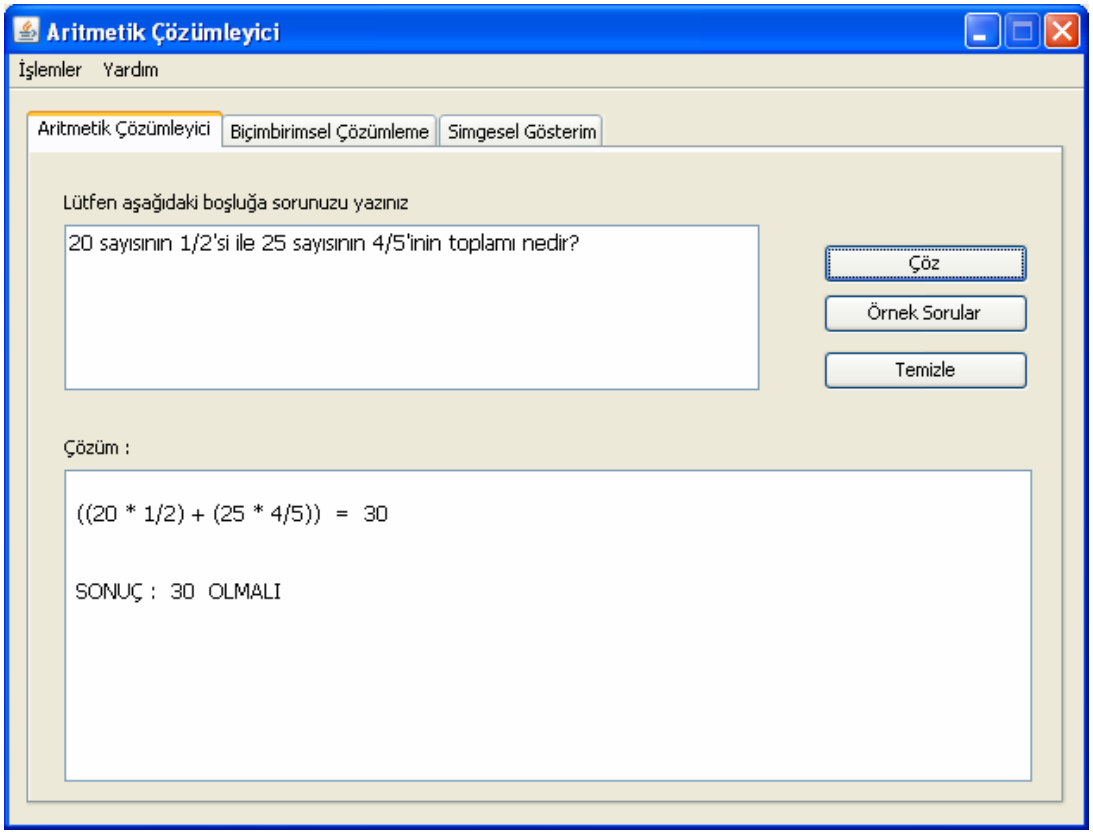
Kullanıcı, aritmetik problemini soru alanına yazar ve “Çöz” düğmesine tıklar. Şekil 6.2’de çözülmesi istenen aritmetik problemin soru alanına girilmiş hali görülmektedir.




