T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DOĞAL DİL İŞLEME TEKNİKLERİNİ KULLANARAK TÜRKÇE METİNLERDEN BİLGİ ÇIKARIMI

Hazırlayan
Furkan BAĞIRGAN

Danışman
Doç. Dr. Beyza GÖRKEMLİ BAYRAM

Yüksek Lisans Tezi

Ağustos 2025
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DOĞAL DİL İŞLEME TEKNİKLERİNİ KULLANARAK TÜRKÇE METİNLERDEN BİLGİ ÇIKARIMI

(Yüksek Lisans Tezi)

Hazırlayan
Furkan BAĞIRGAN

Danışman
Doç. Dr. Beyza GÖRKEMLİ BAYRAM

Ağustos 2025
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Furkan Bağırgan

İmza

“Doğal Dil İşleme Tekniklerini Kullanarak Türkçe Metinlerden Bilgi Çıkarımı” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan	**Danışman**
Furkan BAĞIRGAN	Doç. Dr. Beyza GÖRKEMLİ BAYRAM
İmza	İmza

Bilgisayar Mühendisliği ABD Başkanı

Prof. Dr. Veysel ASLANTAŞ

İmza

Doç. Dr. Beyza GÖRKEMLİ BAYRAM danışmanlığında **Furkan BAĞIRGAN** tarafından hazırlanan **"Doğal Dil İşleme Tekniklerini Kullanarak Türkçe Metinlerden Bilgi Çıkarımı"** adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

08 / 08 / 2025

JÜRİ:

Danışman : Doç. Dr. Beyza GÖRKEMLİ BAYRAM .....

Üye : Prof. Dr. Celal ÖZTÜRK .....

Üye : Dr. Öğr. Üyesi Serhat HIZLISOY .....

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun ……...… tarih ve ……..……sayılı kararı ile onaylanmıştır.

……. /……../ ……

Prof. Dr. Munise Didem DEMİRBAŞ

Enstitü Müdürü

TEŞEKKÜR

Bana çalışmam süresince her türlü destek ve yardımı sağlayan başta danışman hocam Doç. Dr. Beyza GÖRKEMLİ BAYRAMA'a teşekkürlerimi sunarım. Her konuda bana inanan ve desteklerini esirgemeyen aileme de ayrıca teşekkür ederim.

Furkan BAĞIRGAN

Ağustos 2025, KAYSERİ

DOĞAL DİL İŞLEME TEKNİKLERİNİ KULLANARAK TÜRKÇE METİNLERDEN BİLGİ ÇIKARIMI

Furkan BAĞIRGAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Ağustos 2025
Danışman: Doç. Dr. Beyza GÖRKEMLİ BAYRAM

ÖZET

Metinler, insanların bilgiyi aktarmak için kullandığı araçların başında gelmektedir. Bilgiye erişim açısından metinlerden bilgi çıkarımı bu hususta büyük önem taşımaktadır. Metin verilerinin tek tek insanlar tarafından kontrol edilerek gerekli bilgilerin çıkarılması epey bir vakit ve maliyete yol açar. Bu sorunları ortadan kaldırmak için doğal dil işleme teknikleri kullanılır. Doğal dil işlemede yaygın olarak kullanılan anahtar sözcük çıkarımı, bir metnin en önemli ve temsil edici kelimelerini belirleyerek içeriği özetlemeye yardımcı olan temel yöntemlerden biridir. Anahtar sözcük çıkarımı anlamsal bağlam temelli ve sıklık frekansı temelli olmak üzere iki yolla mümkündür. Anlamsal bağlam, cümledeki kelimelerin anlamlarına dayalı bir çıkarım yapmaya olanak tanır. Sıklık frekansı ise kelimelerin metinde tekrarlanma sıklığı üzerinden elde edilir. Sıklık frekanslarının hesaplanması için öncelikle Türkçedeki kelimelerin yapısı bilinmelidir. Hiçbir eke sahip olmayan kökler ve bu köklerin yapım ekleri alarak yeni bir anlam kazanmasıyla oluşan gövdeler, tek başlarına bir anlama sahipken çekim ekleri alan kelimeler yeni bir anlam kazanmaz. Bu çalışmada, önceki çalışmaları temel alan yeni bir gövdeleme yöntemi geliştirilerek, Türkçedeki kısa öyküler üzerinde sıklık frekansları ile anahtar sözcük çıkarımı yapılmıştır. Çalışmanın sonuçları, sözlük kullanan kural tabanlı yeni bir gövdeleme yöntemi ile anahtar sözcük çıkarımının, anlatı türündeki metinler için sade ve etkili bir yaklaşım olup uygulanabilirliğini göstermiştir.

Anahtar Kelimeler: Doğal dil işleme, Metinden bilgi çıkarımı, Anahtar sözcük çıkarımı, Gövdeleme, Sıklık frekansı

EXTRACTING INFORMATION FROM TURKISH TEXTS USING NATURAL LANGUAGE PROCESSING TECHNIQUES

Furkan BAĞIRGAN

Erciyes University, Graduate School of Natural and Applied Sciences
Master Thesis, August 2025
Supervisor: Assoc. Prof. Dr. Beyza GÖRKEMLİ BAYRAM

ABSTRACT

Texts are among the primary tools people use to transfer information. Extracting information from texts is of great importance in terms of accessing information. Checking text data one by one by people to extract the necessary information leads to a lot of time and cost. Natural language processing techniques are used to eliminate these problems. Keyword extraction, widely used in natural language processing, is one of the basic methods that help summarize the content by determining a text's most essential and representative words. Keyword extraction is possible in two ways: semantic context-based and frequency-based. Semantic context allows an inference based on the meanings of the words in the sentence. Frequency-based occurrence is obtained based on the frequency of repetition of the words in the text. Understanding the Turkish language's structural characteristics is essential to compute word frequency. Roots without any suffixes, and stems formed by adding derivational suffixes to these roots, acquire new meanings on their own, while words with inflectional suffixes do not. In this study, a new stemming method based on previous studies was developed to extract keywords from Turkish short stories using frequency. The study results showed that keyword extraction with a new rule-based stemming method using a dictionary is a simple and effective approach, and its applicability for narrative texts.

Keywords: Natural language processing, Information extraction from text, Keyword extraction, Stemming, Word frequency

İÇİNDEKİLER

DOĞAL DİL İŞLEME TEKNİKLERİNİ KULLANARAK TÜRKÇE METİNLERDEN BİLGİ ÇIKARIMI

BİLİMSEL ETİĞE UYGUNLUK ii
YÖNERGEYE UYGUNLUK iii
KABUL VE ONAY iv
TEŞEKKÜR v
ÖZET vi
ABSTRACT vii
İÇİNDEKİLER viii
KISALTMALAR xi
TABLOLAR LİSTESİ xii
ŞEKİLLER LİSTESİ xiii
GİRİŞ 1

1. BÖLÜM

GENEL BİLGİLER

1.1. DDİ ve Bilgi Çıkarımı 2
1.2. Gövdeleme ve ASÇ 3
1.3. Türkçenin Morfolojik Zorlukları 4
1.4. Tezin Amacı ve Kapsamı 5

2. BÖLÜM

LİTERATÜR TARAMASI

2.1. Gövdeleme için Yapılan Çalışmalar 6
2.2. ASÇ için Yapılan Çalışmalar 11

3. BÖLÜM

MATERYAL ve YÖNTEM

3.1. Materyal 15
3.1.1. Türkçede Kelime Türleri 15
3.1.1.1. İsimler 15
3.1.1.1.1. İsim 16
3.1.1.1.2. Sıfat 16
3.1.1.1.3. Zamir 16
3.1.1.1.4. Zarf 17
3.1.1.1.5. Edat 17
3.1.1.1.6. Bağlaç 17
3.1.1.1.7. Ünlem 17
3.1.1.2. Fiiller 17
3.1.1.3. Fiilimsiler 18
3.1.2. Türkçede Kelime Yapısı 19
3.1.2.1. Kökler 20
3.1.2.2. Gövdeler 20
3.1.2.3. Ekler 20
3.1.2.3.1. Yapım Ekleri 22
3.1.2.3.2. Çekim Ekleri 23
3.1.3. Türkçede Ses Olayları 29
3.1.4. Veri Materyalleri 30
3.1.4.1. Durdurma Sözcüklerinin Hazırlanması 31
3.1.4.2. Eklerin Hazırlanması 31
3.1.4.3. Sözlüğün Hazırlanması 33
3.1.4.4. Öykülerin Hazırlanması 34
3.2. Yöntem 35
3.2.1. Ön İşleme 36
3.2.2. Gövdeleme 36
3.2.2.1. İsimler için SDM Tasarımı 38
3.2.2.2. İsimler için SDM Koşulları 39

3.2.2.3. İsimler için Gövde Kontrolü 46
3.2.2.4. Fiiller için SDM Tasarımı 48
3.2.2.5. Fiiller için SDM Koşulları 48
3.2.2.6. Fiiller için Gövde Kontrolü 56
3.2.2.7. Fiilimsiler için SDM Tasarımı 57
3.2.2.8. Fiilimsiler için SDM Koşulları 57
3.2.2.9. Fiilimsiler için Gövde Kontrolü 61
3.2.3. Kelime Türü Seçme 61
3.2.3.1. Tek Harfli Eklerin Kontrolü 62
3.2.3.2. Sözlükte Bulunmayan Gövdelerin Kontrolü 63
3.2.4. Frekans Analizi 67

4. BÖLÜM

BULGULAR ve TARTIŞMA

4.1. Bulgular ve Tartışma 69

5. BÖLÜM

SONUÇ ve ÖNERİLER

5.1. Sonuç ve Öneriler 76

KAYNAKÇA 78
ÖZGEÇMİŞ 83

KISALTMALAR

DDİ: Doğal Dil İşleme

ASÇ: Anahtar Sözcük Çıkarımı

SDM: Sonlu Durum Makinesi

TDK: Türk Dil Kurumu

TABLOLAR LİSTESİ

Tablo 3.1. Gövde Oluşumları 21
Tablo 3.2. Örnek İsimden İsim Üreten Yapım Ekleri 22
Tablo 3.3. Örnek İsimden Fiil Üreten Yapım Ekleri 22
Tablo 3.4. Örnek Fiilden İsim Üreten Yapım Ekleri 23
Tablo 3.5. Örnek Fiilden Fiil Üreten Yapım Ekleri 23
Tablo 3.6. Örnek Durdurma Sözcükleri 31
Tablo 3.7. Örnek Çekim Ekleri 33
Tablo 4.1. Tekli Gramların Frekans Değerleri 72
Tablo 4.2. İkili Gramların Frekans Değerleri 72
Tablo 4.3. Gövdeleme Test Sonucu 73
Tablo 4.4. Yanlış Gövdeleme Sebepleri ve Sayıları 74
Tablo 4.5. Sözlük Düzeltme Sonucu 75

ŞEKİLLER LİSTESİ

Şekil 1.1. Örnek Gövdeleme İşlemi 4
Şekil 3.1. Türkçede Kelime Yapısı 19
Şekil 3.2. Türkçede İsme Gelen Çekim Ekleri Düzeni 25
Şekil 3.3. Türkçede Fiile Gelen Çekim Ekleri Düzeni 27
Şekil 3.4. Türkçede Fiilimsileri Oluşturan Eklerin Düzeni 28
Şekil 3.5. Sözlük Düzenleme Adımları 34
Şekil 3.6. Yöntem Geliştirme Aşamaları 35
Şekil 3.7. Ön İşleme Adımı Kodları 36
Şekil 3.8. Örnek SDM Yapısı 37
Şekil 3.9. İsimler için Oluşturulan SDM 38
Şekil 3.10. Fiiller için Oluşturulan SDM 48
Şekil 3.11. Fiilimsiler için Oluşturulan SDM 57
Şekil 3.12. Frekans Analizi Adımları 67
Şekil 3.13. Yöntem Akış Şeması 68

GİRİŞ

Bilginin önemi günümüzde daha da artarken bilginin elde edilmesi için de her geçen gün yeni yöntemler geliştirilmektedir. Yazılı kaynaklar yani metinler insanların bilgiyi aktarmak için kullandığı araçların başında gelmektedir. Bilgiye erişim açısından metinlerden bilgi çıkarımı bu hususta büyük önem taşımaktadır. Binlerce veya milyonlarca metin verisi tek tek insanlar tarafından kontrol edilerek gerekli bilgilerin çıkarılıp daha sonra bu bilgiler üzerinde istenilen işlemlerin yapılması epey bir vakit ve maliyete yol açacaktır. Bu sorunları ortadan kaldırmak için doğal dil işleme (DDİ) temelli bir sistem kullanılabilir. DDİ Kullanıcının işini kolaylaştırmak ve bilgisayarla doğal dilde iletişim kurma arzusunu gerçekleştirmek için ortaya çıkmıştır. En büyük kullanım alanlarından birisi de metinden bilgi çıkarımıdır. Anahtar sözcükler, metinlerden çıkarılan bilgilerin başında gelmektedir. Anahtar sözcük çıkarımı (ASÇ), bir metnin en önemli ve temsil edici kelimelerini belirleyerek içeriği özetlemeye yardımcı olan temel yöntemlerden biridir. ASÇ anlamsal bağlam temelli ve sıklık frekansı temelli olmak üzere iki yolla mümkündür. Fakat Türkçe dilinin yapısından kaynaklı oluşan bazı problemler bu durumu zorlaştırmaktadır. Anahtar sözcükleri sıklık frekansı üzerinden belirlemek için öncelikle kelimelerin Türkçedeki yapısı bilinmelidir. Türkçe gibi sondan eklemeli yapıya sahip dillerde ASÇ yapmak diğerlerine göre daha zordur. Anahtar sözcükler metin için önem arz ettiğinden farklı anlamlardaki kelimeleri aynı köke sahip oldukları için bir olarak kabul etmek yanlış olacaktır. Bu yüzden her bir kelimenin gövdesine ulaşılması gerekmektedir. Bu tez çalışmasında, daha önce üzerine çalışılmamış Türkçe kısa öyküler için ASÇ yapılmıştır. DDİ teknikleri ile yeni bir gövdeleme yöntemi geliştirilerek kelimelerin gövdesi bulunmuş ardından sıklık frekansı üzerinden ASÇ gerçekleştirilmiştir. Böylelikle Türkçe dil yapısının zorlukları büyük oranda aşılmıştır. Geliştirilen gövdeleme yöntemi daha önce yapılmış Zemberek [14] ve Snowball [24] yöntemleri ile karşılaştırılmış ve daha başarılı olduğu görülmüştür. Çalışma sonucunda Türkçe metinler üzerinde ASÇ için sade ve etkili bir yaklaşım sunulmuştur.

1. BÖLÜM

GENEL BİLGİLER

1.1. DDİ ve Bilgi Çıkarımı

DDİ, insanların günlük yaşamda kullandıkları dilleri bilgisayarların analiz edebilmesini, anlamlandırabilmesini ve üretebilmesini amaçlayan bir yapay zekâ ve dilbilim disiplinidir [1]. Dilsel yapıların hem biçimsel yani, sözdizimsel hem de anlamsal yani semantik boyutlarını dikkate alarak yürütülen çalışmalarla, bilgisayarların insan diline dair verileri işleyebilecek kapasiteye ulaşmasını sağlar. DDİ, son yıllarda makine öğrenmesi ve derin öğrenme tekniklerinin gelişmesiyle birlikte çeşitli alt görevlere ve çok sayıda pratik uygulama alanına sahiptir:

- Dil Modelleme ve Otomatik Metin Üretimi
- Makine Çevirisi
- Adlandırılmış Varlık Tanıma
- Duygu Analizi
- Bilgi Erişimi ve Soru-Cevap Sistemleri
- Metin Sınıflandırma ve Kümeleme
- Otomatik Özetleme

Bu uygulamaların başarısı, dilin yapısal karmaşıklığının ne ölçüde çözümlenebildiğine doğrudan bağlıdır. Özellikle Türkçe gibi eklemeli dillerde bu karmaşıklık daha belirgin hale gelir.

Bilgi çıkarımı ise yapılandırılmamış ya da yarı yapılandırılmış metinlerden anlamlı ve yapılandırılmış bilgi elde etme sürecidir. Bu süreçte amaç, metin içerisindeki kişi, yer, organizasyon, tarih gibi varlıkları tanımlamak, olayları belirlemek ve bu öğeler arasındaki ilişkileri ortaya koymaktır [1]. Bilgi çıkarımı sistemleri sayesinde büyük hacimli metin koleksiyonlarından otomatik olarak anlamlı özetler üretmek, içerik temelli sorgulama yapmak ve karar destek sistemleri oluşturmak mümkün hale gelmiştir. Bu süreç genel olarak aşağıdaki aşamaları içerir:

- **Önişleme:** Metinlerin temizlenmesi, küçük harfe dönüştürülmesi, noktalama işaretlerinin ayıklanması, durdurma sözcüklerinin kaldırılması.
- **Morfolojik Çözümleme:** Kelimelerin kök, gövde ve ek bileşenlerine ayrılması.
- **Adlandırılmış Varlık Tanıma:** Kişi, yer, kurum gibi özel adların belirlenmesi.
- **İlişki Çıkarımı:** Varlıklar arasındaki semantik bağların ortaya konması.
- **Anahtar Sözcük Belirleme:** Metnin anlamını özetleyen kavramların tespiti.

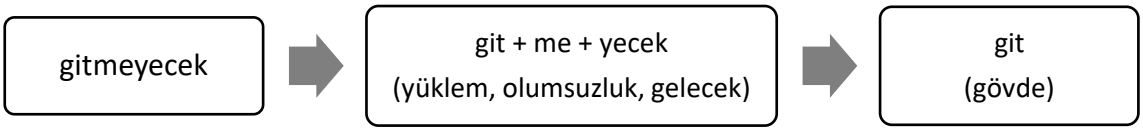
Bilgi çıkarımı süreçleri, sadece yüzeysel metin analizinden ibaret değildir; aynı zamanda metnin anlamsal derinliğini ve tematik yapısını da ortaya koymayı hedefler. Bu nedenle, DDİ tabanlı bilgi çıkarımı çalışmaları, yalnızca teknik bir analiz değil, aynı zamanda içeriksel ve bağlamsal çözümleme anlamına da gelmektedir. Çünkü DDİ ve bilgi çıkarımı, birbirine sıkı sıkıya bağlı iki disiplindir. DDİ'nin sunduğu morfolojik çözümleme, sözdizimsel analiz, anlamsal modelleme gibi araçlar; bilgi çıkarımının temel yapı taşlarını oluşturur. Bu sayede metin içerisindeki kavramlar daha doğru biçimde tanımlanabilir ve metinler arası anlamsal bağlar daha etkili şekilde kurulabilir.

1.2. Gövdeleme ve ASÇ

Gövdeleme, kelimenin kök veya gövdesine indirgenmesi işlemidir [1]. Türkçe gibi çok ekli dillerde bu süreç oldukça karmaşık olabilir. Örneğin:

"koşuyordum" → "koş" (Gövde), "koş" (Kök)

"gözlüksüz" → "gözlük" (Gövde), "göz" (Kök)



Şekil 1.1. Örnek Gövdeleme İşlemi

ASÇ, bir metnin ana fikrini yansıtan sözcük ya da öbeklerin belirlenmesi sürecidir [1]. Bu işlem, bilgi özetleme, belge sınıflandırma ve arama motoru optimizasyonu gibi alanlarda yoğun olarak kullanılmaktadır. ASÇ yapılırken, aşağıdaki iki yaklaşım bir arada kullanılır:

- **Sıklık Tabanlı Yaklaşım:** Sözcük frekansı ve belge frekansı gibi ölçütlere dayanarak belirli kelimelerin önem düzeyleri hesaplanır.
- **Yapısal Yaklaşım:** Cümle uzunluğu, sözcüğün geçtiği bağlam, sözcüğün türü (isim, fiil vb.) gibi yapısal nitelikler dikkate alınır.

Aşağıda örnek bir kısa öyküden çıkarılan anahtar sözcükler yer almaktadır:

Metin: *"Adam, sabah güneşinin altındaki bankta sessizce oturuyordu. Elinde eski bir mektup vardı."*

Anahtar Sözcükler: *adam, sabah, güneş, bank, mektup*

Bu şekilde belirlenen sözcükler, öykünün temel unsurlarını temsil ederken, aynı zamanda anlamı yansıtan bağlamsal veriler de sunar.

1.3. Türkçenin Morfolojik Zorlukları

Türkçe gibi eklemeli ve sondan eklemeli bir dilde, bilgi çıkarımı görevleri özellikle karmaşık hale gelir. Çünkü tek bir sözcük, birçok yapım ve çekim eki alarak bir cümleye eşdeğer hale gelebilir. Bu nedenle Türkçede bilgi çıkarımı sistemlerinin başarısı büyük ölçüde doğru morfolojik çözümleme yapılmasına bağlıdır.

Sözcük kökünün ya da gövdesinin doğru belirlenemediği durumlarda, metin içerisindeki bağlam çözümlenemez ve bilgi çıkarımı hatalı olabilir. Bu durumlar, metin işleme sistemlerinin doğru analiz yapabilmesi için sözcüklerin yalnızca yüzey biçimlerine değil,

iç yapısına da hâkim olmasını zorunlu kılar. Bu bağlamda, kelime köklerinin veya gövdelerinin doğru şekilde tespit edilmesi hem anlam çözümlemesi hem de bilgi çıkarımı açısından kritik bir adımdır.

1.4. Tezin Amacı ve Kapsamı

Türkçedeki klasik yöntemler, önceden tanımlı olan kural tabanlı yapılarla çalışır ve esneklikleri sınırlıdır. Bu tezde, kural tabanlı yapıya istisnai durumlara karşı esneklik sağlayan koşullar ekleyip yeni bir gövdeleme yöntemi geliştirilmiş ve sözcük frekanslarına dayanan bir ASÇ çalışması amaçlanmıştır. Ayrıca literatürde daha önce çalışılmamış kısa öyküler üzerinde test edilerek kullanılabilirliği gösterilmek istenmiştir. Geliştirilen yöntem, anlatı türündeki her türlü metin için uygulanabilir yapıdadır.

2. BÖLÜM

LİTERATÜR TARAMASI

2.1. Gövdeleme için Yapılan Çalışmalar

DDİ uygulamalarında gövdeleme işlemi, yabancı diller için daha çok kelime kökünü bulmayı temsil eder. Türkçe dili için yapılmış bazı çalışmalarda da kelime kökü bulma olarak tanımlansa da gövdeleme işlemi genelde Türkçe için kelimelerin çekim eklerinden arındırılması işlemi olarak tanımlanır. Türkçe için yapılmış olan gövdeleme çalışmaları, çoğunlukla diğer yabancı diller için yapılmış olan yöntemlerin Türkçe için uyarlanmasıyla ortaya çıkmıştır. Her ne kadar farklı uygulamalar olsa da kök bulma yöntemleri, gövdeleme içinde kullanılabildiğinden literatür taramasına dahil edilmiştir. Gövdeleme ve kök bulma işlemleri yöntem bakımından kural tabanlı, istatistiksel ve hibrit olarak sınıflandırılır.

Gövdeleme adına yapılmış ilk çalışmalardan biri 1968 yılında Lovins [2] tarafından İngilizce için yapılmış gövdeleme algoritmasıdır. Bu algoritma bağlama duyarlı ve en uzun eşleşmeye dayalıdır. Kelimelerin son eklerini kaldırmak için iki aşamalı bir sistem kullanılmıştır. İlk aşama gövdeleme algoritması ile köklerin bulunması iken ikinci aşaması köklerin yazım varyasyonlarını ele alır (ör. “absorption” → “absorpt”, “absorbing” → “absorb”). Bu yöntem kural tabanlı yaklaşımın ilk örneklerinden biridir.

Yine İngilizce üzerine 1980 yılında yapılmış kural tabanlı bir yöntem olan Porter’ın [3] geliştirdiği algoritma, basit ve hızlı bir son ek kaldırma algoritmasıdır. Kökler için kural tanımlayarak kök uzunluğunu kontrol eder. Bu algoritma daha sonra 2001 yılında “Snowball” olarak isimlendirilerek açık kaynak olarak yayınlanmıştır. Günümüzde hala temel algoritmalardan biri olarak görülmektedir.