**Şekil 6.2:** Örnek problem

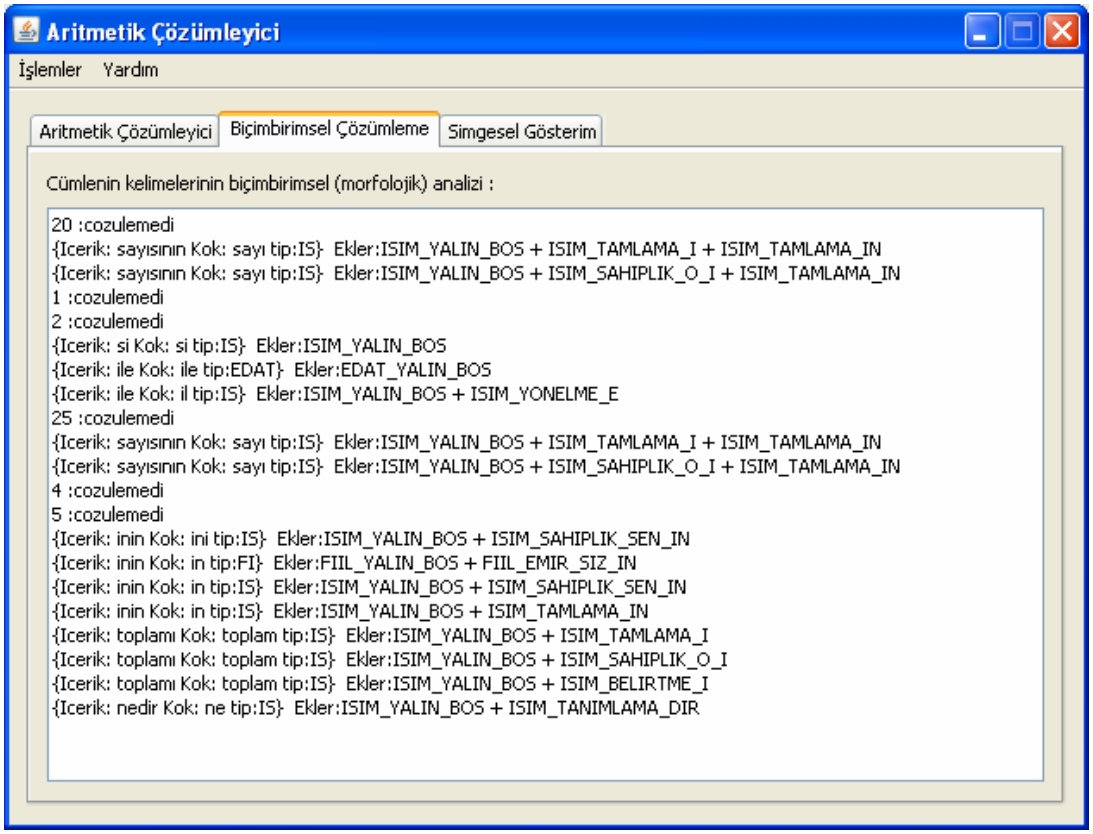
Uygulama, biçimbirimsel analiz, eklerin belirlenmesi, soru kalıbının belirlenmesi, sorunun çözülmesi, çözümün denklemler şeklinde ifade edilmesi ve gösterilmesi, sorunun ve çözümün simgelenmesi aşamalarını birer birer yerine getirir ve soruyu çözer.

Şekil 6.3’de sorunun çözümü kullanıcıya sunulmaktadır.



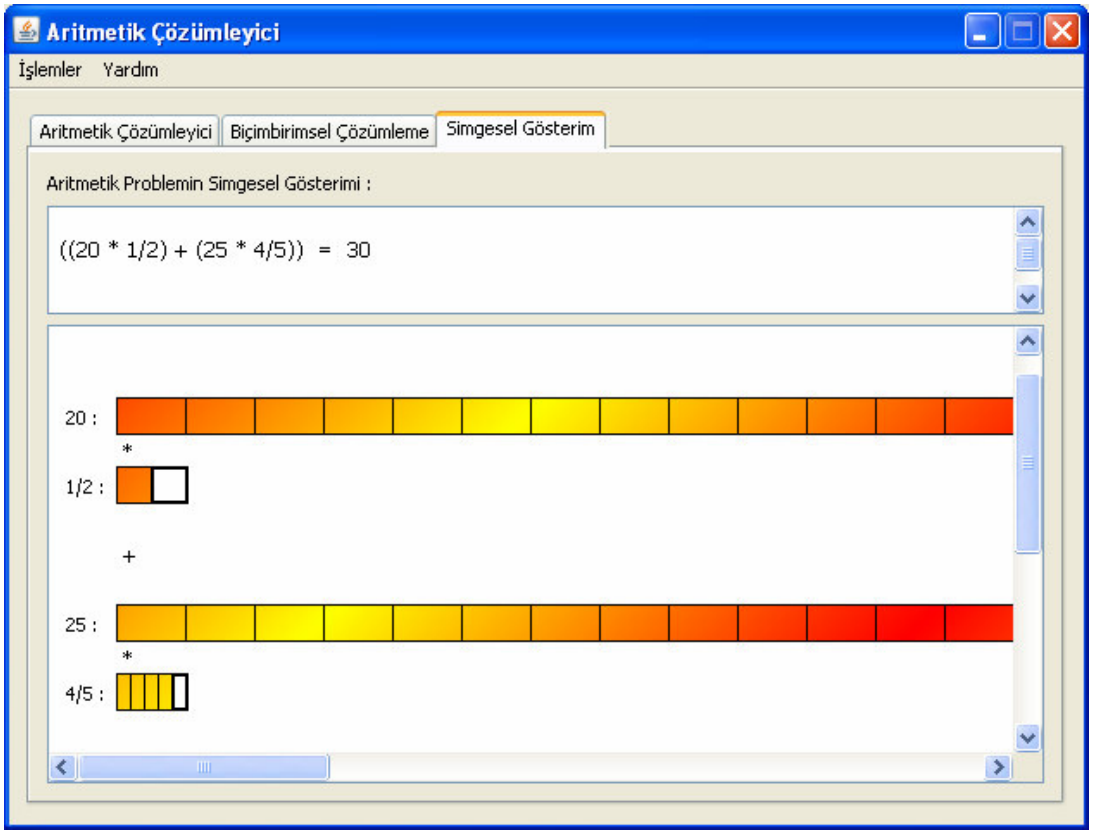

**Şekil 6.3:** Örnek problemin çözümü

Şekil 6.4'de biçimbirimsel analiz sonucu "Biçimbirimsel Çözümleme" sekmesinde kullanıcıya gösterilmektedir.