Koskenniemi vd. [4] 1992 yılında, sonlu durum makinesi (SDM) kullanarak İngilizce söz dizimini analiz etmeye yönelik bir sistem geliştirmişlerdir. Her bir dilbilgisi kuralı bir otomata dönüştürülür. Cümleler, sözcük sınırları ve etiket kombinasyonlarını içeren sonlu durum ağlarıyla temsil edilir. Sonuçta sözlük kullanarak kelimelere sözdizimsel etiket atanır.

Krovetz [5] 1993'te, Lovins [2] ve Porter'ın [3] geliştirmiş olduğu yöntemlerdeki sınırlandırmaları aşmak için yeni bir kök bulma algoritması geliştirmiştir. Bu algoritma sözlük kullanarak kökleri belirler. Porter'ın yöntemiyle karşılaştırma yapılmış ve dört farklı test kümesinde Porter'dan daha iyi sonuç vermiştir.

Türkçe için ilk çalışmalardan biri olan Oflazer'in [6] 1994'te yapmış olduğu çalışmada, Koskenniemi vd. [4] çalışmasını temel alarak Türkçe morfolojik analiz için kural tabanlı iki seviyeli bir model geliştirilmiştir. 23 bin kelimeden oluşan sözlük kullanılmıştır. Kelime yapıları çözümlenerek "PC-Kimmo" (Proctor ve Antworth) [7] ortamında uygulanmıştır.

2002 yılında Cebiroğlu ve Adalı'nın [8] yapmış oldukları çalışmada, sözlük kullanmadan kelime köklerini bulmaya odaklanılmıştır. SDM kullanarak kural tabanlı bir yaklaşım sunmuşlardır. Sondan başa doğru ek silimi yapan yöntem yapılan test sonucunda %98'lik bir başarıya ulaşmıştır.

2003 yılında Sever ve Bitirim'in [9] yapmış olduğu çalışma, sözlük ve morfolojik analiz kullanarak kelime köklerini bulmaktadır. Önce kök bulma ardından morfolojik analiz yapılarak kök seçimi gerçekleştirilmiştir. Son adımda ise sesli/sessiz uyumu kuralları uygulanarak %99'luk bir başarıya ulaşmışlardır.

2004 yılında Eryiğit ve Adalı [10] yapmış oldukları çalışma ile Türkçe için sözlük kullanmadan ek ayırma yöntemi geliştirmişlerdir. Çalışma SDM kullanarak eklerin sondan başa doğru sırasıyla çıkarılmasına dayanır. 2002 yılındaki çalışmadan farkı ise morfolojik kuralların tanımlanmasıdır.

Varlıklar'ın [11] 2005 yılında yapmış olduğu tez çalışmasında, kelimeleri önce kökleri daha sonra çekim ekleri ve son olarak yapım ekleri ayrılır. Önceden hazırlanmış kök ve

ek listeleri kullanılır. Cümle ayırma, kelime ayırma ve kelime türü seçme gibi yan işlemler de gerçekleştirilir.

Çilden [12] 2006 yılında daha önce yapılan "Snowball" algoritmasını Türkçe için gövdeleme algoritmasına uyarlamıştır. Sadece isim soylu kelimeler için geliştirilmiştir. Eryiğit ve Adalı'nın [10] çalışmasındaki SDM'ler referans alınmıştır. Sağdan sola ek ayrımı, durdurma sözcüklerini kaldırılması ve ses uyumları kontrol edilmiştir.

Yüret ve Türe [13] 2006 yılında yapmış oldukları çalışmada ise Türkçedeki morfolojik belirsizliği gidermek yani kök ve ekleri ayırmak için yeni bir yöntem geliştirmişlerdir. Kural tabanlı olan bu yöntem denetimli eğitim kullanan yeni bir karar listesi öğrenme algoritmasından oluşmaktadır. Her özellik için ayrı bir modelle %96, tek bir modelle %91'lik bir başarıya ulaşmışlardır.

2007 yılında A. Akın ve D. Akın'ın [14] gelişirmiş olduğu "Zemberek" adlı framework ile Türkçe ve diğer Türk dilleri için sözlük ve ek ağacı yapısı kullanılarak morfolojik analiz yapılmıştır. Oflazer [6] ile Eryiğit ve Adalı'nın [10] çalışmalarını referans alan bu yöntem kök adayları bulur. Ek bilgileri "XML" dosyaları ile yapılandırılmış ve köklerden oluşan bir sözlük oluşturulmuştur. "Zemberek" Türkçe için temel yöntemlerin başında yer alır.

2007 yılında çıkan bir diğer çalışma ise Sak vd. [15] yapmış oldukları çalışmadır. Collins ve Duffy'nin [16] 2002'de yapmış olduğu "Perceptron" algoritmasının DDİ görevlerine uyarlanmasından referans alınarak Türkçe metinlerdeki morfolojik belirsizliği gidermek için "Perceptron" algoritması kullanılmıştır. Temel amaç bir kelimenin birden fazla kök ve ek kombinasyonlarını bağlam bilgisiyle en doğru şekilde seçmektir. Oflazer'in 1994'teki [6] morfolojik çözümleyicisi kullanılarak bu kombinasyonlar bulunmuştur.

Sak vd. [17] 2008 yılında bir diğer çalışmalarında, 2007'de yapmış oldukları "Perceptron" algoritmasına ek olarak "PC-Kimmo" ortamına bağlı kalmayan iki seviyeli morfolojik ayrıştırıcı geliştirmişlerdir. Ayrıştırıcı SDM kullanmaktadır. Çalışma sonucunda 423 milyon kelimelik bir Türkçe web derlemesi oluşturulmuştur.

Birant'ın [18] 2008 yılında yapmış olduğu tez çalışmasında, daha önce yapılan Varlıklar'ın [11] tez çalışması temel alınarak TDK kök listesi ve dilbilimsel kurallar kullanılarak performans açısından geliştirilmiştir (kök ve ek listelerinin bellekte tutulması gibi).

2010 yılında Çöltekin [19] tarafından yapılan çalışmada, "TMorph" adında iki seviyeli morfolojik analiz yapan kök-ek ayrıştırıcısı geliştirilmiştir. Koskenniemi vd. [4] ve Oflazer'in [6] çalışmalarından yola çıkılmıştır. "Zemberek" çalışmasının sözlüğünden yararlanılmış ve ses olaylarına uyarlama gerçekleştirilmiştir.

2011 yılında ise Ak ve Yıldız'ın [20] yapmış olduğu tez çalışmasında, "Rec-TRIE" ve "Reverse-TRIE" algoritmaları geliştirilmiştir. Bu algoritmalar kelimelerin kök ve eklerini ayırmak için ağaç yapısını ve kelime frekanslarını kullanır. Kelime listeleri ağaç yapısına dönüştürülür, kök ve ekler frekans tabanlı olarak belirlenir.

Şahin vd. [21] tarafından 2013 yılında yapılan çalışmada, bayrak işaretleri kullanan iki seviyeli bir morfolojik analizi yöntemi geliştirilmiştir. Sözlük dışı kelimeler için 2004 yılındaki Eryiğit ve Adalı'nın [10] çalışması kullanılmıştır. Morfemlerin geçerli kombinasyonları kısaltılarak bayrak işaretleri kullanılmıştır.

Şahin [22] tarafından 2014 yılında yapılan tez çalışmasında, SDM kullanılarak iki seviyeli bir morfolojik çözümleyici ve bilinmeyen kelime analizcisi geliştirilmiştir. Sözlük ve bayrak işaretleri kullanılarak doğruluk kontrolü sağlanmıştır. Bu yöntem "ITUMorph" olarak adlandırılmıştır. Oflazer'in [6] çalışması ile karşılaştırılıp daha başarılı olduğu gösterilmiştir.

2014 yılında Eryiğit'in [23] yaptığı çalışmada, "ITU NLP" adında herkese açık web servisi yayınlamıştır. Bu serviste daha önce yapılan Şahin'in [21,22] çalışmalarıyla birlikte morfolojik analizcisi ve morfolojik belirsizlik giderici bir araya getirilmiştir.

2018 yılında Tunçelli'nin [24] geliştirmiş olduğu gövdeleme algoritması ile 2006 yılındaki Çilden'in [12] çalışmasına fiilleri ekleyerek yeni bir yöntem oluşturmuştur.

2018 yılında Zafer [25] tarafından yine Zafer'in geliştirdiği "Nuve" deki [26] istatistiksel dil modeli kullanılarak büyük bir sözlük oluşturulmuştur. Temelinde yine

“Snowball” algoritması kullanılmıştır. Fakat bu büyük sözlük sayesinde başarı oranı artmıştır. Bu yöntem “Resha” olarak adlandırılmıştır.

2018 yılında Tintin’in [27] tez çalışmasında, “Gövde-Türk” olarak adlandırılan yeni bir gövdeleme yöntemi geliştirilmiştir. SDM kullanılarak kelimenin sağından soluna doğru çekim eklerinin silimi gerçekleştirilmiştir. Ek seçimleri en uzun eşleşmeye mantığıyla yapılmaktadır. Yapım ekleri kullanılarak ek silinmesi işleminin durdurulması sağlanmıştır. Böylelikle kelime gövdesinden yanlış harf silinmesi önlenmiştir. Sözlükte kelime arama metin ağaçları ile yapılmaktadır. SDM’ler isimler, fiiller ve fiilimsiler için olarak üçe ayrılır fakat daha sonra tek bir SDM haline getirilirler. Eş sesli kelimeler için ayrı bir analiz eklenmiştir. Çalışma sonucunda “Zemberek” ile karşılaştırma yapılmış ve %97 doğruluk oranına ulaşılmıştır.

Can [28] tarafından 2019 yılında yapılan çalışmada, Türkçe için kodlayıcı-kod çözücü (encoder-decoder) modeli geliştirilmiştir. Bu model tekrarlayan sinir ağlarını (LSTM) kullanarak oluşturulmuş ve ilgi mekanizması ise desteklenmiştir. Eğitim veri kümesi olarak kelime ve kök çiftleri kullanılmıştır. “Zemberek” [14], “PC-Kimmo” [6], “Resha” [25] ve “Snowball” [12] ile karşılaştırılan yöntem %85 doğruluk oranı ile “Zemberek” ve “PC-Kimmo” yöntemlerinin gerisinde, “Snowball” ve “Resha” yöntemlerinin üzerindedir.

Ekin [29] tarafından 2020 yılında yapılan tez çalışmasında, Cebiroğlu ve Adalı’nın [8] çalışması temel alınarak fiiller üzerine SDM kullanılarak ek silimi gerçekleştirilmiştir. Ünlü/ünsüz uyumu uygulanmıştır. Ek grupları SDM’ler ile modellenmiştir. “Zemberek” [14] ve “TMorph” [19] ile karşılaştırılmıştır. Fiillerde %95 başarı oranı elde etmiştir.

Özbey ve Karcılı [30] tarafından 2021 yılında yapılan çalışma ise kökler ve ekler için oluşturulan iki sözlüğü kullanarak kök ve ekleri istatistiksel olarak ayrıştırır. Kelimeler olasılık tabanlı güncellemelerle optimize edilir. Kök ve eklerin en olası kombinasyonları “MDL” prensibiyle seçilir.

2022 yılında Aleçakır vd. [31] tarafından yapılan çalışmada, Can’ın [28] çalışması temel alınarak kodlayıcı-kod çözücü (encoder-decoder) yapısına sahip BLSTM ve ilgi mekanizması kullanarak gövdeleme modeli geliştirilmiştir. %85 doğruluk oranı elde etmiştir.

2.2. ASÇ için Yapılan Çalışmalar

DDİ uygulamalarından biri olan ASÇ birkaç yöntem ile gerçekleştirilir. En çok kullanılanlar istatistiksel olarak metin içerisinde birden fazla tekrar eden kelimelerin bulunması yani frekans analizi gerçekleştirilmesi, diğer yöntemler ise TF-IDF ile n-gram yöntemi yani Terimlerin belgelerdeki sıklığı ve nadirliği dikkate alınarak önem puanları hesaplanması ve n-gramlar ile kombinasyonlar oluşturularak gerçekleştirilmiştir.

2005 yılında Özgür vd. [32] tarafından yapılan çalışmada, "Corpus-Based" ve "Class-Based" olarak iki farklı yaklaşım ile ASÇ yapılmıştır. "Corpus-Based" yaklaşımında, tüm belgelerde en yüksek TF-IDF puanına sahip kelimeler seçilir. "Class-Based" yaklaşımında ise her sınıf için ayrı ayrı TF-IDF puanları hesaplanır ve sınıfa özgü kelimeler seçilir.

2007 yılında ise Külekci [33] tarafından yapılan çalışmada, metin dosyalarında çoklu sabit uzunluklu desenlerin aranması için tasarlanmış bir algoritma olan "TARA" tanıtılmaktadır. ASÇ doğrudan bir hedef olmasa da algoritma belirli desenlerin metin içinde hızlı ve verimli bir şekilde tespit edilmesini sağlar. Özellikle, Türkçe gibi karmaşık karakter kümesine sahip dillerde bile etkili çalışabilmesi için optimizasyonlar içerir. TARA algoritması, bit-paralellik ve pencereli tarama tekniklerini birleştirerek çoklu desen eşleştirme problemini çözer.

Kocatekin ve Ünay [34] 2013 yılında yaptıkları çalışmada, radyoloji raporlarındaki başlık ve altlık alanları çıkarılarak kelimelerin frekansları hesaplanır. Hesaplanan frekans değerlerine göre bulunan anahtar sözcüklere her anatomik bölge için elle belirlenmiş ayırıcı kelimeler de eklenerek nihai anahtar sözcükler belirlenmiş olur. Yapılan test sonucunda %84'lük bir başarı elde edilmiştir.

Özyirmidokuz [35] tarafından 2014 yılında yapılan çalışmada ise Türkçe ekonomi haberlerinden bilgi çıkarmak için metin madenciliği teknikleri kullanılmıştır. Öncelikle "Snowball" [3] algoritması tabanlı bir yöntem ile kelime gövdeleri bulunup TF-IDF yöntemi ile ASÇ yapılmıştır. Bulunan anahtar sözcükler ile dokümanlar arası benzerlik analizi yapılmıştır.

İkikat vd. [36] tarafından 2015 yılında yapılmış olan çalışmada ise Wikipedia sayfalarındaki kelime ve kelime öbeklerinin otomatik olarak bağlantılandırılması için ASÇ yapılmıştır. Türkçe için “Zemberek” [14] İngilizce için “WordNet” yöntemleri kullanılarak kelime kökleri bulunur. Daha sonra TF-IDF ve “Key-Phrasses” yöntemleriyle ASÇ gerçekleştirilir.

2017 yılında Savaş ve Topaloğlu‘nun [37] çalışmasında, TÜİK verilerine göre suç kategorileri belirlenmiş, “Zemberek” [14] kullanılarak tweet'ler köklerine ayrıştırılarak frekans analizi yapılmıştır. Frekans analizi ile suç trendlerinin gerçek zamanlı tespiti sağlanmıştır.

2018 yılında Müngen ve Kaya [38] tarafından yapılan çalışmada, akademik makalelerde eksik olan özet ve anahtar kelimelerin otomatik olarak çıkarılması için bir yöntem önerilmiştir. “Zemberek” [14] kullanılarak kelime kökleri bulunup TF-IDF ile ASÇ gerçekleştirilerek %93’lük başarıya ulaşılmıştır.

2019 yılında Demirci vd. [39] tarafından yapılan çalışma ile duygu analizi bağlamında, metinlerdeki önemli kelimelerin belirlenmesi ve vektörleştirilmesi aşamalarında TF-IDF ve “Bag-of-Words” gibi yöntemler, metinlerdeki önemli kelimelerin ağırlıklandırılmasını sağlamıştır.

Aygün ve Kaya [40] tarafından 2020’de yapılan çalışma ise iki katmanlı bir yaklaşım gerçekleştirmiştir. TF-IDF ve n-gram ile istatistiksel bir yaklaşım iken Word2Vec ile terim adaylarının anlamsal yakınlığı analiz edilerek anlamsal yaklaşım sağlanmıştır. Terimlerin %58'i alana özgü olarak başarıyla tespit edilmiştir.

Erzurumlu [41] tarafından 2022 yılında yapılan tez çalışmasında, konuşma verilerinden konuşmadan metne dönüşümü yapıldıktan sonra ASÇ gerçekleştirilmiştir. Merkez tabanlı dinamik kümeleme algoritması geliştirilmiş ve “RAKE”, “YAKE!”, TF-IDF gibi yöntemlerle karşılaştırma yapılmıştır. Geliştirilen sistem, özellikle gerçek zamanlı konuşma verileri için yüksek doğrulukla çalışabildiğini göstermiştir.

Ayan vd. tarafından [42] 2021 yılında yapılan çalışmada, derin öğrenme tabanlı yaklaşımla “Sentence-BERT” modeli kullanılarak ASÇ gerçekleştirilmiştir. “Word2Vec” yöntemi ile karşılaştırılmış %72,9 duyarlılık, %26,6 kesinlik, %39 F1 skoru elde etmiştir.

Gürbüz ve Uluyol [43] tarafından 2022 yılında yapılan çalışmada, araştırmacıların makalelerinin özet bölümlerinden ASÇ için TF-IDF yöntemini kullanılmıştır. Her makalenin özetinden TF-IDF değeri en yüksek olan 100 kelime seçilerek anahtar sözcükler belirlenmiştir. Çalışma sonucunda hem İngilizce hem de Türkçe metinlerde etkili bir şekilde ASÇ yapılabilmiştir.

Göz [44] tarafından 2022 yılında yapılan tez çalışmasında ise Türkçe akademik makalelerde ASÇ için üç yeni yöntem olan "MGRank", "SkyWords", "SkyRank" açıklanmıştır. "MGRank", çizge tabanlı, denetimsiz bir yöntem. Çok kenarlı tam çizge modeli kullanır ve pencere boyutu parametresine ihtiyaç duymaz. "SkyWords", denetimli ve denetimsiz öğrenmeyi birleştiren hibrit bir yöntemdir. "Skyline" operatörü ve çoğunluk oylama prensibini kullanır. "SkyRank", denetimsiz, istatistiksel bir yöntemdir. "Skyline" operatörü ile baskın özelliklere sahip aday anahtar kelimeleri seçer. Geliştirilen yöntemler, mevcut yöntemlerle karşılaştırıldığında daha yüksek performans göstermiştir.

2023 yılında Kılıç [45] tarafından yapılan tez çalışmasında, ASÇ problemi için hibrit bir model önermektedir. ASÇ'yı model, metin, grafik, gömme ve küme özniteliklerini birleştirerek dizi etiketleme görevi olarak ele alır. "HibritAKÇ" yöntemi en yüksek 0.74 F1 skoru elde etmiştir.

2023 yılında Kat [46] tarafından yapılan çalışmada ise TÜBİTAK'a sunulan mühendislik alanındaki proje önerilerinin otomatik olarak işlenmesi, ASÇ yapılması, metinler arası benzerliklerin tespit edilmesi ve projelerin uygun alt alanlara sınıflandırılması için bir karar destek sistemi geliştirmeyi amaçlamaktadır. Belgeler ön işleme adımlarından geçtikten sonra frekans analizi ile tekli, ikili ve üçlü n-gramlar ile ASÇ gerçekleştirilir. Projelerin anahtar terimleri başarıyla çıkarılmıştır. Özellikle çoklu gramlar, projelerin kavramsal içeriğini daha iyi yansıtmıştır.

2024 yılında Yayla [47] tarafından yapılan çalışma, 1995-1999 arasında Türk Kütüphaneciliği dergisinde yayımlanan ve anahtar kelimesi olmayan 58 makalede "YAKE!" Yöntemiyle ASÇ gerçekleştirilmiştir. Çalışma sonucunda teknik olarak yüksek doğrulukta anahtar kelime atadığı ancak bağlamsal geçerliliğin insan uzmanlarla tam uyumlu olmadığı belirtilmiştir.

Yine 2024 yılında Aydın ve Kantarcı [48] tarafından yapılan çalışmada ise Türkçe metinlerden anahtar sözcük çıkarımı için iki farklı derin öğrenme modeli kullanılmıştır. “LSTM Tabanlı Seq2Seq” model ile ve “BERT Tabanlı Seq2Seq” modeli karşılaştırılmış sonuçta Rogue-1 F1 skorunda 0.39 ile BERT tabanlı model, dönüştürücü mimarisi sayesinde LSTM tabanlı modele kıyasla daha başarılı sonuçlar vermiştir.

2024 yılında yapılmış bir diğer çalışma ise Yusuf ve Saraçlar [49] tarafından gerçekleştirilmiştir. Çalışmada, konuşma içerisinde anahtar kelime arama görevini geliştirmek için yazılı metinlerin kullanıldığı bir yöntem önerilmektedir. Geleneksel yöntemler, otomatik konuşma tanıma sistemlerinin çıktılarını kullanırken, bu çalışmada uçtan uca bir yaklaşım benimsenmiştir. Çalışmada, “JOSTER” adlı bir model önerilmiştir. Bu model, iki ana görevi birlikte öğrenir; konuşma belgesi içinde anahtar kelime arama ve yazılı belge içinde anahtar kelime arama. Önerilen “JOSTER” modeli, özellikle düşük kaynaklı diller ve alan uyarlama senaryolarında etkili bir çözüm sunmuştur.

2025 yılında Babayiğit ve Sattuf [50] tarafından yapılmış son çalışma ise doğrudan Türkçe metinlerden ASÇ için BERT tabanlı bir model önermektedir. 128.715 akademik tez özeti ve yazar tarafından belirlenmiş anahtar kelimeler ile eğitim yapılmış ve %97,77 F-skoru ile %97,49 doğruluk yüzdesi elde etmişlerdir.

Literatür incelendiğinde Türkçe için yapılmış kök bulma ve gövdeleme çalışmalarının kullandığı kural tabanlı yaklaşımın daha başarılı olduğu görülmüştür. Fakat yapılan çalışmaların esnekliği kısıtlı olduğu saptanmıştır. Ayrıca Türkçe dil yapısının bazı kuralları bu yöntemlere tamamen eklenmediği ve eklenmiş olanlarında hatalı olduğu gözlemlenmiştir. Bu tez çalışmasında, gövdeleme yönteminin kural miktarı artırılarak özel durumlara karşı esneklik sağlamak ve başarı oranını yükseltmek hedeflenmiştir. Türkçe kısa öykülerin ise daha önce hiçbir çalışmada kullanılmaması literatürde görülen bir diğer eksikliktir. Türkçe kısa öyküler üzerinde gövdeleme ve ASÇ uygulanarak küçük yaştaki çocuklar için öykülerin içeriği hakkında bilgi çıkarımı sağlanmıştır. Böylelikle dolaylı olarak içeriği çocuklara uygun olmayan öykülerin tespit edilmesine de katkı sağlanacağı düşünülmüştür. Kural tabanlı yaklaşıma sahip olan bu yöntem için SDM ve çekim eklerinin sınıflandırılmasında Tintin [27] tarafından yapılmış “Gövde-Türk” yöntemi ve ASÇ için Kat’in [46] gerçekleştirdiği çalışma temel alınmıştır.

3. BÖLÜM

MATERYAL VE YÖNTEM

3.1. Materyal

3.1.1. Türkçede Kelime Türleri

Türkçede kelime türleri, geleneksel dilbilgisi sınıflandırmalarında isim, fiil, sıfat, zarf, zamir, edat, bağlaç, ünlem gibi çeşitli kategorilere ayrılsa da dilin biçimsel yapısı ve kökensel sözcük üretim süreçlerine göre, temelde isim kökenli ve fiil kökenli olarak iki ana grupta toplanır [51]. Diğer tüm kelime türleri ya isimlerin ya da fiillerin türevleri olarak değerlendirilir. Bu yaklaşım, özellikle yapım eklerinin işlevi ve kelime kökü odaklı çözümlemelerde daha işlevsel ve sistematiktir. Örneğin, *"güzelce"* zarfı *"güzel"* sıfatından, o da *"güzellik"* kavramından türetilmiş bir isim kökenli sözcüktür; benzer şekilde *"yürüyüş"*, *"yürümek"* fiilinden türeyerek isimleşmiştir. Dolayısıyla, Türkçedeki sözcük türlerini biçimsel bir bakışla anlamlandırmak için öncelikle isim ve fiil köklerinin yapısını ve bu köklerden türeyen biçimlerin işleyişini kavramak gerekmektedir.

3.1.1.1. İsimler

İsim kökenli sözcükler, Türkçede bir nesneyi, kavramı, canlıyı ya da durumu karşılayan isim köklerinden türetilen kelimeleri kapsar [51]. Bu kökler, dilin en temel adlandırma işlevini yerine getirir ve diğer türdeki sözcüklerin üretiminde de kaynak rolü oynar. İsim kökleri, çekim ekleriyle çoğul (ör. "kitaplar"), iyelik (ör. "evim", "arabanız") ya da hâl (ör. "okula", "evden") bilgisi kazanabilir. Aynı zamanda yapım ekleriyle yeni anlamlar taşıyan türemiş sözcükler de oluşturabilirler. Örneğin:

"susuz" (Sıfat) → "su" (İsim Kökü) + "-suz" (Çekim Eki)

"suluk" (Türemiş İsim) → "su" (İsim Kökü) + "-luk" (Yapım Eki)

Bu türden türetmeler, isim kökünün yalnızca adlandırıcı işlevini değil, aynı zamanda nitelik bildirici ya da nesne tanımlayıcı işlevlerini de ortaya çıkarır. İsim kökenli sözcükler, sıfatlara, zarflara ya da hatta yeni isimlere dönüşebilir.

İsim köklerinden türeyen sözcükler, DDİ uygulamalarında sözcük türü tanımlama, anlam sınıflandırma ve ASÇ gibi görevlerde temel bileşenler olarak değerlendirilir. Özellikle morfolojik çözümleme açısından, bir kelimenin isim kökenli olup olmadığını belirlemek, dilin yapısal analizinde yüksek öneme sahiptir. İsim kökenli sözcükler sekiz ayrı türden oluşur. Bunlar; isim, sıfat, zamir, zarf, edat, bağlaç ve ünlemdir.

3.1.1.1.1. İsim

İsimler, Türkçede canlı, cansız varlıkları, kavramları, yerleri ve soyut olguları karşılayan temel kelime türüdür. İsimler yalın halde bulunabileceği gibi; çoğul (ör. “evler”), iyelik (ör. “evim”), hâl (ör. “eve”, “evde”, “evden”) ve tamlama (ör. “kitap kapağı”) ekleri alarak çeşitli dilbilgisel işlevler üstlenebilir. Cümlede özne, nesne, dolaylı tümleç gibi görevlerde kullanılabilirler. Biçimsel açıdan isim kökleri, çok sayıda yeni sözcük türünün (ör. sıfat, zarf, fiilimsiler) türemesinde temel yapı taşlarını oluşturur [51].

3.1.1.1.2. Sıfat

Sıfatlar, isimleri nitelik veya nicelik yönünden belirten kelimelerdir. Niteleme sıfatları (ör. “yeşil elbise”, “uzun yol”) ve belirtme sıfatları (ör. “bu kalem”, “üç kişi”) olmak üzere iki temel gruba ayrılır. Yapı bakımından yalın sıfatlar olduğu gibi, isim köklerinden türetilmiş (ör. “sulu”, “güçlü”) ya da birleşik (ör. “kara kuru”, “çalışkan biri”) sıfatlar da bulunur. Sıfatlar, tek başına yüklem olamaz ancak isimlerle birlikte kullanıldığında ad tamlamaları oluşturur [51].

3.1.1.1.3. Zamir

Zamirler, varlıkların ya da kavramların isimlerini doğrudan belirtmek yerine, onları işaret eden kelimelerdir. Kişi (ör. “ben”, “sen”, “o”), işaret (ör. “bu”, “şu”, “o”), belgisiz (ör. “biri”, “herkes”), soru (ör. “kim”, “ne”) ve dönüşlülük (ör. “kendi”) gibi türleri vardır. Zamirler, bağlama bağlı olarak anlam kazandıkları için, bağlamdan kopuk olduklarında anlamsal belirsizlik yaratabilirler [51].

3.1.1.1.4. Zarf

Zarflar, fiilleri, sıfatları, diğer zarfları ya da tüm cümleyi nitelendiren kelimelerdir. Zaman (ör. "bugün", "hemen"), yer (ör. "içeri", "dışarı"), miktar (ör. "çok", "az"), durum (ör. "yavaşça", "hızlıca") ve soru (ör. "ne zaman", "nasıl") zarfları olmak üzere sınıflandırılır. Zarflar da genellikle isim veya sıfat köklerinden türetilir (ör. "hızlıca", "kısaca"*)*. Zarf kullanımı, eylemin bağlamını ve niteliğini belirler [51].

3.1.1.1.5. Edat

Edatlar, tek başlarına anlam taşımayan ancak kendilerinden önceki sözcüklerle anlam ilişkisi kuran kelimelerdir. *"için", "ile", "gibi", "göre", "kadar"* gibi edatlar, anlamı genişletme, yönlendirme ya da benzetme işlevi üstlenirler. Edatlar her zaman bir başka sözcüğe bağlı çalıştıkları için bağımlı dil öğeleridir. Örneğin *"senin için"*, *"ev gibi"*, *"bize göre"* gibi kullanımlarda sözcükler arası mantıksal bağ kurarlar [51].

3.1.1.1.6. Bağlaç

Bağlaçlar, sözcükleri, sözcük gruplarını veya cümleleri birbirine bağlayan kelimelerdir. En yaygın bağlaçlar arasında *"ve", "ya da", "ama", "çünkü", "lakin", "hatta"* gibi örnekler yer alır. Bağlaçlar sayesinde cümlede anlam ilişkileri (ör. neden-sonuç, karşıtlık, seçenek) kurulur. Özellikle çok cümleli yapılarda bağlaçlar, metinlerin mantıksal yapısını anlamlandırmada belirleyicidir [51].

3.1.1.1.7. Ünlem

Ünlemler, duyguları, sesleri ya da tepkileri genellikle doğrudan ifade eden kelimelerdir. *"ah", "oh", "eyvah", "bravo", "tüh", "yaşasın"* gibi sözcükler bu gruba girer. Ünlemler çoğunlukla cümle dışı ögelerdir, yüklem ya da özneleri olmadan kullanılırlar. Yapı bakımından sabit ve sınırlı bir sözcük grubunu oluştursalar da özellikle konuşma dili veya yazılı metinlerdeki duygusal tonlamayı yansıtmada önemli bir yer tutarlar [51].

3.1.1.2. Fiiller

Fiil kökenli sözcükler, Türkçede bir eylemi, oluşu ya da durumu bildiren fiil köklerinden türeyen kelimeleri kapsar [51]. Fiil kökleri, yüklem işleviyle cümlede temel

çekirdek öğeyi oluşturur ve zaman, kip, kişi gibi çekim ekleriyle biçimlenerek geniş bir anlamsal ve işlevsel çeşitliliğe kavuşur. Örneğin, "*gel"* fiil kökü; "*geldi"*, "*gelecek"*, "*gelmişti"* gibi biçimlerde zaman ve kişi bilgisiyle zenginleşir. Fiil kökenli sözcükler yalnızca çekimle değil, aynı zamanda isimleştirme yoluyla da yeni kelimelere kaynaklık eder. Örneğin:

"yüzmek" → "yüzücü", "yüzme", "yüzüş"

Bu dönüşüm sırasında fiil köküne gelen yapım ekleri (ör. "-me", "-iş", "-ici") sözcüğün türünü ve anlamını değiştirerek isim ya da sıfata çevirebilir. Böylece "*koşmak"* fiilinden "*koşu"*, "*koşucu"* gibi farklı işlevlere sahip sözcükler türetilebilir [51]. Fiil kökenli sözcüklerin Türkçedeki üretkenliği, sunduğu esneklik sayesinde oldukça yüksektir. Bu durum, DDİ bağlamında hem morfolojik çözümleme hem de anlamsal yorumlama açısından önem taşır. Özellikle bilgi çıkarımı süreçlerinde, fiil köklerinin doğru şekilde tanımlanması; olay, durum ya da eylem temelli anlam yapılarının kurulabilmesi açısından kritiktir. Bu nedenle fiil kökenli sözcüklerin tanınması, morfolojik ayrıştırma sistemlerinin ve dil modellemelerinin temel yapı taşlarından biridir.

3.1.1.3. Fiilimsiler

Fiilimsiler, Türkçede fiil kök veya gövdelerinden türetilen ancak fiil olma özelliklerini kaybederek isim, sıfat ya da zarf görevinde kullanılan özel sözcük türleridir. Fiil kökenli olmalarına rağmen, çekim ekleriyle zaman, kişi, kip almazlar; bunun yerine yapım ekleri aracılığıyla cümlede adlaşmış ya da niteleyici işlevler üstlenirler. Üç ana gruba ayrılırlar: isim-fiiller (mastarlar), sıfat-fiiller (ortaçlar) ve zarf-fiiller (ulaçlar) [51].

- **İsim-fiiller:** Fiili isimleştirerek nesne ya da kavram haline getirir. Örneğin: "*yüzmek"*, "*gülme"*, "*alışveriş"*.
- **Sıfat-fiiller:** Fiili sıfatlaştırarak niteleme işlevi görür. Örneğin: "*gelen yolcu"*, "*yaptığı iş"*, "*gidecek adam"*.
- **Zarf-fiiller:** Fiili zarf yaparak cümlede zaman, neden, biçim gibi durumları belirtir. Örneğin: "*gelince konuşuruz"*, "*koşarak geldi"*, "*gitmeden önce"*.

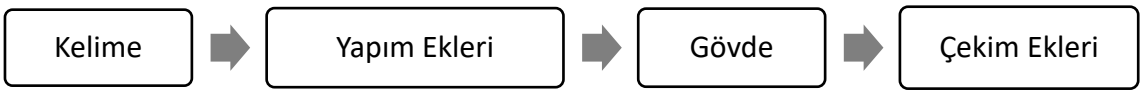
Fiilimsiler, cümlede yüklem olamaz; ancak yüklem dışı hemen her görevde bulunabilirler. Bu yönleriyle hem fiil kökenli hem de isim/sıfat/zarf işlevli hibrit yapılar olarak tanımlanabilirler. Morfolojik olarak bir fiilin türemiş biçimleri olan fiilimsiler, Türkçenin sözcük üretim sisteminde önemli yer tutar. DDİ açısından fiilimsiler; sözcük türü tanıma, anlamsal rol çözümleme ve cümle ögelerinin belirlenmesi gibi görevlerde karmaşık ancak zengin bilgi taşıyan yapılardır.

Örneğin, *"Çalışarak kazandı."* cümlesinde "*çalışarak"* zarf-fiil yalnızca eylemin biçimini değil, aynı zamanda nedenini de ima eder. Bu tür yapılar, özellikle Türkçe gibi eklemeli dillerde anlam çıkarımı için kritik rol oynar.

3.1.2. Türkçede Kelime Yapısı

Türkçenin yapı bakımından en ayırt edici özelliklerinden biri, eklemeli bir dil olmasıdır. Bu özellik, sözcüklerin köklerine çeşitli eklerin getirilmesiyle yeni anlamların ve dilbilgisel işlevlerin kazanılmasını sağlar. Türkçede bir kelime, çoğunlukla bir kök ile başlar ve bu köke, anlam ya da görev değişimi yaratan yapım ekleri ve cümle içindeki işlevini belirleyen çekim ekleri eklenerek genişletilir [51]. Bu eklemeli yapı sayesinde, tek bir kökten onlarca farklı kelime ve biçim türetilebilir; bu da Türkçenin sözcük üretiminde oldukça verimli ve esnek bir sistem sunduğunu gösterir. Sözcüklerin bu biçimde aşamalı olarak türetilmesi hem anlamsal çeşitliliğin hem de dilin yapısal derinliğinin temelini oluşturur.

Kelime yapısını oluşturan ögelerin (kök, gövde, ek) doğru biçimde tanınması, yalnızca dil bilgisinin anlaşılması açısından değil, aynı zamanda DDİ uygulamaları için de büyük önem taşır. Bu çerçevede, Türkçede kelime yapısının bileşenleri olan kökler, gövdeler ve ekler, morfolojik çözümleme ve bilgi çıkarımı süreçlerinin temel yapı taşları olarak değerlendirilmektedir.



Şekil 3.1. Türkçede Kelime Yapısı

3.1.2.1. Kökler

Kök, bir kelimenin anlamlı ve bölünemez en küçük parçasıdır. Bir sözcüğün yapı bakımından temelini oluşturur ve kendi başına anlam taşıyabilir. Türkçede kökler, isim kökü ve fiil kökü olmak üzere ikiye ayrılır [51]. Örneğin *"evlerimizde"* kelimesinin kökü *"ev"*, *"gidiyordum"* kelimesinin kökü ise *"git"* tir. Kökler, daha sonra yapım ve çekim ekleri aracılığıyla yeni sözcükler veya sözcük biçimleri oluşturmak için temel alınır. DDİ süreçlerinde kök çıkarımı veya kök bulma algoritmalarının doğruluğu, dilin kök yapısının doğru anlaşılmasına bağlıdır.

3.1.2.2. Gövdeler

Gövde, bir köke yapım eki eklendikten sonra ortaya çıkan ve yeni bir anlam taşıyan ara yapıdır. Gövde, hâlâ başka ekler alabilecek durumdadır ve yeni sözcüklerin ya da biçimlerin üretiminde temel olabilir [51]. Örneğin:

"gözlük" (Gövde) → "göz" (Kök) + "-lük" (Yapım Eki)

"gözlükçü" (Gövde) → "gözlük" (Gövde) "-çü" (Yapım Eki)

Bu açıdan, gövdeler üretici bir ara düzey olarak değerlendirilebilir. Gövde tanıma işlemleri, özellikle türevli kelimelerin ayrıştırılmasında ve anlam ilişkilerinin analizinde kritik öneme sahiptir.

3.1.2.3. Ekler

Ekler, kök ya da gövdeye eklenerek ya yeni sözcükler türeten ya da sözcüklerin cümle içindeki işlevlerini belirleyen morfolojik birimlerdir [51]. Eklerin doğru belirlenmesi, özellikle Türkçede yoğun biçimde görülen ek zincirlerinin ayrıştırılmasında ve morfolojik çözümlemede belirleyici bir etkendir. Türkçede ekler yapım eki ve çekim eki olarak iki ana gruba ayrılır.

Tablo 3.1. Gövde Oluşumları

Örnek Kelime	Kök	Yapım Eki	Gövde	Çekim Eki	Açıklama
evlerimiz	ev	-	ev	-ler + -imiz	“ev” isim kökü, çoğul, iyelik ve yer hâli ekleri almış.
gözlükçü	göz	-lük + -çü	gözlükçü	-	“göz” köküne yapım ekleriyle yeni anlamlar kazandırılmış.
yazıcılar	yaz	-ıcı	yazıcı	-lar	“yaz” fiil kökü yapım ekiyle isimleşmiş çoğul eki almış.
gitmeyecek	git	-	git	-me + -yecek	Fiil kökü, olumsuzluk ve gelecek zaman ekleri almış.
okunabilir	oku	-n + -abilir	okunabilir	-	“oku” fiil kökü edilgen ve yeterlilik yapım ekleriyle türemiş.
kitapçıda	kitap	-çı	kitapçı	-da	“kitap” isim kökü, meslek bildirir yapım eki ve yer hâli eki almış.

3.1.2.3.1. Yapım Ekleri

Kökten yeni bir kelime türetir. Örneğin *"yaz"* kökünden *"yazıcı"*, *"yazılı"* gibi kelimeler yapım eki aracılığıyla oluşturulur. Geldiği kelimede anlam değişimi yaratır. Yapım ekleri geldikleri köklerin ve bu köklerden ürettikleri gövdelerin türüne göre dörde ayrılır [51].

İsimden İsim Üreten Yapım Ekleri: Bu ekler, isim kök veya gövdelerinden yeni bir isim türetir. Türetilen kelime genellikle asıl kökle anlam ilişkisi taşır, ancak farklı bir kavrama karşılık gelir. Örnek olarak *"-lık/-lik"*, *"-cı/-ci"*, *"-daş"*, *"-lı"*, *"-sız"*, *"-sal"* ekleri verilebilir [51].

Tablo 3.2. Örnek İsimden İsim Üreten Yapım Ekleri

Temel Kelime	Yapım Eki	Yeni Kelime	Anlamı
su	-luk	suluk	su konulan kap
kitap	-çı	kitapçı	kitap satan kişi
kum	-sal	kumsal	kumu olan yer

İsimden Fiil Üreten Yapım Ekleri: Bu ekler, isim kök veya gövdelerinden fiil türetir. Eylem bildiren yeni bir anlama sahip kelime oluşur. Örnek olarak *"-la/-le"*, *"-msa/-mse"*, *"-kır/-kir"*, *"-sa/-se"* ekleri verilebilir [51].

Tablo 3.3. Örnek İsimden Fiil Üreten Yapım Ekleri

Temel Kelime	Yapım Eki	Yeni Kelime	Anlamı
temiz	-le	temizle	temiz hale getirmek
hay	-kır	haykır	bağırmak
önem	-se	önemse	önemli görmek

Fiilden İsim Üreten Yapım Ekleri: Bu tür yapım ekleri, fiil köklerinden ya da gövdelerinden isim türetir. Türetilen kelime çoğunlukla eylemin sonucu, nesnesi, aracı ya da yeriyle ilgilidir. Örnek olarak *"-gi/-gı"*, *"-inç"*, *"-geç"*, *"-kan"*, *"-im"*, *"-ak"* ekleri verilebilir [51].

Tablo 3.4. Örnek Fiilden İsim Üreten Yapım Ekleri

Temel Kelime	Yapım Eki	Yeni Kelime	Anlamı
biç	-im	biçim	dıştan görünüş
sev	-gi	sevgi	yakınlık, ilgi
gül	-ünç	gülünç	komik olan, tuhaf

Fiilden Fiil Üreten Yapım Ekleri: Bu ekler, bir fiilden anlam farklılığına sahip yeni bir fiil türetir. Fiilin yönünü, biçimini, etkenliğini ya da niteliğini değiştirebilir. Örnek olarak *"-dır/-dir"*, *"-er/-ar"*, *"-in/-ın"*, *"-ş"*, *"-ele/-ala"* ekleri verilebilir [51].

Tablo 3.5. Örnek Fiilden Fiil Üreten Yapım Ekleri

Temel Kelime	Yapım Eki	Yeni Kelime	Anlamı
bil	-dir	bildir	başkasına iletmek
çık	-ar	çıkar	elde edilen kazanç
gör	-ün	görün	görünür olmak

3.1.2.3.2. Çekim Ekleri

Sözcüğün anlamını değiştirmeden, onun cümle içindeki görevini belirler. Bu ekler çoğunlukla sözdizimsel işlemede ve ASÇ'de dikkate alınır. İsim, fiil ya da fiilimsi köklerine veya gövdelerine gelmesine göre üçe ayrılırlar [51].

İsimlere Gelen Çekim Ekleri: Türkçede isimlere gelen çekim ekleri, kelimelerin kök veya gövdelerine gelerek o kelimenin cümle içerisinde diğer kelimelerle ilişkiler kurmasını sağlayan, fakat yeni bir kelime üretmeyen ya da gelmiş oldukları kelimelerin anlamlarını değiştirmeyen eklerdir.

Bu eklerin isim soylu kelimelere gelmesiyle o kelimelerin cümle içerisinde bazı görevlere sahip olmasını sağlarlar. Bu görevler, çoğulluk, iyelik, durum/hâl, ilgi, zaman, şahıs ve zarf yapma görevleridir [51].

Çoğulluk Eki, isimleri çoğul hale getirir. Örneğin:

“elmalar” (İsim) → “elma” (İsim Gövdesi) + “-lar” (Çoğulluk Eki)

İyelik ekleri, bir isme aitlik kazandırır. Altı farklı şahıs için çekimlenir:

“evim” (İsim) → “ev” (İsim Kökü) + “-im” (1.Tekil Şahıs İyelik Eki)

“araban” (İsim) → “araba” (İsim Gövdesi) + “-n” (2.Tekil Şahıs İyelik Eki)

“kitabı” (İsim) → “kitap” (İsim Kökü) + “-ı” (3.Tekil Şahıs İyelik Eki)

“yuvamız” (İsim) → “yuva” (İsim Kökü) + “-mız” (1.Çoğul Şahıs İyelik Eki)

“masanız” (İsim) → “masa” (İsim Kökü) + “-nız” (2.Çoğul Şahıs İyelik Eki)

“balonları” (İsim) → “balon” (İsim Gövdesi) + “-ları” (3.Çoğul Şahıs İyelik Eki)

Durum veya hâl ekleri, ismin cümledeki görevini (özne, nesne, yer vs.) gösterir. Yalın, belirtme, yönelme, bulunma, ayrılma olarak beş ayrı türe sahiptir. Örneğin:

“kalem” (İsim) → “kalem” (İsim Gövdesi) + “-” (Yalın Hal Eki)

“dolabı” (İsim) → “dolap” (İsim Gövdesi) + “-ı” (Belirtme Hal Eki)

“okula” (İsim) → “okul” (İsim Gövdesi) + “-a” (Yönelme Hal Eki)

“otobüste” (İsim) → “otobüs” (İsim Gövdesi) + “-te” (Bulunma Hal Eki)

“uçaktan” (İsim) → “uçak” (İsim Gövdesi) + “-tan” (Ayrılma Hal Eki)

İlgi eki, bir ismin başka bir isme ait olduğunu belirtir. Zarf yapmaz, sıfat veya zamir oluşturur. Örneğin:

“benimki” (İsim) → “benim” (İsim Gövdesi) + “-im” (İyelik Eki) + “-ki” (İlgi Eki)

“sabahki” (İsim) → “sabah” (İsim Gövdesi) + “-ki” (İlgi Eki)

Ekfiil ekleri, isimlere yüklem olma, zaman ve kip ekleri getirme işlevi kazandırır. Örneğin:

“mutluydu” → “mutlu” (İsim Gövdesi) + “-ydu” (Ekfiil Eki)

“güzelse” → “güzel” (İsim Gövdesi) + “-se” (Ekfiil Eki)

Şahıs ekleri, isim cümlelerinde yüklem görevinde olan kelimeye kişi katar. Örneğin:

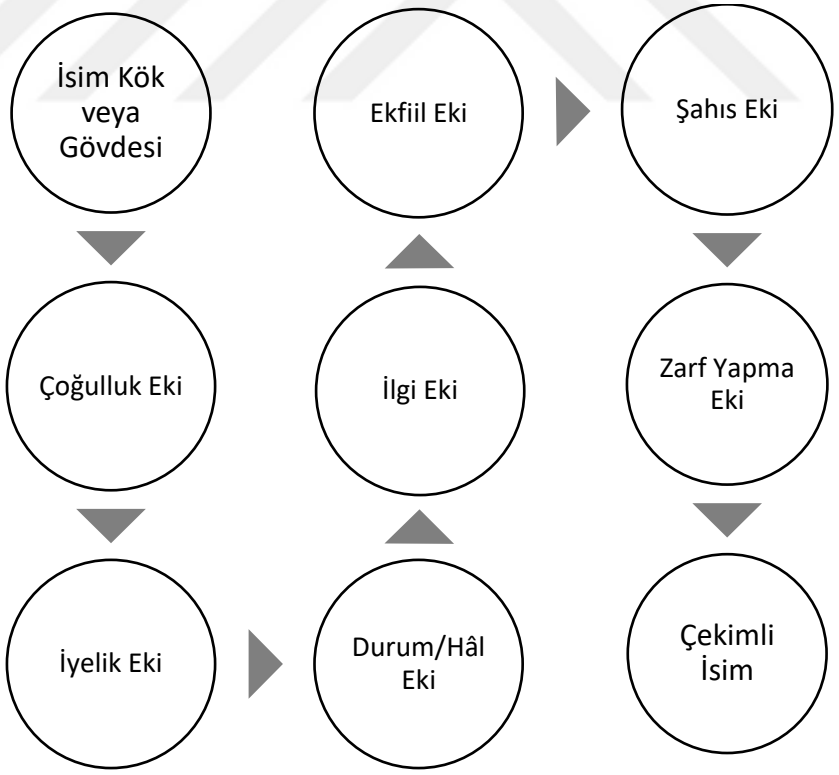
“hastasınız” → “hasta” (İsim Gövdesi) + “-sınız” (Şahıs Eki)

“çalışkandın” → “çalışkan” (İsim Gövdesi) + “-dı” (Ekfiil eki) + “-n” (Şahıs Eki)

Zarf yapma ekleri, isimden zarf türetir. Anlam olarak tarz, biçim, benzerlik bildirir. Örneğin:

“çocukça” → “çocuk” (İsim Kökü) + “-ça” (Zarf Yapma Eki)

Türkçede isim kök veya gövdelerine gelen çekim ekleri belirli bir düzene göre eklenir [51]. Bu düzen Şekil 3.2’deki gibidir:



Şekil 3.2. Türkçede İsme Gelen Çekim Ekleri Düzeni

Fiillere Gelen Çekim Ekleri: Türkçede fiillere gelen çekim ekleri, fiilin cümledeki zaman, kişi gibi işlevlerini belirleyen ve anlamını değiştirmeyen eklerdir. Fiillere gelen çekim ekleri olumsuzluk, bileşik, dilek, haber kipi ve şahıs olarak beş gruba ayrılır [51].