**Şekil 6.4:** Örnek problem için biçimbirimsel çözümleme

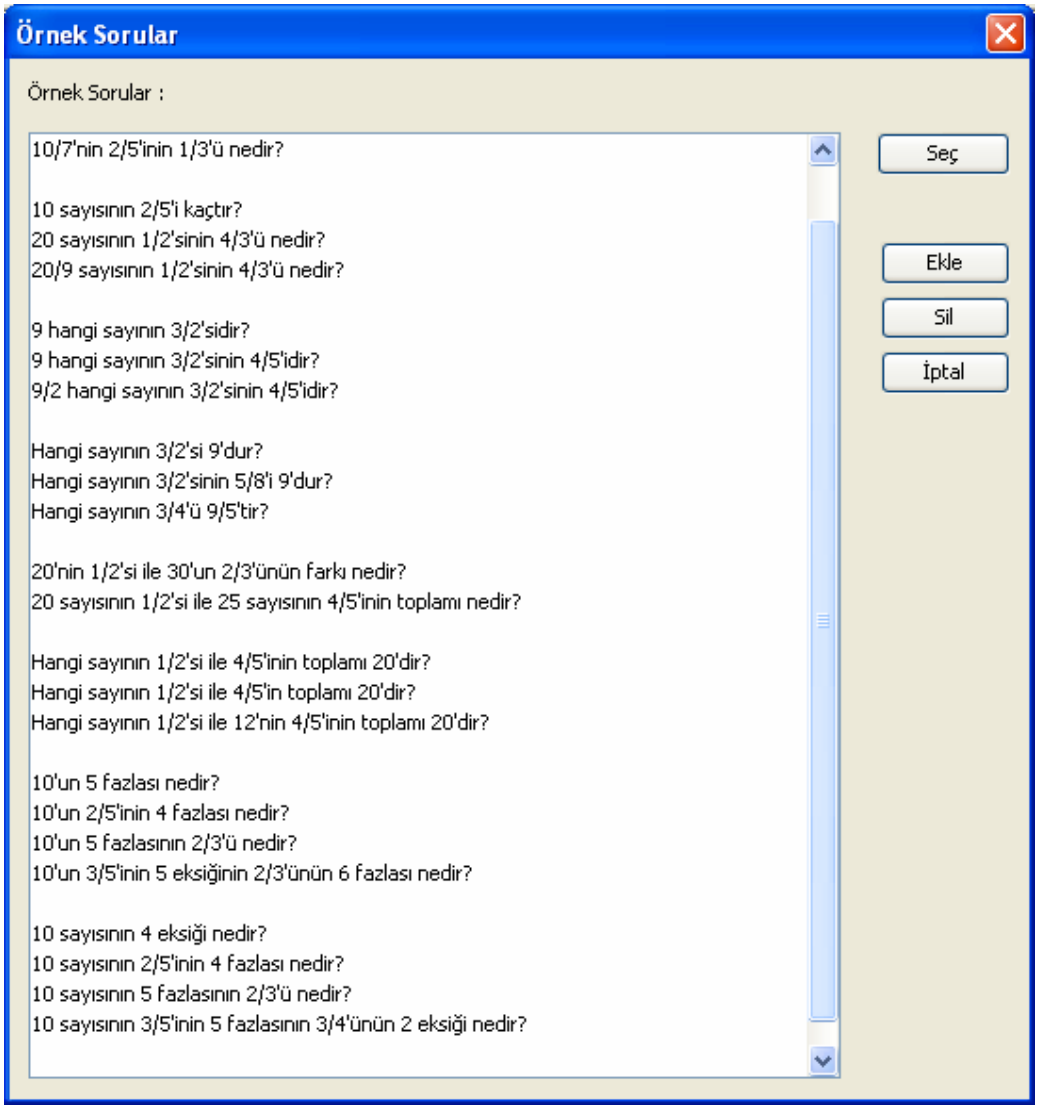
Şekil 6.5'te sorunun ve çözümün simgelenmesi, "Simgesel Gösterim" sekmesinde kullanıcıya gösterilmektedir.