Olumsuzluk eki, fiilin olumsuz biçimini oluşturan ektir. Temel olarak *"-me/-ma"* şeklindedir ve fiil kökü veya gövdesinden hemen sonra gelir. Örneğin:

"gelme" (Olumsuz Fiil) → "gel" (Olumlu Fiil Kökü) + "-me" (Olumsuzluk Eki)

Bileşik eki, Türkçede fiil köklerine gelerek, fiilin anlamını değiştirmeden, hareketin süresi, niteliği ya da derecesi hakkında bilgi verir. Tezlik, yeterlik, sürerlik, yaklaşma ve diğerleri olarak beş başlıkta toplanır. Örneğin:

"yetişiver" (Bileşik Fiil) → "yetiş" (Fiil Gövdesi) + "-iver" (Tezlik Bileşik Eki)

"görebil" (Bileşik Fiil) → "gör" (Fiil Kökü) + "-ebil" (Yeterlik Bileşik Eki)

"bakadur" (Bileşik Fiil) → "bak" (Fiil Kökü) + "-adur" (Sürerlik Bileşik Eki)

"düşeyaz" (Bileşik Fiil) → "düş" (Fiil Kökü) + "-eyaz" (Yaklaşma Bileşik Eki)

"koşarak" (Bileşik Fiil) → "koş" (Fiil Kökü) + "-arak" (Diğer Bileşik Eki)

Dilek kipleri, fiilin zaman belirtmeyen, daha çok konuşanın niyetini, isteğini, şartını ya da gerekliliği bildiren biçimleridir. Bu ekler fiilin anlamını, gerçekleşip gerçekleşmediği yönünde değil, gerçekleşme isteği ya da zorunluluğu üzerinden şekillendirir. Gereklilik, şart, istek ve emir olarak dört ana başlıkta toplanır. Örneğin:

"gitmeli" (Dilek Fiili) → "git" (Fiil Kökü) + "-meli" (Gereklilik Kipi)

"boyasa" (Dilek Fiili) → "boya" (Fiil Kökü) + "-sa" (Şart Kipi)

"süreyim" (Dilek Fiili) → "sür" (Fiil Kökü) + "-e" (İstek Kipi)

"yap" (Dilek Fiili) → "yap" (Fiil Kökü) + "-" (Emir Kipi)

Haber kipleri, fiilin gerçekleşme zamanını bildirir. Türkçede geniş, gelecek, şimdiki, duyulmuş geçmiş ve görülen geçmiş olarak beş temel haber kipi vardır ve her biri farklı zaman anlamı taşır. Örneğin:

“sever” (Haber Fiili) → “sev” (Fiil Kökü) + “-er” (Geniş Zaman Kipi)

“okuyor” (Haber Fiili) → “oku” (Fiil Kökü) + “-yor” (Şimdiki Zaman Kipi)

“uçacak” (Haber Fiili) → “uç” (Fiil Kökü) + “-acak” (Gelecek Zaman Kipi)

“tuttu” (Haber Fiili) → “tut” (Fiil Kökü) + “-tu” (Görülen Geçmiş Zaman Kipi)

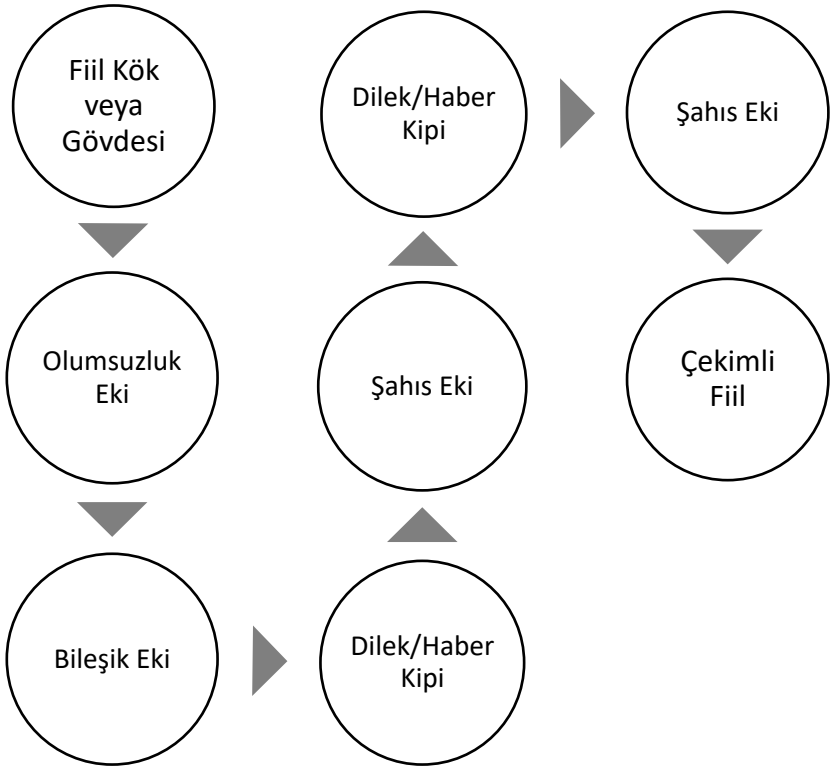
“kaçmış” (Haber Fiili) → “kaç” (Fiil Kökü) + “-mış” (Duyulan Geçmiş Zaman Kipi)

Şahıs ekleri, fiilin kim tarafından yapıldığını gösterir. Her dilek ve haber kipi, ilgili kişi (ben, sen, o, biz, siz, onlar) ile çekimlenebilir. Şahıs ekleri bazen doğrudan fiile eklenir, bazen de dilek ve haber kiplerinden sonra gelir. Örneğin:

“kaçtın” (Fiil) → “kaç” (Fiil Kökü) + “-tı” (Haber Kipi) + “-n” (Şahıs Kipi)

“temiziz” (Fiil) → “temiz” (Fiil Kökü) + “-iz” (Şahıs Kipi)

Türkçede bir fiil birden fazla dilek ya da haber kipi alabilir. Fiil kök veya gövdelerine gelen çekim ekleri belirli bir düzene göre eklenir [51]. Bu düzen Şekil 3.3’teki gibidir:



Şekil 3.3. Türkçede Fiile Gelen Çekim Ekleri Düzeni

Fiilimsileri Oluşturan Çekim Ekleri: Fiil kök veya gövdelerine gelen eklerle yeni bir isim soylu kelime oluşur. Bu yapılar Fiilimsi olarak adlandırılır. Fiil soylu kelimelerden oluştukları için olumsuzluk ve bileşik eklerini alabilirler. Bu eklerin yanı sıra isim-fiil, sıfat-fiil ve zarf-fiil ekleri alabilirler. Bileşik ve Olumsuzluk ekleri fiillere gelen çekim eklerindekiler ile aynıdır [51].

İsim-Fiil ekleri geldikleri fiil kök veya gövdesini isme çeviren eklerdir. Örneğin:

“yüzmek güzeldir.” → “yüz” (Fiil Kökü) + “-mek” (İsim-Fiil Eki)

“konuşuş biçimi farklı.” → “konuş” (Fiil Gövdesi) + “-uş” (İsim-Fiil Eki)

Sıfat-Fiil ekleri geldikleri fiil kök veya gövdesini sıfata çeviren eklerdir. Örneğin:

“gelen yolcu burada.” → “gel” (Fiil Kökü) + “-en” (Sıfat-Fiil Eki)

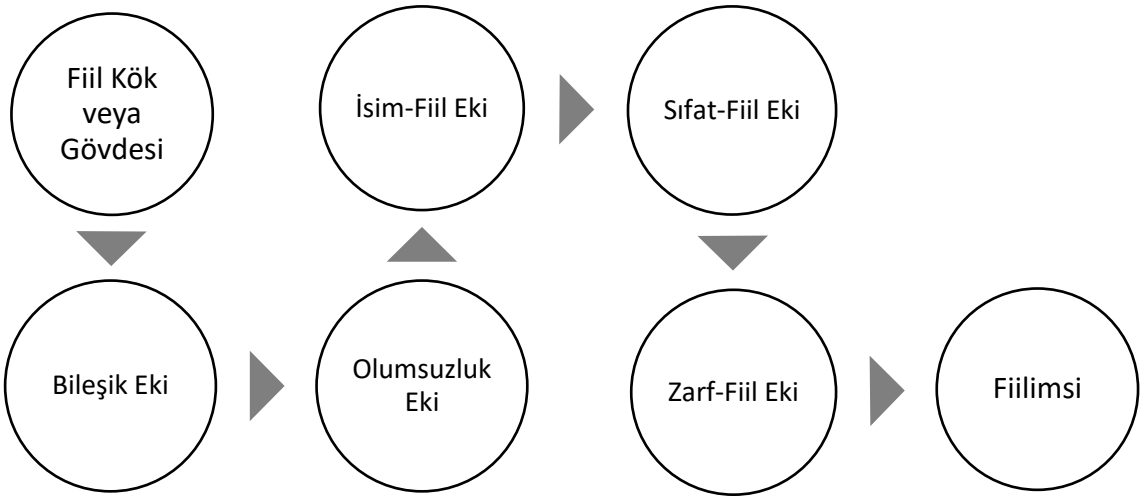
“batmış gemiler var.” → “bat” (Fiil Kökü) + “-mış” (Sıfat-Fiil Eki)

Zarf-Fiil ekleri geldikleri fiil kök veya gövdesini zarfa çeviren eklerdir. Örneğin:

“yürüyerek gitti.” → “yürü” (Fiil Kökü) + “-yerek” (Zarf-Fiil Eki)

“o gidince yaparız.” → “git” (Fiil Kökü) + “-ince” (Zarf-Fiil Eki)

Fiilimsileri oluşturan çekim ekleri belirli bir düzene göre eklenir [51]. Bu düzen Şekil 3.4’teki gibidir:



Şekil 3.4. Türkçede Fiilimsileri Oluşturan Eklerin Düzeni

3.1.3. Türkçede Ses Olayları

Türkçede ses olayları, kelimelerin yapısında yer alan seslerin ek alırken ya da söyleniş sırasında değişime uğramasıyla ortaya çıkan fonolojik (sesbilimsel) değişikliklerdir. Bu olaylar, Türkçenin ses kurallarını ve sözcüklerin daha kolay, akıcı, uyumlu telaffuz edilmesini sağlar [52]. Yazılış ile söyleniş farkları, eklemeye bağlı ses değişiklikleri, harf düşmeleri gibi biçimlerde karşımıza çıkar. Türkçede görülen temel ses olayları aşağıda tanımlanmıştır:

Ünlü Düşmesi: İki heceli bazı kelimeler ek aldığında, ikinci hecedeki dar ünlü (ı, i, u, ü) düşer [52]. Örneğin:

“ağzı” → “ağız” + “-ı”

“burnu” → “burun” + “-u”

Ünlü Daralması: Sonu *“-a/-e”* ile biten bazı fiillere *“-yor”* eki geldiğinde, son hecedeki geniş ünlü daralır (a → ı, e → i) [52]. Örneğin:

“başlıyor” → “başla” + “-yor”

“suluyor” → “sula” + “-yor”

Ünlü Türemesi: Ünsüz ile biten bazı kelimelere *“-cık/-cik”* ekleri geldiğinde araya ünlü (a, e, ı, i, o, ö, u, ü) girer [52]. Örneğin:

“daracık” → “dar” + “-cık”

“gencecik” → “genç” + “-cik”

Ünsüz Yumuşaması: Sert ünsüzle (p, ç, t, k) biten bir kelimeye ünlü ile başlayan bir ek geldiğinde, bu ünsüzler yumuşar (b, c, d, ğ) [52]. Örneğin:

“kitabı” → “kitap” + “-ı”

“çiçeği” → “çiçek” + “-i”

Ünsüz Sertleşmesi: Sert ünsüzle (p, ç, t, k) biten bir kelimeye yumuşak ünsüzle (b, c, d, ğ) başlayan bir ek geldiğinde, ekin başındaki ünsüz sertleşir [52]. Örneğin:

“dolapta” → “dolap” + “-da”

“bitkin” → “bit” + “-gin”

Ünsüz Türemesi: Ünsüzle biten bir kelimeye ünlü ile başlayan bir ek geldiğinde, kelimenin sonundaki ünsüz tekrar eder [52]. Örneğin:

“hakkı” → “hak” + “-ı”

“hissi” → “his” + “-i”

Ünsüz Düşmesi: Ünsüzle biten bir kelimeye ünsüz ile başlayan bir ek geldiğinde, bazı kelimelerin sonundaki ünsüz düşer [52]. Örneğin:

“küçücük” → “küçük” + “-cük”

“adaş” → “ad” + “-daş”

Kaynaştırma Ünsüzü: Ünlüyle biten kelimelere ünlüyle başlayan ek gelirse iki ünlü yan yana gelmemesi için araya *“y”, “ş”, “s”, “n”* gibi kaynaştırma harfleri girer [52]. Örneğin:

“anneye” → “anne” + “-e”

“bunda” → “bu” + “-da”

3.1.4. Veri Materyalleri

Türkçede yapılan DDİ uygulamaları olan gövdeleme işlemlerinde ve ASÇ’de çeşitli materyallere ihtiyaç duyulmaktadır. Performansa bağlı olarak değişiklik gösterse de temel bazı materyaller her gövdeleme çalışmasında kullanılmaktadır. Bu çalışmada da yapım ekleri listesi hariç diğer tüm materyaller kullanılmıştır.

- Durdurma Sözcükleri
- Yapım veya Çekim Ekleri Listesi
- Kök veya Gövde Sözlüğü
- Test için Gerekli Veri Seti

3.1.4.1. Durdurma Sözcüklerinin Hazırlanması

Metinde geçen ve metin hakkında bilgi vermeyen kelimeler Durdurma Sözcükleri olarak tanımlanır. 374 tane durdurma sözcüğü, Türkçe karakter desteği sağlayan UTF-8 formatına uygun bir *"csv"* dosyasında toplanarak ön işleme adımında kullanılmak üzere hazırlanmıştır. Çoğunluğu bağlaç, edat, zamir ve belirteçlerden oluşan durdurma sözcüklerine Tablo 3.6'daki örnekler verilebilir:

Tablo 3.6. Örnek Durdurma Sözcükleri

Sözcükler	Türü
ve, veya, ama	Bağlaç
adeta, biraz, daha	Belirteç
gibi, için, bile	Edat
bana, siz, onu	Zamir

3.1.4.2. Eklerin Hazırlanması

ASÇ'de metin hakkında bilgi veren ve önem teşkil eden kelimeler seçilirken yapılacak uygulamanın amacına göre aynı köke veya gövdeye sahip kelimeler tercih edilir. Bu çalışma içinde kökler yerine gövdelerin önem teşkil ettiği kararlaştırılmıştır. Çünkü yapım ekleri geldikleri isim veya fiil kökünün anlamını ve yapısını değiştirdiklerinden her bir gövde birbirinden ayrı değerlendirmek daha doğru olacaktır. Bunun için kelimeye yeni bir anlam katmayan çekim eklerinin kelimelerden silinmesi yani gövdeleme işleminin yapılması gerekmektedir.

Gövdeleme adımında kullanılmak üzere çekim eklerinin isim soylu, fiil soylu ve fiilimsiler için bir listesi oluşturulmuştur. Böylece gövdeleme işlemi yapılacak olan kelimenin türünün de (isim soylu, fiil soylu, fiilimsi) belirlenmesi sağlanmıştır. İsim soylu kelimelere gelen 217 tane, fiil soylu kelimelere gelen 236 tane ve fiilimsilere gelen 118 tane çekim eki toplanıp üç ayrı UTF-8 formatına uygun *"csv"* dosyası hazırlanmıştır.

İsim soylu kelimelere gelen çekim ekleri aşağıdaki gibi sınıflandırılmıştır:

- Çokluk
- İyelik
- Hâl
- İlgi
- Ekfiil
- Şahıs
- Zarf Yapma

Fiil soylu kelimelere gelen çekim ekleri aşağıdaki gibi sınıflandırılmıştır:

- Olumsuzluk
- Bileşik
- Dilek
- Haber
- Şahıs

Fiilimsileri oluşturan çekim ekleri aşağıdaki gibi sınıflandırılmıştır:

- İsim-Fiil
- Sıfat-Fiil
- Zarf-Fiil

Üç türe ait çekim eklerine Tablo 3.7'deki örnekler verilebilir:

Tablo 3.7. Örnek Çekim Ekleri

Ekler	Adı	Türü
-ler, -lar, -zler, -zlar	Çokluk	İsim Çekim Eki
-m, -n, -miz, -niz	İyelik	İsim Çekim Eki
-in, -si, -da, -dan	Hâl	İsim Çekim Eki
-ki, -kü	İlgi	İsim Çekim Eki
-du, -müş, -se, -tir	Ekfiil	İsim Çekim Eki
-um, -sin, -ız, -ünüz	Şahıs	İsim Çekim Eki
-ken, -la, -ce, -rak	Zarf Yapma	İsim Çekim Eki
-maz, -mu, -yemi	Olumsuzluk	Fiil Çekim Eki
-iver, -adur, -agör	Bileşik	Fiil Çekim Eki
-el, -ysa, -sın	Dilek	Fiil Çekim Eki
-mekte, -üyor, -dü	Haber	Fiil Çekim Eki
-um, -zsın, -yiniz	Şahıs	Fiil Çekim Eki
-ma, -iş, -mek	İsim-Fiil	Fiilimsi Çekim Eki
-yen, -tık, -diği	Sıfat-Fiil	Fiilimsi Çekim Eki
-yalı, -ip, esiye	Zarf-Fiil	Fiilimsi Çekim Eki

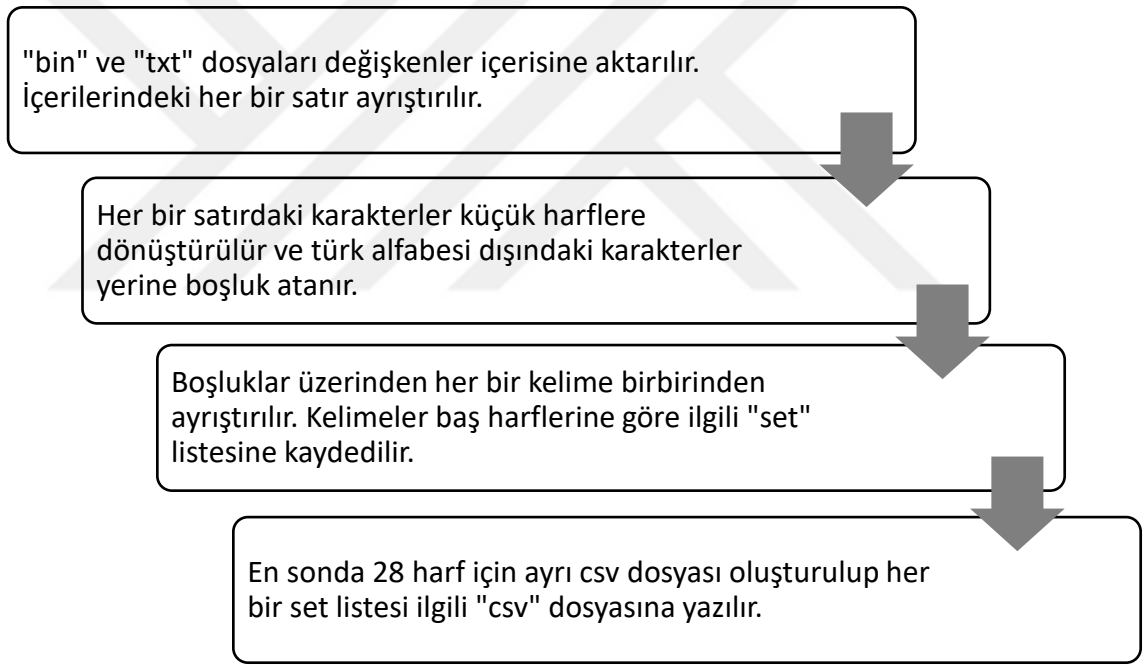
3.1.4.3. Sözlüğün Hazırlanması

Daha önceki çalışmalar göz önüne alındığında sözlük kullanmadan gövdeleme ve köke ulaşma yöntemleri ne kadar performanslı olsa da doğruluk bu çalışma için ön planda tutularak sözlük kullanımı tercih edilmiştir.

Bu çalışmada “Zemberek” [14] çalışmasının sözlüğü ve Türk Dil Kurumu (TDK) üzerinden alınarak oluşturulmuş sözlük [53] birleştirilerek tek bir sözlük haline getirilmiştir. Zemberek çalışmasının sözlüğü kelime köklerinden oluşmaktadır. TDK sözlüğü ise kelime köklerini “-ma”, “-ış”, “-mak” gibi yapım ekleriyle birlikte

bulundurmaktadır. Yani kelime kökleri yalın halde değildir. TDK sözlüğünde ayrıca kelime gövdeleri de bulunmaktadır. Bu yüzden Zemberek çalışmasının sözlüğünde bulunan yalın haldeki kelime kökleriyle TDK sözlüğünde bulunan kelime gövdeleri birleştirilerek yeni bir sözlük oluşturulmuştur. Zemberek çalışmasının kullandığı sözlük *"bin"* dosyası olduğundan içerisindeki veriler bazı işlemlerden geçirilerek ayrıştırılmıştır. Daha sonra TDK sözlüğü *"txt"* dosyası olduğundan içerisindeki iki kelimeli satırlar silinmiştir. Her iki sözlük içerisindeki kelimeler birleştirilerek yöntemin hızlı çalışması için kelimelerin ilk harflerine göre ("ğ" hariç) 28 ayrı UTF-8 formatına uygun *"csv"* dosyasına kaydedilmiştir.

Şekil 3.5'te sözlük üzerinde sırasıyla yapılan işlemler gösterilmiştir:



Şekil 3.5. Sözlük Düzenleme Adımları

3.1.4.4. Öykülerin Hazırlanması

Geliştirilen yöntemin testi için Millî Eğitim Bakanlığının ilkokul ve ortaokul için çıkarmış olduğu kısa öykü kitaplarından alınan öyküler ile yazarı bilinmeyen anonim kısa öykülerin bir araya getirilmesiyle 1000 adet öykü toplanmıştır. Bu çalışmada test için kullanılmak üzere 100 tanesi seçilerek UTF-8 formatına uygun *"csv"* dosyası hazırlanmıştır.

3.2. Yöntem

Türkçe kısa öykülerden ASÇ 4 aşamada geliştirilmiştir. Aşağıda bu aşamalar ve her aşamadaki yapılan işlemler sırasıyla gösterilmiştir:



Şekil 3.6. Yöntem Geliştirme Aşamaları

3.2.1. Ön İşleme

Bu aşamada öykülerin gövdeleme aşamasına girdi olması için bazı işlemlerden geçmesi sağlanmıştır. İlk olarak öykülerin içeriği Türkçe alfabesi dışındaki anlamsız sembollerden ve noktalama işaretlerinden temizlenir. Daha sonra ise kalan metin küçük harflere dönüştürülerek kelimeler birbirinden ayrılır. Son olarak ise metinde geçen fakat metin hakkında bilgi vermeyen durdurma sözcükleri temizlenir.

Şekil 3.7'de sırasıyla yapılan bu işlemlerin *"python"* dilinde kısa kodları gösterilmiştir:

```
x = talesData['text'][taleIndex]
y = re.sub("[^(şŞıİüÜöÖçÇğĞ)(a-zA-Z)]",' ',x,)
tale = y.lower()
words = tale.split()
words = [i for i in words if not i in stopWords]
```

Şekil 3.7. Ön İşleme Adımı Kodları

3.2.2. Gövdeleme

Yapılan önceki çalışmalarda da görüldüğü gibi Türkçedeki sondan eklemeli kelime yapısının kural tabanlı bir sistem ile modellenebileceği kanıtlanmıştır [3,6,10]. Bu aşamada da kural tabanlı yaklaşımın SDM kullanılarak modellenmesine karar verilmiştir.

SDM, girdi sembollerine bağlı olarak sonlu sayıda durum arasında geçiş yapan matematiksel bir modeldir. Otomat kuramının temel yapı taşlarından biridir ve hem teorik bilgisayar bilimlerinde hem de DDİ uygulamalarında yaygın olarak kullanılır. SDM, bir başlangıç durumu ile başlar, girdilere göre durumlar arasında geçiş yapar. Bazı durumlar kabul durumu ya da son durum olarak tanımlanır.

Şekil 3.8'deki örnek SDM yapısında da görüldüğü üzere birden fazla durum arasında uygun koşulların sağlanmasıyla geçiş yapılabilmektedir. Bir duruma bağlı koşullar sağlandığı sürece başka bir duruma geçiş yapmadan aynı durumda kalınabilir.



Şekil 3.8. Örnek SDM Yapısı

Gövdeleme aşamasında, ön işleme aşamasında temizlenen ve birbirinden ayrılan kelimelerin, SDM'ler kullanarak gövdeleri bulunmuştur. Bu aşamada sırasıyla aşağıdaki işlemler yapılmıştır:

- Çekim eklerinin olduğu tablolardan tüm ek türleri çekilerek dizilere kaydedilmiştir. Ayrıca SDM'ler sonucunda belirlenen çekim eklerinin kaydedilmesi için isimlerde *"nounResults",* fiillerde *"verbResults"* ve fiilimsilerde *"gerundialResults"* adlarında diziler oluşturulmuştur.

- Kelime üzerinde daha önce kaydedilmiş olan dizilerdeki çekim eklerinin var olup olmadığını tespit edecek *"findSuffix"* adında fonksiyon oluşturulmuştur. Bu fonksiyon bir döngü içerisinde, kelime sonundan bir harf alarak eklerin yer aldığı dizide arama işlemi yapar ve sonucunda en uzun eşleşen eki seçer.

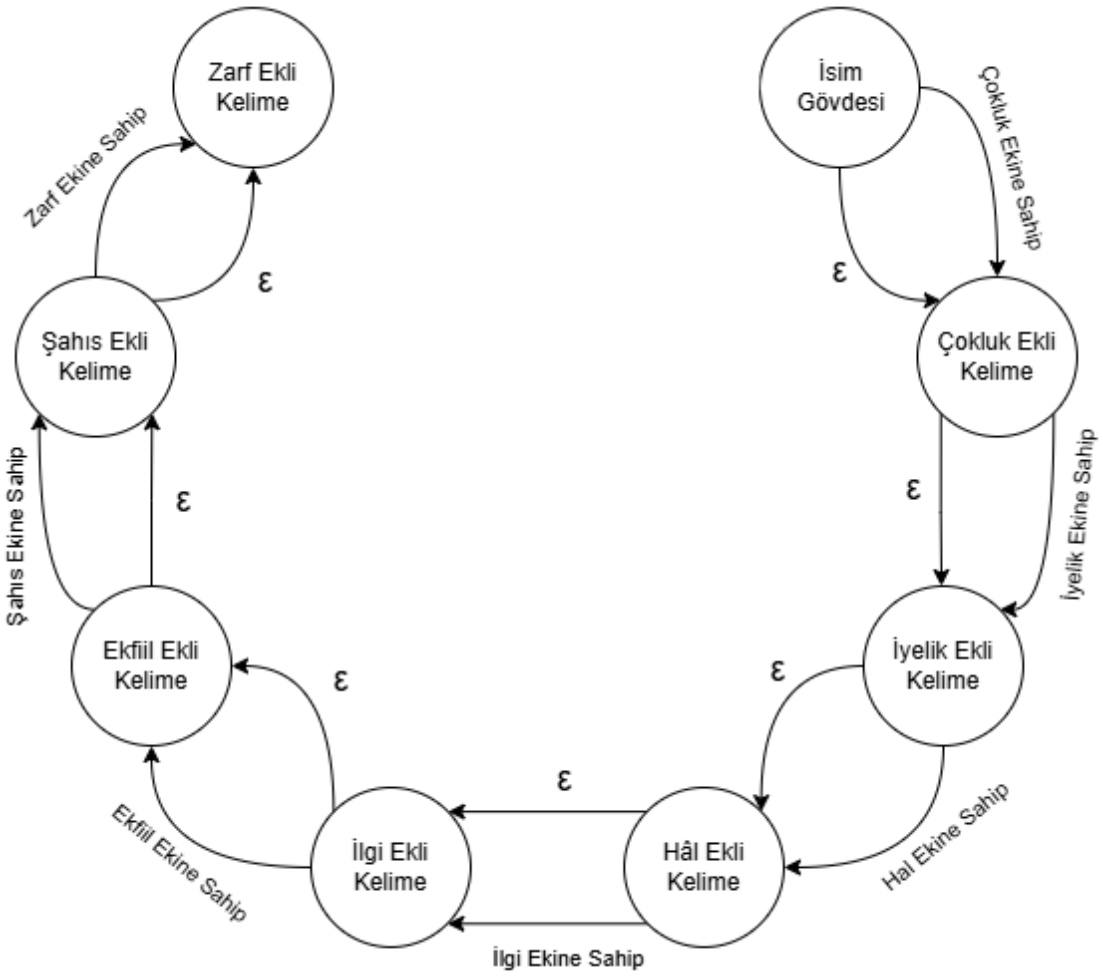
- Çekim ekleri daha önce isim, fiil ve fiilimsilere gelenler olarak üç türe ayrılmıştı. Bu üç tür için ayrı ayrı SDM tasarlanmış ve içerisinde yer alan tüm durumlar için koşullar oluşturulmuştur.

- SDM girdileri öncesinde kelime sözlükte aratılıp sözlükte yer alıyorsa frekans analizi için diziye kaydedilerek gövdeleme işlemi sonlandırılır. Böylelikle daha hızlı gövde bulunmuş ve olası yanlış gövdelerden kaçınılmıştır. Sözlükte yer almıyor ise, SDM'lere girdi olarak verilerek SDM sonucunda bulunan gövdenin kelime türü seçilmiştir.

- Kelime türü seçilen gövde doğruluk için sözlükte aratılmıştır. Sözlükte yer alıyor ise frekans analizi için diziye kaydedilmiştir. Sözlükte yer almıyor ise çeşitli kontrollere tabi tutulup sonucunda oluşan gövde tekrardan sözlükte aratılmıştır. Sözlükte yer alıyor ise bulunan gövde, yer almıyorsa ise ona en yakın olan kelime sözlükten seçilerek frekans analizi için diziye kaydedilmiştir.

3.2.2.1. İsimler için SDM Tasarımı

İsim soylu kelimelere gelen çekim eklerinin düzeni daha önce Şekil 3.2'de gösterilmişti. Bu düzen, eklerin kelimeden silinebilmesi için aktarılarak Şekil 3.9'daki gibi bir SDM oluşturulmuştur:



Şekil 3.9. İsimler için Oluşturulan SDM

Türkçede isim kökü veya gövdesine gelen çekim ekleri her zaman Şekil 3.9'deki düzende tüm eklere sahip olmayabilir. Örneğin çokluk ekine sahip bir isim, iyelik eki almadan hal eki alabilir. Bu yüzden SDM tasarlanırken bir durum içerisinde o duruma ait bir ek bulunsa da bir sonraki duruma geçiş yapılabilir bulunmasa da bir sonraki duruma geçiş yapılabilmektedir.

3.2.2.2. İsimler için SDM Koşulları

İsim soylu kelimeler için SDM hazırlandıktan sonra her bir durum içerisinde koşullar oluşturulmuştur.

Zarf Ekli Kelime Durumu:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.
    a. Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.
    b. Koşul sağlanıyor ise 2.adıma geçilir.
2. Zarf yapma eklerinin olduğu diziyi kullanarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.
    a. Bir ek bulunmaz ise kelime bir sonraki duruma gönderilir.
    b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.
3. *word* en az üç harflimi kontrol edilir.
    a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?
        i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.
        ii. Bu koşullar sağlanıyor ise *word*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.
            1. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.

2. Koşullardan biri sağlanıyor ise bulunan ekin kendisi, ekin türünü belirten *adverb* etiketi ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.

b. Koşul sağlanıyor ise bulunan ekin kendisi, ekin türünü belirten *adverb* etiketi ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.

1.adımda yer alan kelime uzunluğunun ikiden büyük olma koşulu Türkçedeki dil yapısından kaynaklanır. Türkçede anlamlı bir kelime en az iki harfli olmalıdır. 3.adımda *word* en az üç harfli ise bir problem teşkil etmezken bu koşul sağlanmadığında *word*'un iki harfli ya da daha az uzunlukta olduğu tespit edilir. 3.a adımında *word*'un iki harfli olması ve bulunan ekin en az iki harfli olması istenmiştir. Tek harfli ekler bazen ek olmayıp kelimenin gövdesine ait olan harfler olabilmektedir. Bulunan ekin en az iki harfli olması ile ek olmayan harfler kelimeden silinmeyerek daha doğru sonuç elde edilmiştir. Türkçede anlamlı iki harfli kelimeler sesli harfler ile başlar (ör. "iş", "ok", "aç"). İstisnai olarak sadece "de" ve "ye" kelimeleri sessiz harf ile başlar (ör. "demek", "dedim", "yemek", "yedim"). Bu iki durum, 3.a.ii adımında kontrol edilir. Bulunan ek *nounResults* dizisine kaydedilirken *false* etiketinin kullanılmasının sebebi sonraki durumlarda daha uzun eşleşen bir ekin bulunabileceği ihtimalinden dolayı ekin henüz kelimeden silinmemesi gerektiğini temsil etmesi için eklenir.

Şahıs, Ekfiil, İlgi, Hal ve İyelik Ekli Kelime Durumları:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.

a. Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.

b. Koşul sağlanıyor ise 2.adıma geçilir.

2. Şahıs, ilgi, hal ve iyelik eklerinin olduğu dizilerden biri kullanılarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.

a. Bir ek bulunmaz ise 5.adıma geçilir.

b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.

3. *word* en az üç harflimi kontrol edilir.

a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?

i. Koşullar sağlanmıyor ise 5.adıma geçilir.

ii. Koşullar sağlanıyor ise *word*'un son harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

1. Koşullar sağlanmıyor ise 5.adıma geçilir.

2. Koşullardan biri sağlanıyor ise 4.adıma geçilir.

b. Koşul sağlanıyor ise 4.adıma geçilir.

4. *nounResults* dizisi boş mu kontrol edilir.

a. Boş ise bulunan ekin kendisi, ekin türünü belirten *adperson, additional, relative, case, possessive* etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.

b. Boş değil ise *nounResults* dizisine en son kaydedilen ek ile yeni bulunan ek harf sayısına göre karşılaştırılır.

i. *nounResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha uzun veya eşit ise 5.b adımına geçilir.

ii. *nounResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha kısa ise en son kaydedilen ek diziden silinir. Yeni bulunan ekin kendisi, ekin türünü belirten *adperson, additional, relative, case,*

possessive etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.

5. *nounResults* dizisinde daha önce adperson etiketine sahip bir ek bulunmuş ise bu ek diziden silinir (Sadece Ekfiil Ekli Durum için geçerlidir). *nounResults* dizisi boş mu kontrol edilir.

a. Boş ise kelime bir sonraki duruma gönderilir.

b. Boş değil ise *nounResults* dizisine en son kaydedilen ek, kelimeden silinerek yeni oluşan kelime *word* değişkenine kaydedilir. Şahıs, ilgi, hal ve iyelik eklerinin olduğu dizilerden biri kullanılarak *findSuffix* fonksiyonu ile *word*'un sonundan başlayıp ek arama işlemi yapılır.

i. Bir ek bulunmaz ise kelime bir sonraki duruma gönderilir.

ii. Bir ek bulunur ise *word*'dan silinir ve ekin silinmiş olduğu kelime *nextWord* adında bir değişkene kaydedilerek c adımına geçilir.

c. *nextWord* en az üç harflimi kontrol edilir.

i. Koşul sağlanmıyor ise d adımına geçilir.

ii. Koşul sağlanıyor ise *nounResults* dizisine en son kaydedilen ekin *false* etiketi *true* olarak değiştirilir. Yeni bulunan ekin kendisi, ekin türünü belirten *adperson, additional, relative, case, possessive* etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.

d. Bulunan yeni ek en az iki harfli mi? ve *nextWord* tam iki harfli mi? kontrol edilir.

i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.

ii. Koşullar sağlanıyor ise e adımına geçilir.

e. *nextWord*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.

ii. Koşullardan biri sağlanıyor ise *nounResults* dizisine en son kaydedilen ekin *false* etiketi *true* olarak değiştirilir. Yeni bulunan ekin kendisi, ekin türünü belirten *adperson, additional, relative, case, possessive* etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.

1., 2. ve 3.adım zarf ekli kelime durumunda yer alanlarla neredeyse aynıdır. Tek fark 2. ve 3.adımdaki koşullar sağlanmaz ise 5.adıma, 3.adımdaki koşullar sağlanır ise 4.adıma geçilmesidir. 4.adımda bir önceki durumda bulunan ek var mı kontrol edilir. Eğer ek yoksa yeni bulunan ek *false* etiketiyle *nounResults* dizisine kaydedilir. *false* etiketinin kullanılmasının sebebi sonraki durumlarda daha uzun eşleşen bir ekin bulunabileceğinden dolayıdır. Eğer ek varsa önceki durumda bulunan ek, yeni bulunan ek ile harf uzunluğuna göre karşılaştırılır. Yeni bulunan ek daha uzun ise önceki durumda bulunan ek *nounResults* dizisinden silinir ve yeni ek, *false* etiketiyle *nounResults* dizisine kaydedilir. Böylelikle en uzun eşleşen ek tercih edilmiş olur. Eğer önceki durumda bulunan ek yeni bulunan ekten daha uzun veya eşitse 5.b adımına geçilir. 2. ve 3.adımlardaki koşullarda sağlanmadığında 5.adıma geçilir. 2. ve 3.adımlarda önceki durumda bulunan ek kelimeden silinmeyerek kontrol edilir. İsim soylu bir kelimede ekfiil eki olmadan şahıs eki yer alamaz. Bu yüzden 5.adımda sadece ekfiil ekli kelime durumu için *nounResults* dizisinde *adperson* etiketine sahip bir ek var ise bu ek diziden silinir.

Çokluk Ekli Kelime Durumu:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.

a. Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.

b. Koşul sağlanıyor ise 2.adıma geçilir.

2. Çokluk eklerinin olduğu diziyi kullanarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.
    a. Bir ek bulunmaz ise 5.adıma geçilir.
    b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.
3. *word* en az üç harflimi kontrol edilir.
    a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?
        i. Koşullar sağlanmıyor ise 5.adıma geçilir.
        ii. Koşullar sağlanıyor ise *word*'un son harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.
            1. Koşullar sağlanmıyor ise 5.adıma geçilir.
            2. Koşullardan biri sağlanıyor ise 4.adıma geçilir.
    b. Koşul sağlanıyor ise 4.adıma geçilir
4. *nounResults* dizisi boş mu kontrol edilir.
    a. Boş ise bulunan ekin kendisi, ekin türünü belirten *plural* etiketi ve ekin kelimeden silindiğini temsil eden *true* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.
    b. Boş değil ise *nounResults* dizisine en son kaydedilen ek ile yeni bulunan ek harf sayısına göre karşılaştırılır.
        i. *nounResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha uzun veya eşit ise 5.b adımına geçilir.
        ii. *nounResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha kısa ise en son kaydedilen ek diziden silinir. Yeni bulunan

ekin kendisi, ekin türünü belirten *plural* etiketi ve ekin kelimeden silindiğini temsil eden *true* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

5. *nounResults* dizisi boş mu kontrol edilir.

a. Boş ise kelime gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

b. Boş değil ise *nounResults* dizisine en son kaydedilen ek, kelimeden silinerek yeni oluşan kelime *word* değişkenine kaydedilir. *nounResults* dizisine en son kaydedilen ekin *false* etiketi *true* olarak değiştirilir. Çokluk eklerinin olduğu diziyi kullanarak *findSuffix* fonksiyonu ile *word*'un sonundan başlayıp ek arama işlemi yapılır.

i. Bir ek bulunmaz ise *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

ii. Bir ek bulunur ise *word*'dan silinir ve ekin silinmiş olduğu kelime *nextWord* adında bir değişkene kaydedilerek c adımına geçilir.

c. *nextWord* en az üç harflimi kontrol edilir.

i. Koşul sağlanmıyor ise d adımına geçilir.

ii. Koşul sağlanıyor ise yeni bulunan ekin kendisi, ekin türünü belirten *plural* etiketi ve ekin kelimeden silindiğini temsil eden *true* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından *nextWord* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

d. Bulunan yeni ek en az iki harfli mi? ve *nextWord* tam iki harfli mi? kontrol edilir.

i. Koşullar sağlanmıyor ise *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

ii. Koşullar sağlanıyor ise e adımına geçilir.

e. *nextWord*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

i. Koşullar sağlanmıyor ise *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

ii. Koşullardan biri sağlanıyor ise yeni bulunan ekin kendisi, ekin türünü belirten *plural* etiketi ve ekin kelimeden silinmesi gerektiğini temsil eden *true* etiketi bir araya getirilerek *nounResults* dizisine kaydedilir. Ardından *nextWord* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

1., 2. ve 3.adım diğer durumda yer alanlarla tamamen aynıdır. Diğer durumlarda 4.a adımında *nounResults* dizisine *false* etiketi ile kaydedilip bir sonraki duruma geçiş yapılırken burada *nounResults* dizisi boş ise *true* etiketi ile kaydedilir, *word* gövde olarak etiketlenir. *true* etiketinin kullanılması ve word'un gövde olarak etiketlenmesi, çokluk ekli kelime durumunun SDM'nin son durumu olduğundan daha sonra bir ek bulunmayacağı içindir. 4.b.ii, 5.b, 5.c.ii ve 5.e.ii adımları da aynı durumdadır. 5.b adımında *nounResults* dizisine en son kaydedilen etiketin *true*'ya çevrilmesi gövdeleme işlemi sonlandırılmadan en son bulunan ekin kelimeden silinmesinin sağlanmasıdır.

3.2.2.3. İsimler için Gövde Kontrolü

İsim SDM'sinden çıkan gövde direkt sonuç olarak kabul edilmeden önce gövde olarak etiketlenen kelime *nounResult* adında bir değişkenine kaydedilir ve son bir kontrol yapılır.

1. *nounResult* kaç tane "ki" değerine sahip kontrol edilir.

a. Bu değere sahip değilse 2.adıma geçilir.

b. En az bir tane "ki" değerine sahip ise *nounResults* dizisi boşaltılarak *nounResult* SDM'ye girdi olarak verilir. Yeni ekler *nounResults* dizisine kaydedilip 2.adıma geçilir.

2. *nounResults* dizisi boş mu kontrol edilir.

a. Boş ise 3.adıma geçilir.

b. Boş değil ise *nounResults*'a kaydedilmiş son ek kontrol edilir.

i. *relative* etiketine sahip değil ise 3.adıma geçilir.

ii. *relative* etiketine sahip ise *nounResult*'ın sonuna "ki" değeri eklenir ve *nounResults* dizisine kaydedilen "-ki" eki diziden silinerek 3.adıma geçilir.

3. *nounResult* kaç tane "ğ" harfine sahip kontrol edilir.

a. Bu harfe sahip değilse *nounResult* gövde olarak etiketlenir.

b. En az bir tane "ğ" harfine sahipse bu harf veya harfler *nounResult*'un 1., 2., veya 3.harfi mi kontrol edilir.

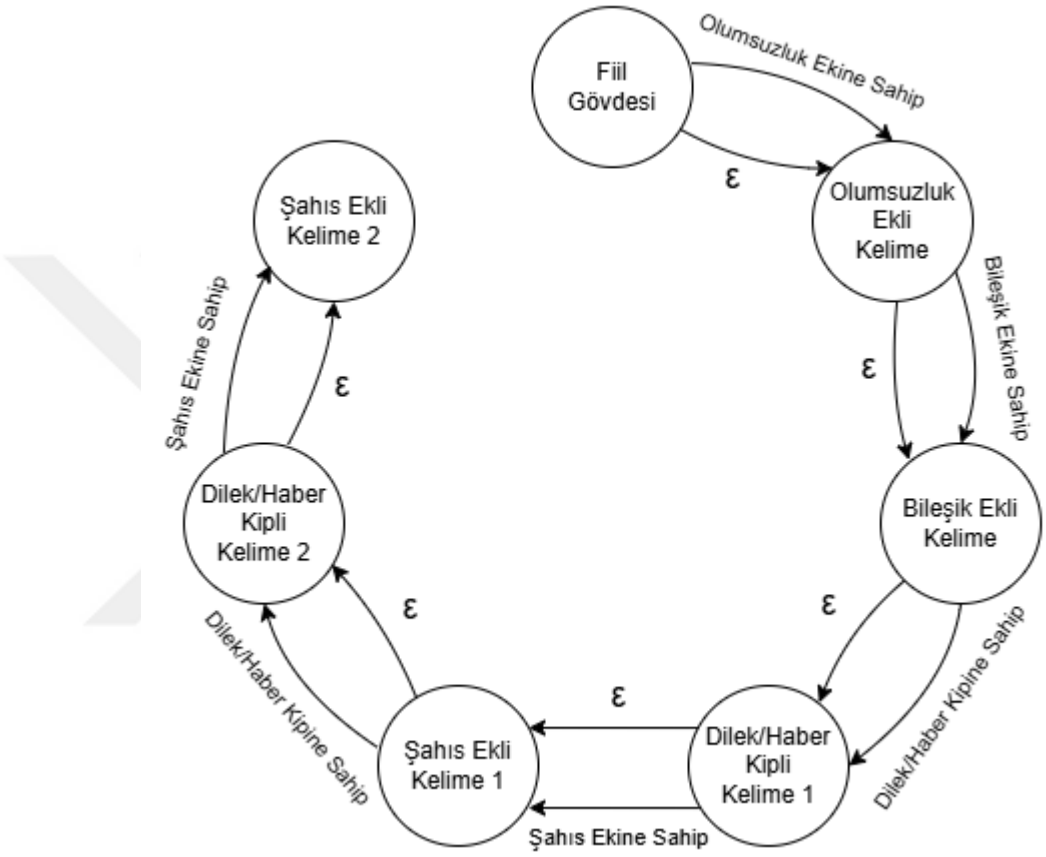
i. Koşul sağlanmıyor ise *nounResult*'ta bulunan "ğ" harfi veya harfleri "k" harfi ile değiştirilir.

ii. Koşul sağlanıyor ise *nounResult* gövde olarak etiketlenir.