**Şekil 6.5:** Örnek problemin simgesel gösterimi

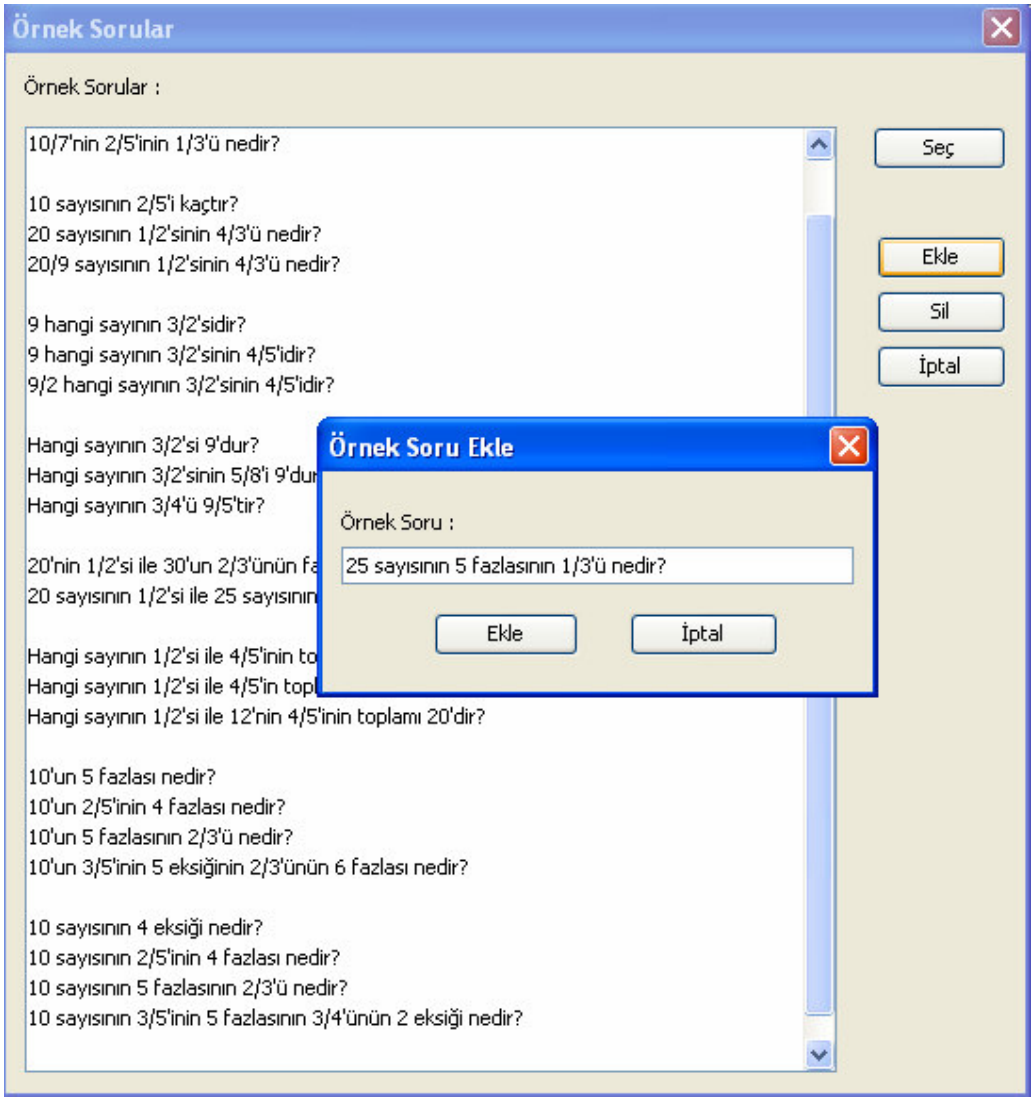
### 6.2. Örnek Uygulama II

Kullanıcı uygulamanın "Örnek Sorular" düğmesine tıklar ve daha önceden dosyaya yazılmış örnek sorulardan birini seçer. "Örnek Sorular" ekranı Şekil 6.6'da gösterilmektedir.