Türkçedeki kelime yapısında "ki" ilgi ekini alan bir kelime yeni bir isim oluşmuş gibi tekrardan hal, iyelik ve çokluk gibi ekleri alabilmektedir. Bu yüzden SDM içerisinde "ki" yapısından öncekilere gidilememektedir. 1.adımda İsim SDM'sinden çıkan gövde "ki" değerine sahip ise tekrar SDM'ye girdi olarak verilerek ilgi ekinden önceki eklerinde bulunması sağlanmıştır. 2.adımda da *nounResults* dizisine en son kaydedilen ek *"-ki"* eki ise ondan önceki eklere sahip olunmadığı anlaşılarak bulunan ekin aslında gövdeye ait olduğuna karar verilir ve tekrardan gövdeye eklenir. Çünkü Türkçede tek başına ilgi eki *"-ki"* kullanılmaz. 3.adım ise kelimenin gövdesinde yer alan "ğ" harfi ile gövdesinde yer almayan fakat kendisine gelen ekler sebebiyle ünsüz yumuşamasına maruz kalarak "ğ" harfine dönüşen "k" harfi ayırt edilmiş olur. Türkçede anlamlı kelime en az iki veya üç harfli olmasından kaynaklı kelimenin 1., 2., veya 3.harfinden biri "ğ" ise gövdenin bir parçası olarak kabul edilir.

3.2.2.4. Fiiller için SDM Tasarımı

Fiil soylu kelimelere gelen çekim eklerinin düzeni daha önce Şekil 3.3'te gösterilmişti. Bu düzen, eklerin kelimeden silinebilmesi için aktarılarak Şekil 3.10'daki gibi bir SDM oluşturulmuştur:



Şekil 3.10. Fiiller için Oluşturulan SDM

Türkçede fiil kökü veya gövdesine gelen çekim ekleri her zaman Şekil 3.10'daki düzende tüm eklere sahip olmayabilir. Örneğin olumsuzluk ekine sahip bir fiil, bileşik eki veya şahıs eki almadan dilek/haber kipi alabilir. Bu yüzden SDM tasarlanırken bir durum içerisinde o duruma ait bir ek bulunsa da bir sonraki duruma geçiş yapılabilir bulunmasa da bir sonraki duruma geçiş yapılabilmektedir.

3.2.2.5. Fiiller için SDM Koşulları

Fiil soylu kelimeler için SDM hazırlandıktan sonra her bir durum içerisinde koşullar oluşturulmuştur.

Şahıs Ekli Kelime 2 Durumu:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.
    a. Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.
    b. Koşul sağlanıyor ise 2.adıma geçilir.
2. Şahıs eklerinin olduğu diziyi kullanarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.
    a. Bir ek bulunmaz ise kelime bir sonraki duruma gönderilir.
    b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.
3. *word* en az üç harflimi kontrol edilir.
    a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?
        i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.
        ii. Bu koşullar sağlanıyor ise *word*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.
            1. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.
            2. Koşullardan biri sağlanıyor ise bulunan ekin kendisi, *person* etiketi ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.
    b. Koşul sağlanıyor ise bulunan ekin kendisi, ekin türünü belirten *person* etiketi ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.

1.adımda yer alan kelime uzunluğunun ikiden büyük olma koşulu Türkçedeki dil yapısından kaynaklanır. Türkçede anlamlı bir kelime en az iki harfli olmalıdır. 3.adımda *word* en az üç harfli ise bir problem teşkil etmezken bu koşul sağlanmadığında *word* iki harfli ya da daha az uzunlukta olduğu tespit edilir. 3.a adımında, 1.adımdaki durumdan dolayı *word*'un iki harfli olması ve bulunan ekin en az iki harfli olması istenmiştir. Tek harfli çekim ekleri bazen ek olmayıp kelimenin gövdesine ait olan harfler olabilmektedir. Bulunan ekin en az iki harfli olması ile ek olmayan harfler kelimeden silinmeyerek daha doğru sonuç elde edilmiştir. Türkçede anlamlı iki harfli kelimeler sesli harfler ile başlar (ör. "iş", "ok", "aç"). İstisnai olarak sadece "de" ve "ye" kelimeleri sessiz harf ile başlar (ör. "demek", "dedim", "yemek", "yedim"). Bu iki durum, 3.a adımındaki koşullar sağlanır ise 3.a.ii adımında kontrol edilir. Bulunan ek sonraki durumlarda daha uzun eşleşen bir ekin bulunabileceğinden dolayı *false* etiketiyle *verbResults* dizisine kaydedilir.

Haber 2, Şahıs 1, Haber 1 ve Bileşik Ekli Kelime Durumları:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.

   a. Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.

   b. Koşul sağlanıyor ise *verbResults* dizisinde daha önce *person* etiketine sahip bir ek bulunmuş mu kontrol edilir. Bulunmuş ise kelime bir sonraki duruma gönderilir (Sadece Şahıs Ekli Kelime 1 Durumu için geçerlidir). Diğer durumlarda 2.adıma geçilir.

2. Haber, şahıs ve bileşik eklerinin olduğu dizilerden biri kullanılarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.

   a. Bir ek bulunmaz ise 5.adıma geçilir.

   b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.

3. *word* en az üç harflimi kontrol edilir.

   a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?

i. Koşullar sağlanmıyor ise 5.adıma geçilir.

ii. Koşullar sağlanıyor ise *word*'un son harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

1. Koşullar sağlanmıyor ise 5.adıma geçilir.

2. Koşullardan biri sağlanıyor ise 4.adıma geçilir.

b. Koşul sağlanıyor ise 4.adıma geçilir.

4. *verbResults* dizisi boş mu kontrol edilir.

a. Boş ise bulunan ekin kendisi, ekin türünü belirten *tense, person, compound* etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.

b. Boş değil ise *verbResults* dizisine en son kaydedilen ek ile yeni bulunan ek harf sayısına göre karşılaştırılır.

i. *verbResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha uzun veya eşit ise 5.b adımına geçilir.

ii. *verbResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha kısa ise en son kaydedilen ek diziden silinir. Yeni bulunan ekin kendisi, ekin türünü belirten *tense, person, compound* etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından kelime bir sonraki duruma gönderilir.

5. *verbResults* dizisi boş mu kontrol edilir.

a. Boş ise kelime bir sonraki duruma gönderilir.

b. Boş değil ise *verbResults* dizisine en son kaydedilen ek, kelimeden silinerek yeni oluşan kelime *word* değişkenine kaydedilir. Haber, şahıs ve

bileşik eklerinin olduğu dizilerden biri kullanılarak *findSuffix* fonksiyonu ile *word*'un sonundan başlayıp ek arama işlemi yapılır.

i. Bir ek bulunmaz ise kelime bir sonraki duruma gönderilir.

ii. Bir ek bulunur ise *word*'dan silinir ve ekin silinmiş olduğu kelime *nextWord* adında bir değişkene kaydedilerek c adımına geçilir.

c. *nextWord* en az üç harflimi kontrol edilir.

i. Koşul sağlanmıyor ise d adımına geçilir.

ii. Koşul sağlanıyor ise *verbResults* dizisine en son kaydedilen ekin *false* etiketi *true* olarak değiştirilir. Yeni bulunan ekin kendisi, ekin türünü belirten *tense, person, compound* etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.

d. Bulunan yeni ek en az iki harfli mi? ve *nextWord* tam iki harfli mi? kontrol edilir.

i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.

ii. Koşullar sağlanıyor ise e adımına geçilir.

e. *nextWord*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.

ii. Koşullardan biri sağlanıyor ise *verbResults* dizisine en son kaydedilen ekin *false* etiketi *true* olarak değiştirilir. Yeni bulunan ekin kendisi, ekin türünü belirten *tense, person, compound* etiketlerinden biri ve ekin henüz kelimeden silinmemesi gerektiğini temsil eden *false* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.

1., 2. ve 3.adım şahıs ekli kelime 2 durumunda yer alanlarla neredeyse aynıdır. Tek fark 2. ve 3.adımdaki koşullar sağlanmaz ise 5.adıma, 3.adımdaki koşullar sağlandığında ise 4.adıma geçilmesidir. Fiil soylu bir kelimede şahıs eki bir tane bulunur. Bu yüzden 1.adımda sadece şahıs ekli kelime 1 durumu için bu kontrol yapılarak *verbResults* dizisinde daha önce *person* etiketine sahip bir ek var ise işlemler sonlandırılır ve bir sonraki duruma geçilir. 4.adımda bir önceki durumda bulunan ek var mı kontrol edilir. Eğer bir ek yoksa yeni bulunan ek sonraki durumlarda daha uzun eşleşen bir ekin bulunabileceğinden dolayı *false* etiketiyle *verbResults* dizisine kaydedilir. Eğer bir ek varsa önceki durumda bulunan ek, yeni bulunan ek ile harf uzunluğuna göre karşılaştırılır. Yeni bulunan ek daha uzun ise önceki durumda bulunan ek *verbResults* dizisinden silinir ve yeni ek, *false* etiketiyle *verbResults* dizisine kaydedilir. Eğer önceki durumda bulunan ek yeni bulunan ekten daha uzun veya eşitse 5.b adımına geçilir. Böylelikle en uzun eşleşen ek tercih edilmiş olur. 2. ve 3.adımlardaki koşullarda sağlanmadığında 5.adıma geçilir. 2. ve 3.adımlarda önceki durumda bulunan ek kelimeden silinmeyerek kontrol edilir. Bu adımda ise ek kelimeden silinerek aynı koşullar tekrar kontrol edilir.

Olumsuzluk Ekli Kelime Durumu:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.

   a. Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.

   b. Koşul sağlanıyor ise 2.adıma geçilir.

2. Olumsuzluk eklerinin olduğu diziyi kullanarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.

   a. Bir ek bulunmaz ise 5.adıma geçilir.

   b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.

3. *word* en az üç harflimi kontrol edilir.

   a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?

i. Koşullar sağlanmıyor ise 5.adıma geçilir.

ii. Koşullar sağlanıyor ise *word*'un son harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

1. Koşullar sağlanmıyor ise 5.adıma geçilir.

2. Koşullardan biri sağlanıyor ise 4.adıma geçilir.

b. Koşul sağlanıyor ise 4.adıma geçilir.

4. *verbResults* dizisi boş mu kontrol edilir.

a. Boş ise bulunan ekin kendisi, ekin türünü belirten *negative* etiketi ve ekin kelimeden silindiğini temsil eden *true* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

b. Boş değil ise *verbResults* dizisine en son kaydedilen ek ile yeni bulunan ek harf sayısına göre karşılaştırılır.

i. *verbResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha uzun veya eşit ise 5.b adımına geçilir.

ii. *verbResults* dizisine en son kaydedilen ek, yeni bulunan ekten daha kısa ise en son kaydedilen ek diziden silinir. Yeni bulunan ekin kendisi, ekin türünü belirten *negative* etiketi ve ekin kelimeden silindiğini temsil eden *true* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

5. *verbResults* dizisi boş mu kontrol edilir.

a. Boş ise kelime gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

b. Boş değil ise *verbResults* dizisine en son kaydedilen ek, kelimeden silinerek yeni oluşan kelime *word* değişkenine kaydedilir. *verbResults* dizisine en son kaydedilen ekin *false* etiketi *true* olarak değiştirilir. Sahiplik eklerinin olduğu

diziyi kullanarak *findSuffix* fonksiyonu ile *word*'un sonundan başlayıp ek arama işlemi yapılır.

i. Bir ek bulunmaz ise *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

ii. Bir ek bulunur ise *word*'dan silinir ve ekin silinmiş olduğu kelime *nextWord* adında bir değişkene kaydedilerek c adımına geçilir.

c. *nextWord* en az üç harflimi kontrol edilir.

i. Koşul sağlanmıyor ise d adımına geçilir.

ii. Koşul sağlanıyor ise yeni bulunan ekin kendisi, ekin türünü belirten *negative* etiketi ve ekin kelimeden silindiğini temsil eden *true* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından *nextWord* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

d. Bulunan yeni ek en az iki harfli mi? ve *nextWord* tam iki harfli mi? kontrol edilir.

i. Koşullar sağlanmıyor ise *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

ii. Koşullar sağlanıyor ise e adımına geçilir.

e. *nextWord*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

i. Koşullar sağlanmıyor ise *word* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

ii. Koşullardan biri sağlanıyor ise yeni bulunan ekin kendisi, ekin türünü belirten *negative* etiketi ve ekin kelimeden silinmesi gerektiğini temsil eden *true* etiketi bir araya getirilerek *verbResults* dizisine kaydedilir. Ardından *nextWord* gövde olarak etiketlenir ve gövdeleme işlemi sonlandırılır.

1., 2. ve 3.adım diğer durumda yer alanlarla tamamen aynıdır. 4. ve 5.adım ise diğer durumda yer alanlarla neredeyse aynıdır. Diğer durumlardaki 4.a adımında *verbResults* dizisine kaydetme işlemi *false* etiketi ile yapılıp bir sonraki duruma geçiş yapılmaktaydı. Buradaki 4.a adımında ise *verbResults* dizisi eğer boş ise diziye kaydetme işlemi *true* etiketi ile yapılıp *word* gövde olarak etiketlenir, gövdeleme işlemi sonlandırılır. *true* etiketinin kullanılma ve *word*'un gövde olarak etiketlenmesi olumsuzluk ekli kelime durumu SDM'nin son durumu olduğundan daha sonra bir ek bulunmayacağı içindir. 4.b.ii, 5.b, 5.c.ii ve 5.e.ii adımları da aynı durumdadır. 5.b adımında gerçekleştirilen *verbResults* dizisine en son kaydedilen ekin *false* etiketinin *true*'ya çevrilmesi işlemi, bu durumun son durum olmasından dolayı gövdeleme işlemi sonlandırılmadan önce en son bulunan ekin kelimeden silinmesinin sağlanmasıdır.

3.2.2.6. Fiiller için Gövde Kontrolü

Fiil SDM'sinden çıkan gövde direkt sonuç olarak kabul edilmeden önce gövde olarak etiketlenen kelime *verbResult* adında bir değişkenine kaydedilir ve son bir kontrol yapılır.

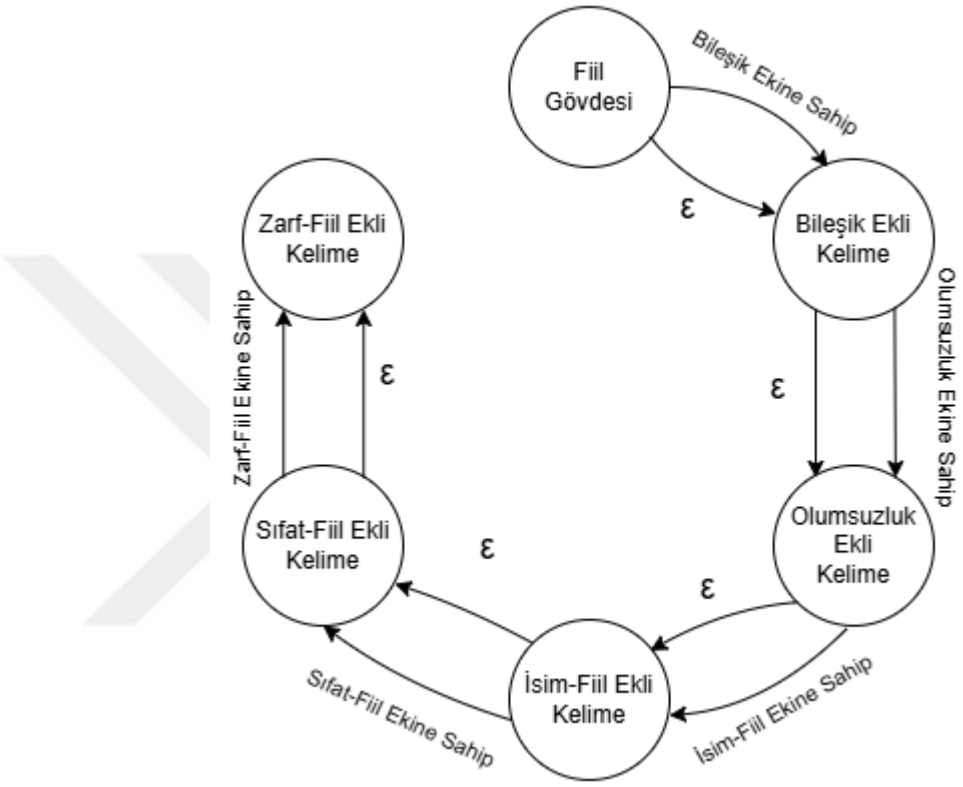
1. *verbResult* kaç tane "ğ" harfine sahip kontrol edilir.

- **a.** Bu harfe sahip değilse *verbResult* gövde olarak etiketlenir.
- **b.** En az bir tane "ğ" harfine sahipse bu harf veya harfler *verbResult*'un 1., 2., veya 3.harfi mi kontrol edilir.
  - **i.** Koşul sağlanmıyor ise *verbResult*'ta bulunan "ğ" harfi veya harfleri "k" harfi ile değiştirilir.
  - **ii.** Koşul sağlanıyor ise *verbResult* gövde olarak etiketlenir.

Bu kontrol ile kelimenin gövdesinde yer alan "ğ" harfi ile gövdesinde yer almayan fakat kendisine gelen ekler sebebiyle ünsüz yumuşamasına maruz kalarak "ğ" harfine dönüşen "k" harfi ayırt edilmiş olur. Türkçede anlamlı bir kelimenin en az iki veya üç harfli olmasından kaynaklı kelimenin 1., 2., veya 3.harfinden biri "ğ" ise gövdenin bir parçası olarak kabul edilir.

3.2.2.7. Fiilimsiler için SDM Tasarımı

Fiilimsileri oluşturan çekim eklerinin düzeni daha önce Şekil 3.4'te gösterilmişti. Bu düzen, eklerin kelimeden silinebilmesi için aktarılarak Şekil 3.11'daki gibi bir SDM oluşturulmuştur:



Şekil 3.11. Fiilimsiler için Oluşturulan SDM

Türkçede fiilimsileri oluşturan çekim ekleri her zaman Şekil 3.11'daki düzende tüm eklere sahip olmayabilir. Örneğin bileşik ekine sahip bir fiilimsi, isim-fiil eki veya sıfat-fiil eki almadan zarf-fiil eki alabilir. Bu yüzden SDM tasarlanırken bir durum içerisinde o duruma ait bir ek bulunsa da bir sonraki duruma geçiş yapılabilir bulunmasa da bir sonraki duruma geçiş yapılabilmektedir.

3.2.2.8. Fiilimsiler için SDM Koşulları

Fiilimsi olan kelimeler için SDM hazırlandıktan sonra her bir durum içerisinde koşullar oluşturulmuştur.

Zarf-Fiil, Sıfat-Fiil, İsim-Fiil ve Bileşik Ekli Kelime Durumları:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.
    a. Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.
    b. Koşul sağlanıyor ise 2.adıma geçilir.
2. Zarf-Fiil, sıfat-fiil, isim-fiil ve bileşik eklerinin olduğu dizilerinden biri kullanılarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.
    a. Bir ek bulunmaz ise kelime bir sonraki duruma gönderilir.
    b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.
3. *word* en az üç harflimi kontrol edilir.
    a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?
        i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.
        ii. Bu koşullar sağlanıyor ise *word*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya "*de-"* ile mi başlıyor? kontrol edilir.
            1. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.
            2. Koşullardan biri sağlanıyor ise bulunan ekin kendisi, ekin türünü belirten *gerundial* ve *compound* etiketlerinden biri ve ekin kelimeden silinmesi gerektiğini temsil eden *true* etiketi bir araya getirilerek *gerundialResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.
    b. Koşul sağlanıyor ise bulunan ekin kendisi, ekin türünü belirten *gerundial* ve *compound* etiketlerinden biri ve ekin kelimeden silinmesi gerektiğini temsil

eden *true* etiketi bir araya getirilerek *gerundialResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.

1.adımda yer alan kelime uzunluğunun ikiden büyük olma koşulu Türkçedeki dil yapısından kaynaklanır. Türkçede anlamlı bir kelime en az iki harfli olmalıdır. 3.adımda *word* en az üç harfli ise bir problem teşkil etmezken bu koşul sağlanmadığında *word* iki harfli ya da daha az uzunlukta olduğu tespit edilir. 3.a adımında, 1.adımdaki durumdan dolayı *word*'un iki harfli olması ve bulunan ekin en az iki harfli olması istenmiştir. Tek harfli çekim ekleri bazen ek olmayıp kelimenin gövdesine ait olan harfler olabilmektedir. Bulunan ekin en az iki harfli olması ile ek olmayan harfler kelimeden silinmeyerek daha doğru sonuç elde edilmiştir. Türkçede anlamlı iki harfli kelimeler sesli harfler ile başlar (ör. "iş", "ok", "aç"). İstisnai olarak sadece "de" ve "ye" kelimeleri sessiz harf ile başlar (ör. "demek", "dedim", "yemek", "yedim"). Bu iki durum, 3.a adımındaki koşullar sağlanır ise 3.a.ii adımında kontrol edilir. Türkçede fiilimsi olan bir kelime zarf-fiil, sıfat-fiil ve isim-fiil eklerinden aynı anda sadece birine sahip olabilir. Bu yüzden bulunan ek *gerundialResults* dizisine kaydedilirken *true* etiketi kullanılır ve *word* bir sonraki duruma gönderilir. t*rue* etiketinin kullanılmasının sebebi zarf-fiil, sıfat-fiil ve isim-fiil durumlarından birinde ek bulunduğunda diğer iki durumda bir ek bulunmayacağı için kelimeden silinmesi gerektiğini temsil etmesidir. *word*'un bir sonraki duruma gönderilmesinin sebebi de aynı şekilde diğer iki durumda bir ek bulunmayacağı içindir.

Olumsuzluk Ekli Kelime Durumu:

1. Kelime uzunluğunun iki harften büyük olup olmadığı kontrol edilir.

- **a.** Koşul sağlanmıyor ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.
- **b.** Koşul sağlanıyor ise *gerundialResults* dizisi boş mu kontrol edilir.
  - **i.** Boş ise gövdeleme işlemi sonlandırılarak kelime gövde olarak etiketlenir.
  - **ii.** Boş değil ise 2.adıma geçilir.

2. Olumsuzluk eklerinin olduğu diziyi kullanarak *findSuffix* fonksiyonu ile kelimenin sonundan başlayıp ek arama işlemi yapılır.

   a. Bir ek bulunmaz ise kelime bir sonraki duruma gönderilir.

   b. Bir ek bulunur ise kelimeden silinir ve ekin silinmiş olduğu kelime *word* adında bir değişkene kaydedilerek 3.adıma geçilir.

3. *word* en az üç harflimi kontrol edilir.

   a. Koşul sağlanmıyor ise iki koşul kontrol edilir. Bulunan ek en az iki harfli mi? ve *word* tam iki harfli mi?

      i. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.

      ii. Bu koşullar sağlanıyor ise *word*'un ilk harfi sesli harf mi? ya da başı *"ye-"* veya *"de-"* ile mi başlıyor? kontrol edilir.

         1. Koşullar sağlanmıyor ise kelime bir sonraki duruma gönderilir.

         2. Koşullardan biri sağlanıyor ise bulunan ekin kendisi, ekin türünü belirten *negative* etiketi ve ekin kelimeden silinmesi gerektiğini temsil eden *true* etiketi bir araya getirilerek *gerundialResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.

   b. Koşul sağlanıyor ise bulunan ekin kendisi, ekin türünü belirten *gerundial* etiketi ve ekin kelimeden silinmesi gerektiğini temsil eden *true* etiketi bir araya getirilerek *gerundialResults* dizisine kaydedilir. Ardından *word* bir sonraki duruma gönderilir.

Olumsuzluk ekli kelime durumu, zarf-fiil, sıfat-fiil, isim-fiil ve bileşik ekli kelime durumlarının tüm adımları ile aynıdır. Değişen tek kısım 1.adımda yer alan kelime uzunluğunun ikiden büyük olma koşulu sağlandığında *gerundialResults* dizisinin kontrol edilmesidir. Boş değil ise işlem aynı şekilde devam ederken boş olması durumunda gövdeleme işlemi sonlandırılır ve kelime gövde olarak etiketlenir. Bu kontrolün amacı

eğer bir kelime fiilimsi eklerinden birini almamışsa o kelime fiilimsi olmayacağından olumsuzluk ekinin aranmasına gerek olmamasıdır.

3.2.2.9. Fiilimsiler için Gövde Kontrolü

Fiilimsi SDM'sinden çıkan gövde direkt sonuç olarak kabul edilmeden önce gövde olarak etiketlenen kelime *gerundialResult* adında bir değişkenine kaydedilir ve son bir kontrol yapılır.