**Şekil 6.6:** Örnek Sorular Ekranı

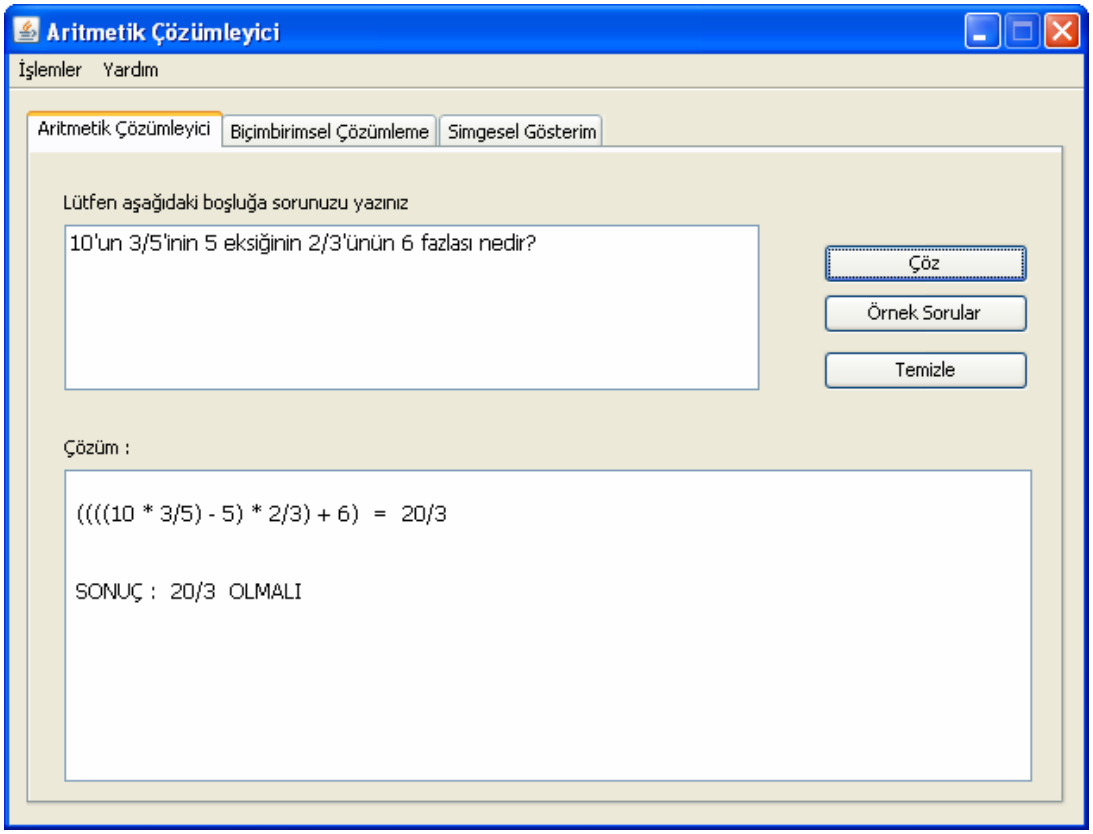
Kullanıcı bu ekranda örnek sorulara yeni soru ekleyebilir veya örnek sorulardan herhangi birini silebilir. "Örnek Soru Ekle" ekranı Şekil 6.7'de gösterilmektedir.




**Şekil 6.7:** Örnek Soru Eklenmesi

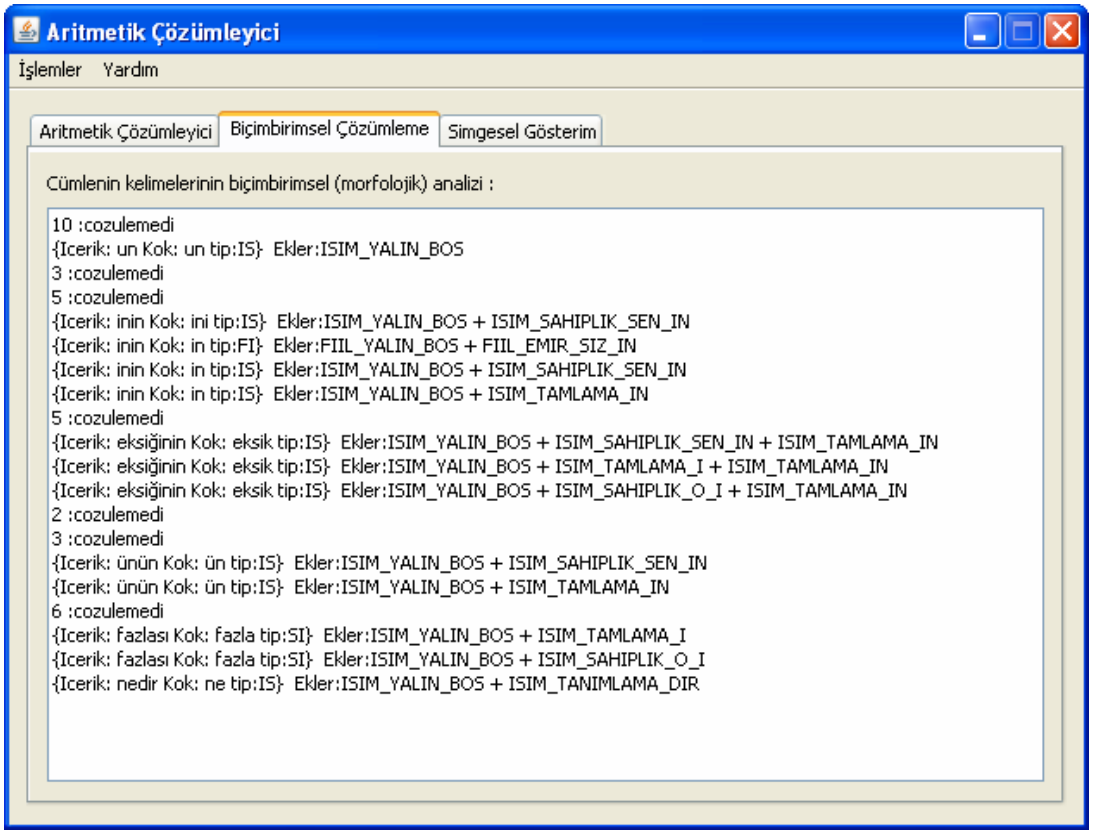
Uygulama, biçimbirimsel çözümleme, eklerin belirlenmesi, soru kalıbının belirlenmesi, sorunun çözülmesi, çözümün denklemler şeklinde ifade edilmesi ve gösterilmesi, sorunun ve çözümün simgelenmesi aşamalarını birer birer yerine getirir ve soruyu çözer.