1. *gerundialResult* kaç tane "ğ" harfine sahip kontrol edilir.

- **a.** Bu harfe sahip değilse *gerundialResult* gövde olarak etiketlenir.
- **b.** En az bir tane "ğ" harfine sahipse bu harf veya harfler *gerundialResult*'un 1., 2., veya 3.harfi mi kontrol edilir.
  - **i.** Koşul sağlanmıyor ise *gerundialResult*'ta bulunan "ğ" harfi veya harfleri "k" harfi ile değiştirilir.
  - **ii.** Koşul sağlanıyor ise *gerundialResult* gövde olarak etiketlenir.

Bu kontrol ile kelimenin gövdesinde yer alan "ğ" harfi ile gövdesinde yer almayan fakat kendisine gelen ekler sebebiyle ünsüz yumuşamasına maruz kalarak "ğ" harfine dönüşen "k" harfi ayırt edilmiş olur. Türkçede anlamlı kelimenin en az iki veya üç harfli olmasından kaynaklı kelimenin 1., 2., veya 3.harfinden biri "ğ" ise gövdenin bir parçası olarak kabul edilir.

3.2.3. Kelime Türü Seçme

Tüm SDM'ler hazırlandıktan sonra bir kelimeye ait gövdeleme işlemi sonucunda ortaya çıkan gövdenin doğruluğu için kelime türü bulunmalıdır. Doğru gövdeyi bulmak için kelime, sırasıyla isim ve fiil SDM'lerine girdi olarak verilir. Her iki SDM'den çıkan gövdeler ve bulunan eklerin dizisi *nounResult* ve *verbResult* adlarında değişkenlere kaydedilir. Frekans analizinde kullanılması için bulunan gövdelerin kaydedileceği *results* adında bir dizi oluşturulur.

3.2.3.1. Tek Harfli Eklerin Kontrolü

Türkçedeki kaynaştırma harfleriyle birleşen tek harfli çekim ekleri bazen kelimenin gövdesine ait olabilmektedir. Fiillerde tek harfli veya kaynaştırma harfiyle birleşen tek harfli çekim ekleri olmadığı için sadece isimler için üç tane kontrol eklenmiştir.

1. *nounResult* içerisinde, bulunan eklerin olduğu dizi kontrol edilir eğer tek bir ek bulunmuş ise ve bu ek kaynaştırma ekleriyle birleşen tek harfli eklerden ise (ör. "-ye", "-ya", "-ni", "-nı", "-su", "-sü") bu ek kelimeye tekrardan eklenerek sözlükte aratılır.
    a. Eğer sözlükte yer alıyor ise frekans analizi için *results* dizisine kaydedilerek bir sonraki kelimenin gövdeleme işlemine geçilir.
    b. Eğer sözlükte yer almıyor ise 2.kontrole geçilir.
2. *nounResult* içerisinde, bulunan eklerin olduğu dizi kontrol edilir eğer iki ek bulunmuş ise ve ilk bulunan ek kaynaştırma ekleriyle birleşen tek harfli eklerden ise (ör. "-ye", "-ya", "-ni", "-nı", "-su", "-sü") ve son bulunan ek tek harfli ise (ör. “-e”, “-a”, “-i”, “-ı”) bu ekler kelimeye eklenerek tekrardan sözlükte aratılır.
    a. Eğer sözlükte yer alıyor ise frekans analizi için *results* dizisine kaydedilerek bir sonraki kelimenin gövdeleme işlemine geçilir.
    b. Eğer sözlükte yer almıyor ise sadece son bulunan tek harfli ek kelimeye eklenerek tekrardan sözlükte aratılır.
        i. Eğer sözlükte yer alıyor ise frekans analizi için *results* dizisine kaydedilerek bir sonraki kelimenin gövdeleme işlemine geçilir.
        ii. Eğer sözlükte yer almıyor ise 3.kontrole geçilir.
3. *nounResult* içerisinde, bulunan eklerin olduğu dizi kontrol edilir eğer tek bir ek bulunmuş ise ve bulunan ek tek harfli ise (ör. “-e”, “-a”, “-i”, “-ı”) ve kelime duruduma sözcükleri içerisinde yer alıyor ise bulunan ek kelimeye eklenerek tekrardan sözlükte aratılır.
    a. Eğer sözlükte yer alıyor ise frekans analizi için *results* dizisine kaydedilerek bir sonraki kelimenin gövdeleme işlemine geçilir.

b. Eğer sözlükte yer almıyor ise ekin kelimeye eklenmiş hali frekans analizi için *results* dizisine kaydedilerek sonraki kelimenin gövdeleme işlemine geçilir.

Bu üç kontrolden de kelime geçer ise *nounResult* ve *verbResult* içerisinde yer alan gövdeler yine sırasıyla baş harfleriyle başlayan sözlük içerisinde aratılır. Sözlük aramasından çıkan sonuçlar *prevNoun* ve *prevVerb* adlarında değişkenlere kaydedilir. Kelime türü bulunurken dört farklı durum oluşur.

- *prevNoun* sözlükte bulunursa kelime isim soylu kabul edilerek nihai gövde olarak *results* dizisine kaydedilir (1.durum).

- *prevNoun* sözlükte bulunamaz ise *prevVerb* kontrol edilir. *prevVerb* sözlükte bulunmuş ise kelime fiil soylu kabul edilerek nihai gövde olarak *results* dizisine kaydedilir (2.durum).

- Her ikisi de sözlükte bulunmaz ise *nounResult* ve *verbResult* içerisindeki bulunan ek sayıları karşılaştırılır. *nounResult* içerisindeki bulunan ek sayısı fazla ya da eşit ise kelime isim soylu kabul edilir fakat *prevNoun* gövde olarak etiketlenmez çünkü sözlükte bulunmamıştır (3.durum).

- *verbResult* içerisindeki bulunan ek sayısı daha fazla ise kelime fiil soylu kabul edilir fakat *prevVerb* gövde olarak etiketlenmez çünkü sözlükte bulunmamıştır (4.durum).

3.2.3.2. Sözlükte Bulunmayan Gövdelerin Kontrolü

1. ve 2.durumlarda kelime türü seçilmiştir fakat 3. ve 4.durumlar için sözlükte bir karşılığı bulunmadığı için ek kontroller gerçekleştirilir. 3.durum için aşağıdaki kontroller yapılır:

1. *nounResult* içerisindeki ek sayısı sıfır mı, en son kaydedilen ek tek harfli mi ve *prevNoun* boş mu kontrol edilir.

a. Üç koşulda sağlanıyor ise *nounResult* içerisindeki gövdenin sonuna, tek harfli ek geri eklenir ve gövde tekrardan baş harfiyle başlayan sözlükte aratılır.

i. Sözlükte bulunmaz ise kelime fiilimsi olarak kabul edilerek 2.adıma geçilir.

ii. Sözlükte bulunur ise *nounResult* içerisindeki gövde nihai gövde olarak *results* dizisine kaydedilir.

b. Üç koşuldan biri bile sağlanmıyor ise kelime fiilimsi olarak kabul edilerek 2.adıma geçilir.

2. Kelime fiilimsi SDM'sine girdi olarak verilir çıkan sonuç *gerundialResult* adında bir değişkene kaydedilir ve 3.adıma geçilir.

3. *gerundialResult,* baş harfiyle başlayan sözlükte, %60 benzerlikteki 10 kelimeyi kabul edecek şekilde aratılır ve sonuçlar *resultNoun* adında bir diziye kaydedilerek 4.adıma geçilir.

4. *nounResults* adında bir dizi oluşturularak *gerundialResult*'un harf uzunluğu üçten büyük mü kontrol edilir.

a. Üçten büyük değil ise üçe eşit mi kontrol edilir.

i. Eşit değil ise *resultNoun* dizisindeki elemanlardan sırasıyla kelimenin tüm harfleriyle eşleşenler *nounResults* dizisine eklenerek 5.adıma geçilir.

ii. Eşit ise *resultNoun* dizisindeki elemanlardan sırasıyla kelimenin ilk iki harfiyle eşleşenler *nounResults* dizisine eklenerek 5.adıma geçilir.

b. Üçten büyük ise *resultNoun* dizisindeki elemanlardan sırasıyla kelimenin ilk üç harfiyle eşleşenler *nounResults* dizisine eklenerek 5.adıma geçilir.

5. *nounResults* dizisi boş mu kontrol edilir.

a. Boş ise 4.adımdaki filtreleri uygulamadan *gerundialResult* sözlük içerisinde %60 benzerliğinde aratılarak ilk eşleşen kelime nihai gövde olarak seçilir ve *results* dizisine kaydedilir.

b. Boş değil ise *nounResults* dizisi içerisinden *findMostCommonWord* fonksiyonu kullanılarak gövdeye en yakın kelime nihai gövde olarak seçilerek *results* dizisine kaydedilir.

1.adımda *nounResult* içerisindeki gövdenin sözlükte yer almamasından dolayı tek harfli bir eke sahipse bu ekin gövdenin harfi olma olasılığı yüksek olduğu için tekrardan gövdeye ekleme işlemi yapılır. 2.adımda ise yeni oluşan gövde sözlükte bulunmaz ise kelimenin fiilimsi olduğuna karar verilir ve fiilimsi SDM'sine girdi olarak verilir. 3.adımda sözlükten %60 benzerlik oranına uyan 10 kelime seçilir ve 4.adımda bu kelimeler harf uzunluğuna göre filtrelenerek en uygunları *nounResults* dizisine kaydedilir. 5.adımda ise *nounResults* dizisi boş ise filtrelere uyan hiçbir kelime olmadığı için bir filtre yapılmadan %60 ve üstü benzerlikteki ilk eşleşen kelime sözlükten seçilerek nihai gövde belirlenir. Eğer *nounResults* dizisi boş değil ise dizideki kelimelerden en uygunu *findMostCommonWord* fonksiyonu ile seçilir ve nihai gövde belirlenir.

4.durum için ise aşağıdaki kontroller yapılır:

1. *verbResult,* baş harfiyle başlayan sözlükte, %60 benzerlikteki 10 kelimeyi kabul edecek şekilde aratılır ve sonuçlar *resultVerb* adında bir diziye kaydedilerek 2.adıma geçilir.

2. *verbResults* adında bir dizi oluşturularak *verbResult*'un harf uzunluğu üçten büyük mü kontrol edilir.

    a. Üçten büyük değil ise üçe eşit mi kontrol edilir.

        i. Eşit değil ise *resultVerb* dizisindeki elemanlardan sırasıyla kelimenin tüm harfleriyle eşleşenler *verbResults* dizisine eklenerek 3.adıma geçilir.

        ii. Eşit ise *resultVerb* dizisindeki elemanlardan sırasıyla kelimenin ilk iki harfiyle eşleşenler *verbResults* dizisine eklenerek 3.adıma geçilir.

    b. Üçten büyük ise *resultVerb* dizisindeki elemanlardan sırasıyla kelimenin ilk üç harfiyle eşleşenler *verbResults* dizisine eklenerek 3.adıma geçilir.

3. *verbResults* dizisi boş mu kontrol edilir.

    a. Boş ise 2.adımdaki filtreleri uygulamadan *verbResult* sözlük içerisinde %60 benzerliğinde aratılarak ilk eşleşen kelime nihai gövde olarak seçilir ve *results* dizisine kaydedilir.

b. Boş değil ise *verbResults* dizisi içerisinden *findMostCommonWord* fonksiyonu kullanılarak gövdeye en yakın kelime nihai gövde olarak seçilerek *results* dizisine kaydedilir.

4.durumun adımları 3.durum için oluşturulan 3., 4. ve 5.adımlarla aynıdır.

findMostCommonWord fonksiyonu, kendi içerisine *countCommonLetters* adında bir fonksiyon daha bulunduran bu fonksiyon, girdi olarak içerisine bir kelime dizisini tutacak *wordList* adında bir dizi ve yakınlık karşılaştırılmasının yapılacağı kelimeyi tutacak bir *targetWord* adında girdi değişkenleri alır. Sonuçta *countCommonLetters* fonksiyonunu da kullanarak dizideki kelimeler arasından bu kelimeye en yakınını seçer ve döndürür. Aşağıda *findMostCommonWord* fonksiyonunun adımları gösterilmiştir:

1. En yakın kelimenin kaydedileceği *mostCommonWord* değişkeni boş şekilde ve en fazla yakınlık puanını tutacak *maxCommonCount* değişkeni sıfır değeri ile oluşturulur.

2. *wordList* içerisindeki her kelime sırasıyla *targetWord* ile *countCommonLetters* fonksiyonuna gönderilir ve fonksiyondan dönen puan değeri *commonCount* değişkenine kaydedilir. Ardından *commonCount* değeri ile *maxCommonCount* değeri karşılaştırılır.

   a. *maxCommonCount* değeri daha büyük ise bir işlem yapılmaz ve dizideki bir sonraki kelimeye geçilir.

   b. *commonCount* değeri daha büyük ise *commonCount*'un değeri *maxCommonCount* değerine atanır ve puanı döndürülen kelime *mostCommonWord* içerisine kaydedilir.

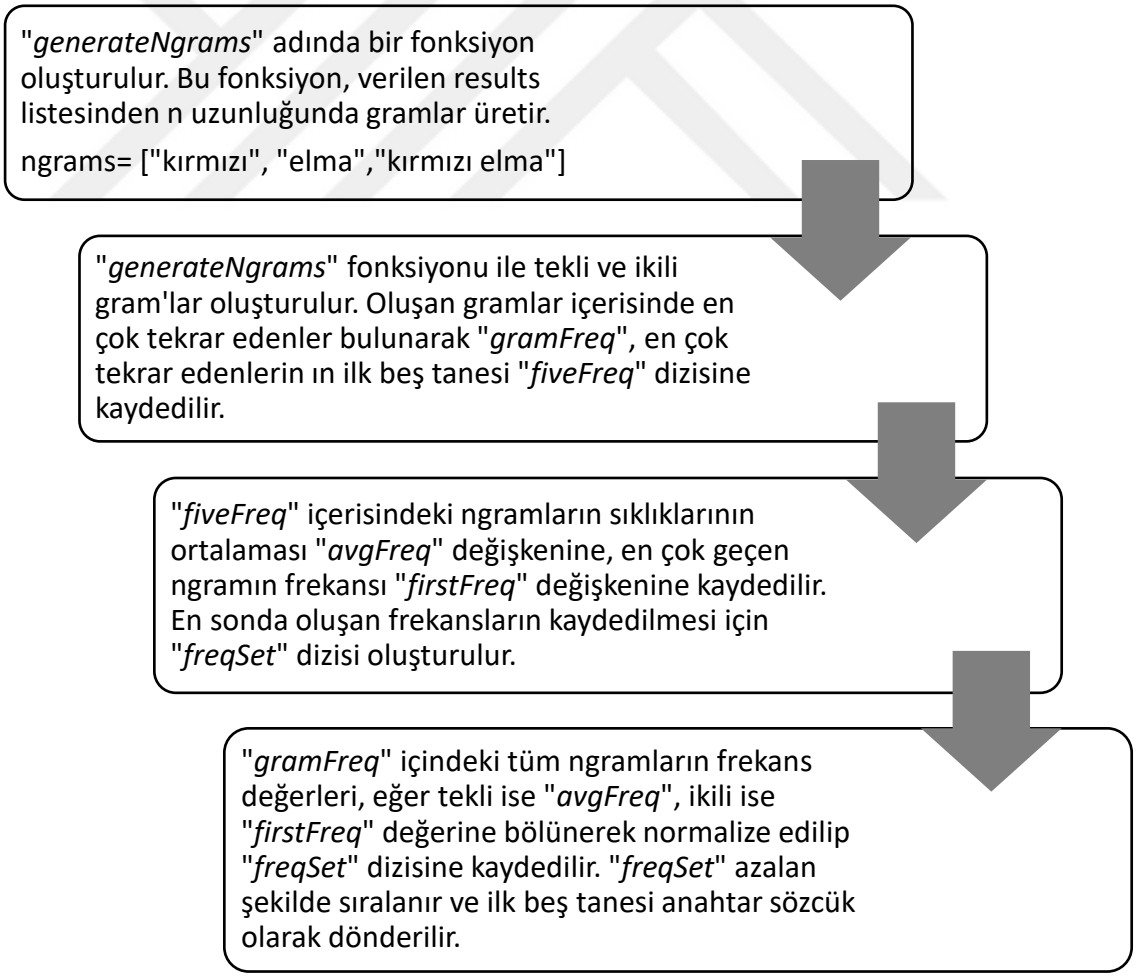
3. Tüm kelimeler için 2.adım uygulanır ve *mostCommonWord* fonksiyon sonucu olarak dönderilir.

countCommonLetters fonksiyonu ise girdi olarak *word1* ve *word2* olarak iki kelime alır, soldan sağa doğru ortak harf sayısını bularak puanlandırır ve bu puanı döndürür. İlk koşul olarak ilk kelimenin diğerinden harf uzunluğu olarak uzun ya da eşit olmasına bakar. Eğer bu koşul sağlanır ise harf sırasına göre farklı olan her harfte -1 puan düşürür. Aynı olan

harflerde ise bir önceki harfin aynı olması durumunda yeni harfe +1 puan verir fakat bir önceki harf farklı ise puan vermez. Eğer ikinci kelime ilk kelimeden daha uzun olur ise aynı şekilde puanlandırma yapılır fakat iki kelime arasındaki harf uzunluğu farkı puandan eksiltilir. Bu sayede ana kelime olan ilk kelimeden daha uzun olan ikinci kelimenin ne kadar ortak harf sayısı olsa da uzunluk bakımından ondan çok farklı olduğu saptanır.

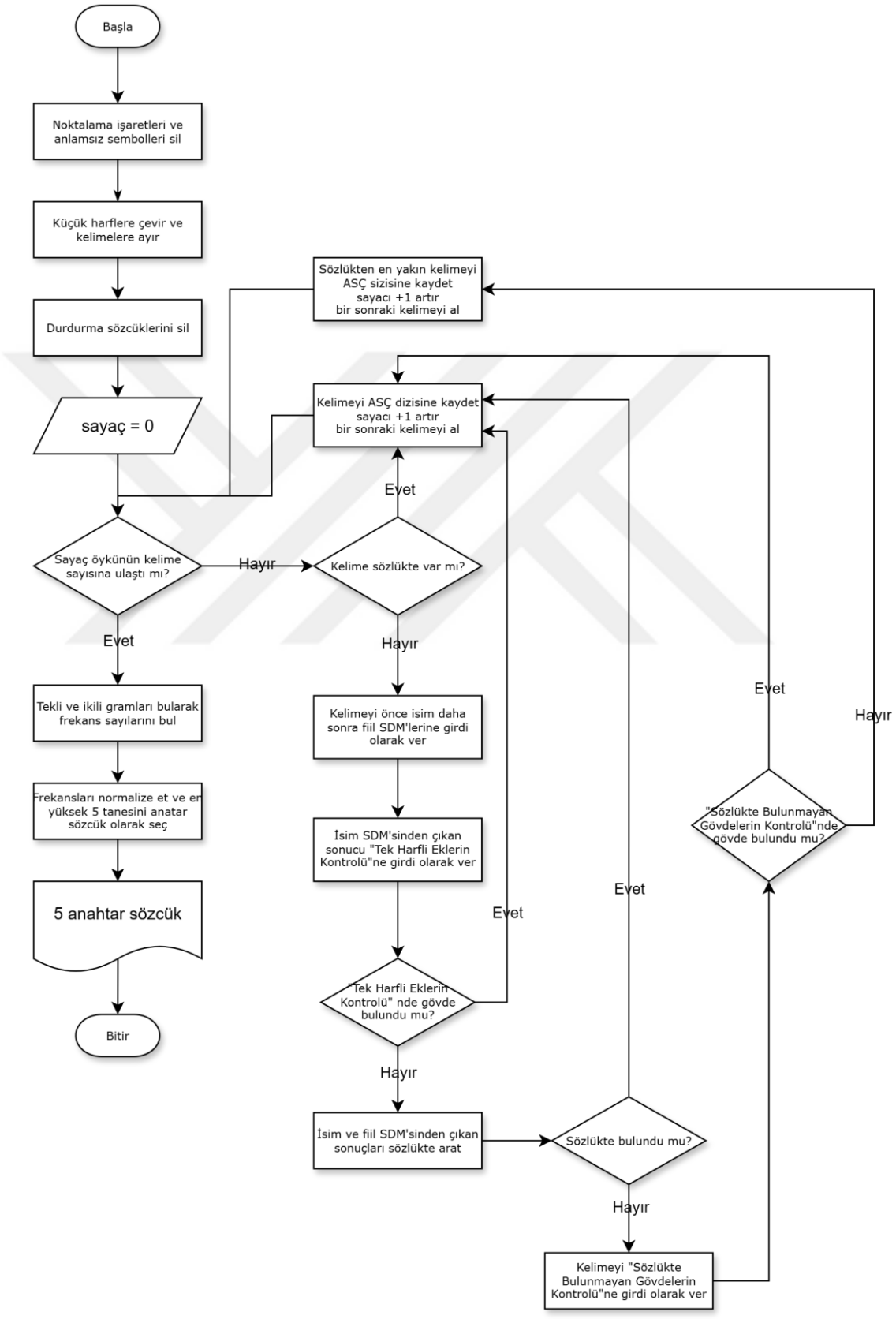
3.2.4. Frekans Analizi

Bu aşamada öykülerden elde edilen gövdelerin sıklık frekansları bulunur. Frekans analizinde kelimler "gram" olarak adlandırılırlar. Bu aşamada, sıklık frekansları tekli gram ve ikili gram olarak anahtar sözcük seçiminde kullanılırlar. Tekli gramlar ve ikili gramların frekans değerleri denklemler [46] ile normalize edilir. Böylece tekli ve ikili gramların anahtar sözcüğü olma ihtimalleri eşitlenmektedir.



Şekil 3.12. Frekans Analizi Adımları

Bir öykünün algoritmaya girdiğinde geçtiği tüm adımlar Şekil 3.13'te gösterilmiştir:



Şekil 3.13. Yöntem Akış Şeması

4. BÖLÜM

BULGULAR VE TARTIŞMA

4.1. Bulgular ve Tartışma

Öykülerin kayıtlı olduğu *"csv"* dosyasından sırasıyla öyküler bir döngü içerisinde ön işleme adımlarına tabi tutulurlar. Temizlenmiş ve SDM'ler için hazır hale getirilmiş kelimeler yine sırasıyla bir döngü içerisinde gövdeleme işlemine tabi tutulurlar. SDM'lerden gelen sonuçlar kelime türü seçme ve gövde kontrolünden geçtikten sonra frekans analizi gerçekleştirilir. Bu işlem sonucunda da o öyküye ait anahtar sözcükler belirlenmiş olur. Aşağıda *"csv"* dosyasındaki öykülerden birinin adım adım ASÇ sonucu gösterilmiştir.

ÖYKÜ
Aslan, eşek ve tilki birlikte bir av partisi düzenlemeye karar vermişler. Heyecanlı ve zorlu bir kovalamacadan sonra koskoca, semiz mi semiz bir geyik yakalamışlar. Çok acıkan aslan kükreyerek eşeğe, "Buraya gel arkadaş! Bir an evvel şu avı paylaştır yoksa açlıktan öleceğim", demiş. Eşek, adil olsun diye geyiği üç eşit parçaya bölmeye çalışırken aslan kükreyerek üzerine saldırıp kuvvetli bir pençe ile onu yere sermiş. Bu sefer tilkiye dönerek şöyle demiş: "Bakalım sen bu geyiği iki parçaya nasıl böleceksin?". Eşeğin başına geleni gören tilki, küçük bir parça ısırdıktan sonra avın bütününü aslana sunmuş. Bu paylaşmayı beğenen aslan başını sallayarak sormuş: "Aferin sana, hakça bir paylaşım yaptın. Bunu sana kim öğretti?". Tilki, "Arkadaşımız eşeğin sonu benim için iyi bir dersti efendim", diye karşılık vermiş.