Şekil 6.8'de sorunun çözümü kullanıcıya sunulmaktadır.



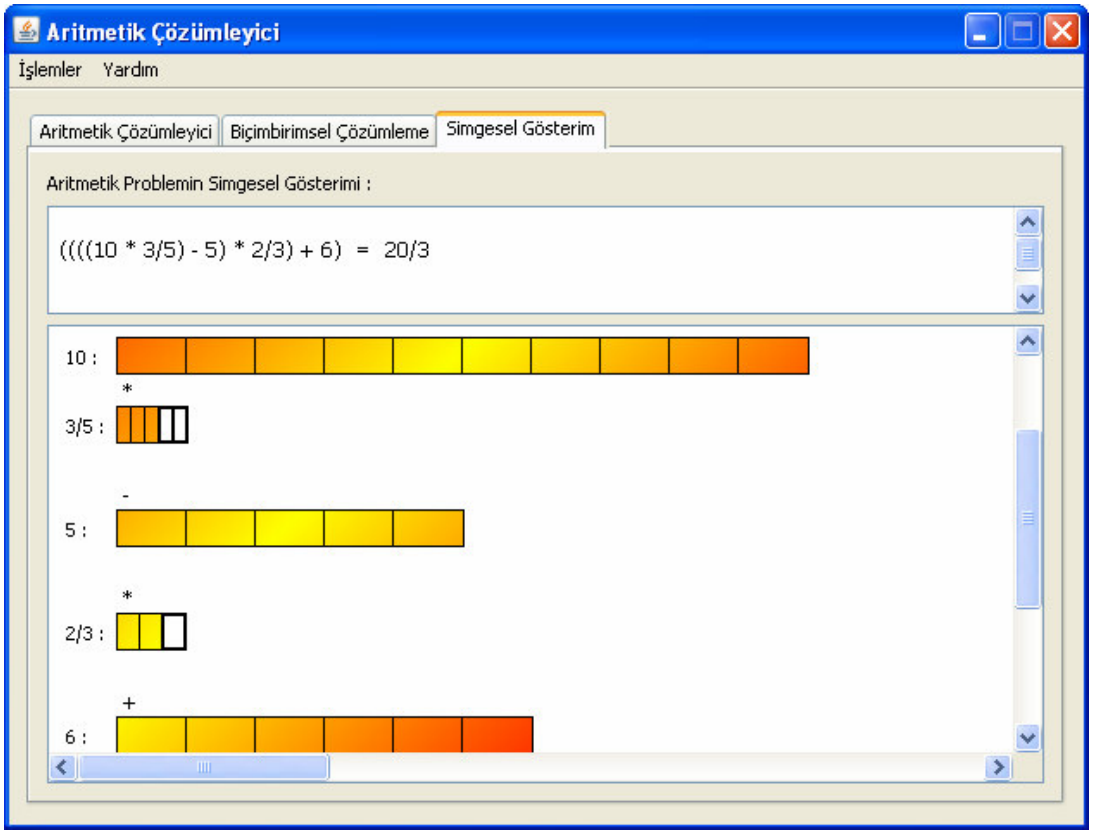

**Şekil 6.8:** Örnek Problemin Çözümü

Şekil 6.9'da biçimbirimsel analiz sonucu "Biçimbirimsel Çözümleme" sekmesinde kullanıcıya gösterilmektedir.




**Şekil 6.9:** Örnek problem için biçimbirimsel çözümleme

Şekil 6.10'da sorunun ve çözümün simgelenmesi, "Simgesel Gösterim" sekmesinde kullanıcıya gösterilmektedir




**Şekil 6.10:** Örnek problemin simgesel gösterimi

# 7. SONUÇLAR VE TARTIŞMA

Doğal dil işlemenin amacı; bir doğal dili çözümlemek, anlamak, yorumlamak, tasarlamak ve gerçekleştirmektir. Bu tez çalışmasında doğal dil işlemenin aşamaları kullanılarak Türkçe için çalışan bir uygulama geliştirilmiştir. Yapılan bu çalışmanın bu alanda yapılan diğer çalışmaların geliştirilmesine katkıda bulunacağına inanılmaktadır.

Problem çözme tabanlı sistemlerin temel amacı; insanların akılla çözebildiği matematik, fizik, mekanik gibi problemlerin bilgisayarlarla çözülebilmesi ve bu sayede insanların doğru sonucu bulup bulmadığının tespit edilebilmesidir.

Tez kapsamında ilköğretim öğrencileri seviyesindeki aritmetik matematik problemlerinin çözümlenmesi ve simgelenmesi gerçekleştirilmektedir. Problemlerin çözümlenmesi için öncelikle aritmetik problem, kullanıcıdan bir arayüz vasıtasıyla alınmakta ve diğer ara işlemlerden geçirilerek çözümlenmekte ve simgelenmektedir.

Bu çalışma daha önceden de belirtildiği gibi, Türkçe doğal dil işleme uygulamalarının sayısını ve çeşitliliğini artırmak, yeni fikirlere öncülük etmek, ilköğretim öğrencilerine aritmetik problemlerinin çözümü konusunda yardımcı olmak ve test imkanı sunmak, ilköğretim öğrencileri arasında bilgisayar kullanımını teşvik etmek, insanı anlayan makine kavramına katkıda bulunmak amacıyla geliştirilmiştir.

Geliştirilen uygulama ile aritmetik problem içindeki tüm kelimelerin biçimbirimsel analizleri yapılarak kökleri ve aldıkları ekler tespit edilmekte ve bu sayede cümle içindeki isim tamlamaları belirlenmektedir.

İlköğretim seviyesindeki Türkçe aritmetik problemler incelendiğinde bu problemlerin çeşitli tiplerde olduğu anlaşılmaktadır. Geliştirilen uygulamada girilen problemin hangi tipte olduğunun öğrenilmesi ve öğrenilen bu tipe göre problemin çözümlenmesi yöntemi benimsenmiştir. Bu sebeple Türkçe aritmetik problemler için 10 tane soru tipi belirlenmiş ve bu 10 soru tipinin çözümlenmesi için sınıflandırma yöntemlerinden karar ağaçları kullanılmıştır. Karar ağaçları ile soru tipinin

belirlenmesi için öncelikle bir model oluşturulması gerektiğinden, 75 tane örnek aritmetik problemi ve çözümünü içeren bir öğrenme kümesi oluşturulmuştur. Öğrenme kümesi oluşturulurken 75 tane örnek problem için 13 tane nitelik belirlenmiş ve örnek kümesindeki bütün örnekler için nitelik değerleri sırayla girilmiştir. Uygulama bu örnek kümesiyle eğitildikten sonra karar ağacı oluşturulmuş ve çözümlenmesi istenen problemin nitelik değerleri karar ağacı ile belirlenerek problemin hangi tipte olduğu belirlenmiştir.

Tipi belli olan problem çeşitli işlemlerden geçirilerek çözümlenmekte, denklemler ve dört işlem sembolleriyle ifade edilmekte ve ilköğretim öğrencilerine kesirli sayıların anlatılmasında kullanılan sembollerle simgelenmektedir.

Yapılan çalışma sonucu 10 farklı tipteki aritmetik problemlerin çözümlenme başarısı %100 olarak gözlemlenmiştir. Şimdiki aşamada yeni tipte bir problemin çözümlenmesi, öğrenme kümesinde problemin çözümü bulunmadığından gerçekleştirilememektedir.

Bu konuda bundan sonra yapılabilecek çalışmalar, soru tipi sayısının artırılmasıyla daha çok sayıda problemin çözümlenmesi ve simgelenmesi, bilinmeyen bir soru tipi geldiğinde uygulamanın bu soru tipini ve çözümünü otomatik öğrenmesi, soru kalıbı belirlenirken kullanılan nitelik sayılarının artırılması ve bu sayede soru kalıbının belirlenme başarımının artırılması ve Türkçe aritmetik problem çözmenin yanında Türkçe aritmetik problem üretme olarak sıralanabilir.

Geliştirilen uygulama diğer doğal dil işleme çalışmaları ile beraber kullanılabilirse daha faydalı uygulamalar ortaya çıkabilir. Örneğin geliştirilen uygulamaya metinden-sese çevirme (text-to-speech) ve sesten-metine çevirme(speech-to-text) teknikleri eklenirse, bilgisayar girilen problem ve çözümünü sesli olarak kullanıcıya iletebilir ve kullanıcıda sesli olarak çözümlenmesini istediği problemi bilgisayara aktarabilir. Bu sayede konuşan, düşünen, insanı anlayan bilgisayar kavramına katkıda bulunulmuş olunur.

## KAYNAKLAR


[1] **Oflazer, K. and Bozsahin, C.,** 1994. Natural Language Processing in Turkish, *Proceedings of Turkish Informatics Society TBD'94.*

[2] **Wikipedia**, 2007.

Erişim : http://en.wikipedia.org/wiki/Natural_language_processing.