ÖN İŞLEME SONUCU
aslan, eşek, tilki, av, partisi, düzenlemeye, karar, vermişler, heyecanlı, zorlu, kovalamacadan, koskoca, semiz, semiz, geyik, yakalamışlar, acıkan, aslan, kükreyerek, eşeğe, gel, arkadaş, an, avı, paylaştır, açlıktan, öleceğim, demiş, eşek, adil, olsun, geyiği, eşit, parçaya, bölmeye, çalışırken, aslan, kükreyerek, üzerine, saldırıp, kuvvetli, pençe, yere, sermiş, sefer, tilkiye, dönerek, demiş, bakalım, geyiği, parçaya, böleceksin, eşeğin, başına, geleni, gören, tilki, küçük, parça, ısırdıktan, avın, bütününü, aslana, sunmuş, paylaşmayı, beğenen, aslan, başını, sallayarak, sormuş, hakça, paylaşım, yaptın, öğretti, tilki, arkadaşımız, eşeğin, sonu, iyi, dersti, efendim, karşılık, vermiş.

Ön işleme adımında ilk olarak öykülerin içeriği Türkçe alfabesi dışındaki anlamsız sembollerden ve noktalama işaretlerinden temizlendi. Daha sonra ise kalan metin küçük harflere dönüştürülerek kelimeler birbirinden ayrıldı. Son olarak ise metinde geçen fakat metin hakkında bilgi vermeyen durdurma sözcüklerinden temizlendi.

Sırasıyla durdurma sözcükleri olarak temizlenen kelimeler: "*ve", "birlikte", "bir", "ve", "bir", "sonra", "mi", "bir", "çok", "buraya", "bir", "şu", "yoksa", "diye", "üç", "bir", "ile", "onu", "bu", "şöyle", "sen", "bu", "iki", "nasıl", "bir", "sonra", "bu", "aferin", "sana", "bir", "bunu", "sana", "kim", "benim", "için", "bir"*.

GÖVDELEME SONUCU
aslan, eşek, tilki, av, parti, düzenleme, karar, ver, heyecanlı, zorlu, kovalamaca, koskoca, semiz, semiz, geyik, yakala, acık, aslan, kükre, eşek, gel, arkadaş, an, av, paylaş, açlık, öl, de, eşek, adil, ol, geyik, eşit, parça, bölme, çalışır, aslan, kükre, üzeri, saldır, kuvvet, pençe, yer, ser, sefer, tilki, dön, de, bak, geyik, parça, böl, eşek, baş, gelen, gören, tilki, küçük, parça, ısır, av, bütün, aslan, sun, paylaşma, beğen, aslan, baş, salla, sor, hakça, paylaşım, yap, öğret, tilki, arkadaş, eşek, son, iyi, ders, efendi, karşılık, ver.

Gövdeleme sonucunda elde edilen kelime gövdeleri bazı durumlarda SDM'ye girmeden sözlükte bulunur. Bazı durumlarda ise doğru tespit edilen çekim eklerinin kelimeden silinmesiyle bulunur. Ya da yanlış çekim ekleri tespit edilse de belirlenen gövde, sözlükte

bulunmadığı için yapılan gövde kontrolleri ile doğru gövdeler bulunur. Direkt olarak sözlükte bulunan gövdeler şu şekildedir:

“aslan” → “aslan” (Gövde)

“parça” → “parça” (Gövde)

“karşılık” → “karşılık” (Gövde)

Doğru tespit edilen çekim eklerinin kelimeden silinmesiyle bulunan gövdeler şu şekildedir:

“partisi” → “parti” (Gövde) + “-si” (İyelik Eki)

“vermişler” → “ver” (Gövde) + “-miş” (Dilek/Haber Kipi) + “-ler” (Şahıs Eki)

Yanlış çekim ekleri tespit ederek sözlükte bulunmadığı için gövde kontrolleriyle doğru bulunan gövdeler şu şekildedir:

(Yanlış) “düzenlemeye” → “düzenlem” (Gövde) + “-e” (İyelik Eki) + “-ye” (Hal Eki)

(Doğru) “düzenlemeye” → “düzenleme” (Gövde) + “-ye” (Hal Eki)

(Yanlış) “aslana” → “asl” (Gövde) + “-a” (İyelik Eki) + “-na” (Hal Eki)

(Doğru) “aslana” → “aslan” (Gövde) + “-a” (Hal Eki)

“düzenlemeye” kelimesi içerisindeki hal ekinin silinmesinden sonra “-me” ekinin yapım eki olmasından dolayı kelime gövdesi “düzenle” değil “düzenleme” olarak bulunmuştur.

Eğer kelime “düzenleme” olarak verilmiş olsaydı “-me” eki fiil SDM’sinden olumsuzluk eki olarak çıkarak “düzenle” gövdesine ulaşılırdı. Fakat doğruluğu için kelimenin cümle içindeki bağlamına bakılması gereklidir. Böylelikle “-me” ekinin olumsuzluk eki mi yoksa yapım eki mi olarak kullanıldığı belirlenebilir. Fakat bu yöntemde bağlama bakılmadan gövde bulma işlemi yapıldığından ve sözlükte “düzenle” kelimesi bulunacağı için doğru olarak kabul edilecekti.

Tablo 4.1. Tekli Gramların Frekans Değerleri

Kelime	Frekans
aslan	1.25
eşek	1.25
tilki	1.00
av	0.75
geyik	0.75

Frekans analizinde tekli gramlar kelimelerin tek başına kendisini ifade etmektedir. Tekli gramların en yüksek beş frekans değerleri Tablo 4.1'deki gibidir.

Tablo 4.2. İkili Gramların Frekans Değerleri

Kelime	Frekans
aslan kükre	1.00
aslan eşek	0.50
eşek tilki	0.50
tilki av	0.50
av parti	0.50

İkili gramlar kendisiyle beraber sağındaki ve solundaki kelimeyle olan birleşmesini ifade etmektedir. İkili gramların en yüksek beş frekans değerleri ise Tablo 4.2'deki gibidir.

FREKANS ANALİZİ SONUCU
[("aslan", 1.25), ("eşek", 1.25), ("tilki", 1.00), ("aslan kükre", 1.00), ("av", 0.75)]

Tüm işlemlerin ardından en yüksek beş frekans değerleri öykünün anahtar sözcükleri olarak seçilir.

Gövdeleme için daha önce oluşturulmuş bir test verisi bulunamamıştır ve önceki çalışmalarda yapılan test verileri ise ya yayınlanmamış ya da gövdeleme için değil kök bulma işlemi için oluşturulmuştur. Bu yüzden toplanan kısa öykülerden 100 tanesi seçilip test için her bir kelimenin doğru gövdesi manuel olarak belirlenmiştir. Toplamda 10 bin kelimelik bir test verisi oluşturulmuştur.

Test için kısa öyküler üzerinde yapılmış gövdeleme sonuçları Tablo 4.3'te gösterilmiştir. Toplam 10 bin kelimeden oluşan test işleminde özellikle bu çalışma içinde kullanılabilirliği asıl amaç olduğu için kısa öyküler üzerinde yapılmasının daha doğru olacağına karar verilmiştir.

Tablo 4.3. Gövdeleme Test Sonucu

Yöntem	Doğru Gövde Sayısı	Yanlış Gövde Sayısı	Doğruluk Oranı
Geliştirilen Yöntem	8140	1860	%81.43
Zemberek	7863	2137	%78.63
Snowball	6039	3961	%60.39

Test işleminde geliştirilen yöntem ile Zemberek'in gövdeleme yöntemi [14] ve Snowball [24] yöntemleri karşılaştırılmıştır. Diğer çalışmaların kaynak kodları ve materyalleri paylaşılmadığı için testler için kullanılacak yöntemler kısıtlı sayıdadır. Özellikle gövdeleme alanında yapılmış çalışma sayısının az olması ve diğer yöntemlerinde daha çok kök bulma üzerine tasarlanmış olması da bir etkendir.

Tablo 4.3'te geliştirilen yöntem ile diğer yöntemlerin bulmuş oldukları toplam doğru ve yanlış gövde sayılarıyla beraber doğruluk oranları gösterilmiştir. Bu oranlara göre geliştirilen yöntemin diğer yöntemler göre daha başarılı olduğu gözükmüştür.

Tablo 4.4'te ise Her yöntemin yanlış bulduğu gövdelerin nelerden kaynaklandığı ve sayıları gösterilmiştir.

Tablo 4.4. Yanlış Gövdeleme Sebepleri ve Sayıları

Yöntem	Fazla Ek Bulma	Eksik Ek Bulma	Ses Olayları	Diğer
Geliştirilen Yöntem	568	1028	137	127
Zemberek	1431	223	367	116
Snowball	1547	1924	306	184

Geliştirilen yöntemin diğer ve eksik ek bulmadan kaynaklanan gövdeler incelendiğinde yanlış bulduğu en fazla gövde sözlükten kaynaklı olduğu tespit edilmiştir. Geliştirilen yöntem doğru ekleri bulup kelimeden silse de sözlük kontrolünde bulunan ve gövde olarak kaydedilen kelimenin TDK sözlüğünden kaynaklı olarak sonu yapım ekleriyle (ör. "koşma", "tutuş", "vurmak") ve çoğul ekleriyle bitmesinden kaynaklanmaktadır.

(Yanlış) "koştum" → "koşma"

(Doğru) "koştum" → "koş"

(Yanlış) "kurbağalardı" → "kurbağalar"

(Doğru) "kurbağalardı" → "kurbağa"

Ayrıca Zemberek'in sözlüğünde bulunan anlamsız ve yabancı kelimelerden dolayı da çoğu kelime gövdesi yanlış tespit edilmiştir.

(Yanlış) "suyu" → "suzy"

(Doğru) "suyu" → "su"

Sözlük içerisindeki sadece bu test işleminde hatalı sonuç veren 276 kelime düzeltildiğinde geliştirilen yöntemin doğruluk oranı artmıştır.

Tablo 4.5. Sözlük Düzeltme Sonucu

Yöntem	Doğru Gövde Sayısı	Yanlış Gövde Sayısı	Doğruluk Oranı
Geliştirilen Yöntem	8416	1584	%84.16

Ses olaylarından kaynaklanan hatalar çoğu yöntemde olduğu gibi geliştirilen yöntemde de yanlış sonuçlar verebilmektedir.

(Yanlış) “karnı” → “karn”

(Doğru) “karnı” → “karın”

Geliştirilen yöntemde fazla ek bulma sorunu ise daha çok yapım ekleriyle aynı olan çekim eklerinden kaynaklanmaktadır.

(Yanlış) “kararmış” → “kar” (Gövde) + “-ar” (1.Haber Kipi) + “-mış” (2.Haber Kipi)

(Doğru) “kararmış” → “karar” (Gövde) + “-mış” (1.Haber Kipi)

Yine geliştirilen yöntemde eksik ek bulma sorunu ise yine yapım ekleriyle aynı olan çekim eklerinden kaynaklanmaktadır.

(Yanlış) “vurmadı” → “vurma” (Gövde) + “-dı” (1.Haber Kipi)

(Doğru) “vurmadı” → “vur” (Gövde) + “-ma” (Olumsuzluk Eki) + “-dı” (1.Haber Kipi)

Test için ayrılmış 100 kısa öykü üzerinde yapılmış ASÇ, her öykü için anlamlı sonuçlar vermiş ve önceki çalışmalarda da kanıtlandığı gibi sözcük frekanslarının ASÇ’de etkili olduğu bu çalışmada da kanıtlanmıştır. Özellikle ASÇ için herkes tarafından kabul edilecek bir sonuç bulmak tahmin edildiği üzere zordur. Fakat genel olarak metin hakkında bilgi veren ve metnin tamamını okumadan okuyucunun kafasında bir fikir oluşmasını sağlamak anahtar sözcükler için kabul görmüş bir durumdur. Bundan dolayı genel bir doğru ve yanlış ayrımının yapılamayacağı için farklı yöntemlerle karşılaştırma işlemine gerek görülmemiştir.

5. BÖLÜM

SONUÇ VE ÖNERİLER

5.1. Sonuç ve Öneriler

Bu tez çalışması kapsamında Türkçe kısa öyküler üzerinde ASÇ yapılması hedeflenmiş ve bu amaç doğrultusunda yeni bir gövdeleme yöntemi geliştirilerek başarılı bir sonuç elde edilmiştir. ASÇ uygulanırken kelime frekanslarından yararlanılarak anahtar sözcükler bulunmuştur.

Günümüzde daha çok öğrenmeli dil modelleri kullanılarak ASÇ ve gövdeleme işlemeleri denense de henüz kural tabanlı yöntemlerden daha iyi sonuç veren bir çalışma görülmemiştir. Bu çalışmayla beraber Türkçe dili için daha önce yapılmış gövdeleme ve kök bulma yöntemleri detaylıca incelenmiş ve çekim eklerinin doğru sınıflandırılması ve gövdeleme işleminde kullanılacak Türkçe dil kurallarının daha detaylı koşullandırmalarla daha yüksek başarı oranları elde edilebileceği düşünülerek, daha fazla koşul ve kural tanımıyla yeni bir yöntem geliştirilmiştir. Geliştirilen yöntemle beraber daha yüksek başarı oranına ulaşılmıştır.

Türkçe dili üzerine yapılan morfolojik ve dil yapısını ilgilendiren çalışmalarda sözlük kullanımının önemi bu çalışma ile bir kez daha anlaşılmıştır. Çalışmada kullanılacak sözlüğün doğru kelime gövdeleri veya köklerine sahip olduğu durumlarda doğruluğunda aynı şekilde yükseleceği kanıtlanmıştır.

Bundan sonraki çalışmalarda bu yöntemin geliştirilmesine devam edilmesi planlanmaktadır. Çalışmanın eksik görülen kısımları aşağıda belirtilmiştir. Bu eksik kısımlar ileride yöntemin geliştirilmesiyle kapatılabilecektir.

- Ses olaylarına karşı daha hassas bir yapı sağlanabilir. Bunun için her bir ses olayını kontrol edecek koşullar ve kurallar tanımlanabilir.

- Yanlış yazılmış kelimeleri düzeltecek bir kontrol modülü geliştirilerek daha doğru bir gövdeleme işlemi yapılabilir.

- Kesme işareti veya soru kalıpları (ör. "yapıldı mı?") kontrol edilerek ayrı bir kontrol modülü geliştirilebilir.

- Kelimelerin cümle içerisindeki rolleri daha detaylı bir şekilde bulunarak POS (Part of speech) etiketleme yapılabilir.

- Sözlükler *"csv"* dosyası yerine daha farklı bir veri yapısı içerisinde tutularak arama performansı geliştirilebilir.

- İçeriği daha doğru ve temizlenmiş bir sözlük kaynağı oluşturulabilir.

- ASÇ'de sadece kelime frekansına dayalı olmayan ve kelimenin cümle içindeki bağlamının da değerlendirilmesiyle daha doğru sonuçlar elde edilebilir.

KAYNAKÇA

[1] Khurana, D., Koli, A., Khatter, K., Singh, S. 2023. Natural language processing: state of the art, current trends and challenges. Multimedia Tools and Applications, 82 (3).

[2] Lovins, J. B. 1968. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11 (June).

[3] Porter, M. F. 1980. An algorithm for suffix stripping. Program: electronic library and information systems 14 (3).

[4] Koskenniemi, K., Tapanainen, P., Voutilainen, A. 1992. Compiling and using finite-state syntactic rules. In COLING: The 14th International Conference on Computational Linguistics, 1992, Nantes.

[5] Krovetz, R. 1993. Viewing morphology as an inference process, pp. 191-202. Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Infofmation Retrieval.

[6] Oflazer, K. 1994. Two-level description of turkish morphology. Literary and Linguistic Computing, 9 (2).

[7] Antworth, E. L. 1990. PC-KIMMO: A Two-Level Processor for Morphological Analysis. Summer Institute of Linguistics.

[8] Cebiroğlu, G., Adalı, E. 2002. Root reaching method without dictionary. In TBD 19. Bilişim Kurultayı.

[9] Sever, H., & Bitirim, Y. 2003. FindStem: Analysis and Evaluation of a Turkish Stemming Algorithm, pp. 238–251. In International Symposium on String Processing and Information Retrieval, 2003, Berlin Heidelberg.

[10] Eryiğit, G., Adali, E. 2004. An affix stripping morphological analyzer for Turkish. Proceedings of the IASTED International Conference, 2004, Austria.

[11] Varlıklar, Ö. 2005. Development of a Method to Determine Root and Suffixes for Turkish Words to Generate Large Scale Turkish Corpus. Dokuz Eylül Üniversitesi, Yüksek Lisans Tezi, İzmir, 79 s.

[12] Çilden, E. K. 2006. Stemming Turkish words using snowball. Snowball ile Çalışmalar Raporu.

[13] Yüret, D., Türe, F. 2006. Learning morphological disambiguation rules for Turkish, pp. 328–334. Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, 2006, New York.

[14] Akın, A. A., Akın, M. D. 2007. Zemberek, an open source NLP framework for Turkic languages. Structure, 10, (1-5).

[15] Sak, H., Güngör, T., Saraçlar, M. 2007. Morphological disambiguation of Turkish text with perceptron algorithm pp. 107-118. In International Conference on Intelligent Text Processing and Computational Linguistics, 2007, Berlin Heidelberg.

[16] Collins, M., Duffy, N. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2002.

[17] Sak, H., Güngör, T., Saraçlar, M. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus, pp. 417-427. In International conference on natural language processing, 2008, Berlin Heidelberg.

[18] Birant, Ç. C. 2008. Root-Suffix Seperation of Turkish Words. Dokuz Eylül Üniversitesi, Yüksek Lisans Tezi, İzmir, 64 s.

[19] Çöltekin, Ç. 2010. A freely available morphological analyzer for Turkish. Proceedings of the 7th International Conference on Language Resources and Evaluation, 2010.

[20] Ak, K., Yıldız, O. T. 2011. Unsupervised Morphological Analysis Using Tries. Işık Üniversitesi, Yüksek Lisans Tezi, 49 s.

[21] Şahin, M., Sulubacak, U., Eryigit, G. 2013. Redefinition of Turkish morphology using flag diacritics. In The 10th Symposium on Natural Language Processing.

[22] Şahin, M. 2014. ITUMorph - a more accurate and faster wide coverage morphological analyzer for Turkish. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2013.

[23] Eryiğit, G. 2014. ITU Turkish NLP Web Service, pp. 1-4. Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics.

[24] Tunçelli, O. 2018. Stemmer algorithm for Turkish. (Web sayfası: https://github.com/otuncelli/turkish-stemmer-python), (Erişim tarihi: Temmuz 2025).

[25] Zafer, H. R. 2018. RESHA: Stemmer for Turkish. (Web sayfası: https://github.com/harunzafer/resha-turkish-stemmer), (Erişim tarihi: Temmuz 2025).

[26] Zafer, H. R. 2018. NUVE: A Natural Language Processing Library for Turkish. (Web sayfası: https://github.com/harunzafer/nuve), (Erişim tarihi: Temmuz 2025).

[27] Tintin, R. 2017. Gövde-Türk: A Turkish stemming method. Çanakkale Onsekiz Mart Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2017, Çanakkale.

[28] Can, B. 2019. Stemming Turkish Words with LSTM Networks. Bilişim Teknolojileri Dergisi, 12 (3).

[29] Ekin, M. F. 2020. A New Method for Turkish Morphological Analysis. Maltepe Üniversitesi, Yüksek Lisans Tezi, 2020, İstanbul.

[30] Özbey, C., & Karcılı, A. 2021. Unsupervised Lexicon-Based Stemming by Dual Dictionary Models. In 2021 Innovations in Intelligent Systems and Applications Conference, ASYU, 2021.

[31] Aleçakır, H., Bölücü, N., Can, B. 2022. TurkishDelightNLP: A Neural Turkish NLP Toolkit. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session.

[32] Özgür, A., Özgür, L., Güngör, T. 2005. Text categorization with class-based and corpus-based keyword selection. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

[33] Külekci, M. O. 2007. TARA: An algorithm for fast searching of multiple patterns on text files. 22nd International Symposium on Computer and Information Sciences, ISCIS, 2007.

[34] Kocatekin, T., Ünay, D. 2013. Text mining in radiology reports. 21st Signal Processing and Communications Applications Conference (SIU), 2023.

[35] Özyirmidokuz, E. K. 2014. Mining Unstructured Turkish Economy News Articles. Procedia Economics and Finance, 16.

[36] Yeşim İkikat, F., Gürhan, B., Diri, B. 2015. Automatic linking of wikipedia pages by their semantic similarity. International Symposium on Innovations in Intelligent SysTems and Applications, 2015.

[37] Savaş, S., Topaloğlu, N. 2017. Crime intelligence from social media: A case study. IEEE 14th International Scientific Conference on Informatics, 2017.

[38] Müngen, A. A., Kaya, M. 2018. Extracting abstract and keywords from context for academic articles. Social Network Analysis and Mining, 8 (1).

[39] Demirci, G. M., Keskin, S. R., Dogan, G. 2019. Sentiment Analysis in Turkish with Deep Learning. IEEE International Conference on Big Data, 2019.

[40] Aygun, I., Kaya, M. 2020. Automatic Term Extraction on Turkish Scientific Texts. 2020 International Conference on Decision Aid Sciences and Application, DASA, 2020.

[41] Erzurumlu, H. Y. 2022. Keyword extraction from speech data using center based clustering. Gebze Teknik Üniversitesi, Yüksek Lisans Tezi, Gebze.

[42] Ayan, E. T., Arslan, R., Zengin, M. S., Duru, H. A., Salman, S., Bardak, B. 2021. Turkish keyphrase extraction from web pages with BERT. 29th IEEE Conference on Signal Processing and Communications Applications, Proceedings, SIU, 2021.

[43] Gürbüz, T., Uluyol, Ç. 2023. Research article classification with text mining method. Concurrency and Computation: Practice and Experience, 35 (1).

[44] Göz, F. 2022. New approaches for keyword extraction in academic articles. Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Kocaeli.

[45] Kılıç, H. 2023. A novel sequential labeling based hybrid keyword extraction model. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi.

[46] Kat, B. 2023. Natural language processing for the Turkish Academic texts in the engineering field and development of a decision support system: the case of TUBITAK project proposals. Mühendislik Mimarlık Fakültesi Dergisi, 14.

[47] Yayla, K. 2024. Enhancing Access to Scientific Articles in the Turkish Librarianship Journal Digital Archive: Assigning Keywords from Turkish Abstracts. Türk Kütüphaneciliği, 38 (4).

[48] Aydın, Ö., Kantarcı, H. 2024. A Comparison of LSTM and BERT Based Models in Turkish Keyword Extraction. Bilgisayar Bilimleri ve Mühendisliği Dergisi, 17 (1).

[49] Yusuf, B., Saraçlar, M. 2024. Written Term Detection Improves Spoken Term Detection. Transactions on Audio, Speech, and Language Processing, 32 (3213–3223).

[50] Babayigit, B., Sattuf, H. 2025. BERT-based keyword extraction model for the Turkish language. Neural Computing and Applications, 37 (16).

[51] Banguoğlu, T. 1974. Türkçenin Grameri. Baha Matbaası, İstanbul, 140 s.

[52] Demir, N., Yılmaz, E. 2011. Türkçe Ses Bilgisi. Anadolu Üniversitesi, Eskişehir, 167 s.

[53] Çarkacı, N. 2020. TDK - Türkçe Sözlük Kelime Toplayıcı. (Web sayfası: https://github.com/ncarkaci/TDKDictionaryCrawler), (Erişim tarihi: Temmuz 2025)

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı:

Uyruğu:

Doğum Tarihi ve Yeri:

Medeni Durum:

e-mail:

Yazışma Adresi:

EĞİTİM

Derece	**Kurum**	**Mezuniyet Tarihi**
Lisans	Erciyes Üniversitesi, Bilgisayar Mühendisliği	2021
Lise	Çözüm Temel Lisesi, Kayseri	2017

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görev**
2024-Halen	Erciyes Üniversitesi	Araştırma Görevlisi

YABANCI DİL

İngilizce

YAYINLAR

1. Bağırgan F., Görkemli B., Keyword Extraction from Turkish Stories using Natural Language Processing Techniques, ICENSS 2025: International Congress of Engineering and Natural Sciences Studies, 2025, Ankara, Turkey