[3] **Özyurt, Ö. ve Köse, C.,** 2006. Türkçe Tabanlı Diyalog Sistemi Tasarımı ve Kodlanması, *Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu ve Fuarı-ELECO'2006*, Bursa, Türkiye, 06-10 Aralık.

[4] **Altan, Z. ve Orhan, Z.,** 2006. Anlam Belirsizliği İçeren Türkçe Sözcüklerin Hesaplamalı Dilbilim Uygulamalarıyla Belirginleştirmesi*, XX. Ulusal Dilbilim Kurultayı*, İstanbul, Türkiye, 12-13 Mayıs.

[5] **Amasyalı, M. F. ve Yıldırım, T.,** 2004. Otomatik Haber Metinleri Sınıflandırma*, SIU 2004*, Kuşadası, Türkiye.

[6] **Amasyalı, F.**, Internet'ten Bilgi Çıkarımı, Y.T.Ü., İstanbul, Türkiye.

[7] **Erenler, Y.,** 2005. BLG 505 Doğal Dil İşleme Ders Notları, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[8] **Türk Dil Kurumu**, 2007. Erişim : www.tdk.gov.tr

[9] **Çakıroğlu, Ü.,** 2004. Türkçe İçin Doğal Dil İşleme Çalışmaları – 1, 20 Haziran.

Erişim : http://www.teknoturk.org/docking/yazilar/tt000135-yazi.htm

[10] **Bobrow, D.,** 1968. Natural language input for a computer problem-solving system, *Semantic Information Processing*, pp. 135-215, ed. M. Minsky, MIT Press, Cambridge, MA.

[11] **Charniak, E.,** 1969. Computer solution of calculus word problems, pp. 303-316, Proc. Int. Jt. Conf. Art. Intel., Washington, D. C.

[12] **Steele, G. L.,** 1990. Common Lisp the Language, Digital Press, Woburn, MA.

[13] **McIntosh, H.V. and Guzmán, A.,** 1967. A Miscellany of CONVERT Programming, Project MAC Memorandum MAC M 346, M.I..T., Cambridge, Mass., April.

[14] **Bundy, A., Byrd, L., Luger, G., Mellish, C., Milne, R. and Palmer, M.,** 1979. MECHO: a program to solve mechanics problems, Working Paper 50, Department of Artificial Intelligence, Edinburgh University.

[15] **Say, A. C. C.,** 2001. Understanding arithmetic problems in Turkish, *International Journal of Pattern Recognition and Artificial Intelligence* Vol. 15, pp. 359-374.

[16] **Hamzaoğlu, İ.,** 1993. Machine translation from Turkish to other Turkic languages and an implementation for the Azeri language, *M. S. Thesis*, Department of Computer Engineering, Boğaziçi University, İstanbul.

[17] **Hengirmen, M.,** 2005. Türkçe Dilbilgisi, Engin Yayınevi, Ankara, Türkiye.

[18] **Selçuk Üniversitesi Uzaktan Eğitim Programı**, 2007.

Erişim: http://farabi.selcuk.edu.tr/suzep/turk_dili/ders_notlari/bolum_17/bolum17.html#K1

[19] **Zemberek**, 2007. Erişim : https://zemberek.dev.java.net/

[20] **Zemberek Projesi Sözlük Yapısı ve Kök Seçicileri**, 2007.

Erişim : https://zemberek.dev.java.net/files/documents/2681/13349/zemberek.pdf

[21] **Eker, H.,** 2005. Veri Madenciliği veya Bilgi Keşfi.

Erişim : http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=538

[22] **Alpaydın, E,.** 2000. Zeki veri madenciliği : Ham veriden altın bilgiye ulaşmaın yöntemleri, *Bilişim 2000 Veri Madenciliği Eğitim Semineri*, İstanbul.

[23] **Gündüz-Öğüdücü, Ş.,** 2005. BLG 607 Veri Madenciliği Ders Notları, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[24] **Özekes S.,** 2003. Veri Madenciliği Modelleri ve Uygulama Alanları, *İstanbul Ticaret Üniversitesi Dergisi* , **3** ,65-82.

[25] **Şiap, İ., ve Duru, A.,** 2004. Kesirlerde Geometriksel Modelleri Kullanabilme Becerisi, Gazi Üniversitesi, Kastamonu Eğitim Dergisi, Cilt 12, No 1, s.89.

## ÖZGEÇMİŞ

1982 Trabzon doğumlu Ömer Bahadır, orta öğrenimini Trabzon 100.Yıl İlköğretim Okulu'nda, lise öğrenimini ise Erzurum Fen Lisesi'nde tamamlamıştır. 1999 yılında Erzurum Fen Lisesi'nden mezun olduğu sene, İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği bölümünü kazanmış ve 2004 yılında bu bölümden mezun olmuştur. Aynı yıl İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans Programı'na girmiş olup, 2007 Haziran itibarıyla yüksek lisans tezini sunmaktadır.