TÜRKÇEDE MORFOLOJİK ANALİZ YAPAN BİR SİSTEMİN MORFOLOJİK TÜRETME İÇİN KULLANILMASI

USING A TURKISH MORPHOLOGICAL ANALYZER FOR WORD GENERATION

MUSTAFA BURAK ÖZTÜRK

YRD. DOÇ. DR. BURCU CAN BUĞLALILAR

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

Ağustos 2016

MUSTAFA BURAK ÖZTÜRK' ün hazırladığı **"TÜRKÇEDE MORFOLOJİK ANALİZ YAPAN BİR SİSTEMİN MORFOLOJİK TÜRETME İÇİN KULLANILMASI"** adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI** 'ında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Prof. Dr. Cem BOZŞAHİN
Başkan ....

Yrd. Doç. Dr. Burcu CAN BUĞLALILAR
Danışman ....

Doç. Dr. Harun ARTUNER
Üye ....

Yrd. Doç. Dr. Cengiz ACARTÜRK
Üye ....

Yrd. Doç. Dr. Gönenç ERCAN
Üye ....

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Salih Bülent ALTEN
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı.

beyan ederim.

15/08/2016

MUSTAFA BURAK ÖZTÜRK

ÖZET

TÜRKÇEDE MORFOLOJİK ANALİZ YAPAN BİR SİSTEMİN MORFOLOJİK TÜRETME İÇİN KULLANILMASI

Mustafa Burak ÖZTÜRK

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Burcu CAN BUĞLALILAR

Ağustos 2016, 66 sayfa

Makine çevirisi, soru-yanıt sistemleri gibi doğal dil işleme uygulamalarında sözdizim ve anlama göre sözcük formlarının morfolojik olarak türetilmesine ihtiyaç duyulur. Türkçe, zengin ve üretken bir morfolojiye sahiptir. Bir Türkçe sözcük binlerce farklı sözcük formuna sahip olabilmektedir. Bu özellikleriyle Türkçe, morfolojik üretme gibi doğal dil işleme çalışmalarında zorlu ve ilgi çekici bir dil olmuştur.

Bu çalışmada, Türkçe sözcükleri denetimsiz olarak türetebilen bir model önerilmiştir. Çalışmada Türkçede denetimli morfolojik analiz yapan bir sistem ile sözcükler morfemlerine ayrıştırılmıştır. Sonlu durum özdevinirleri (FSA), Türkçenin sözdizimsel özelliklerini ele alabilmek için kullanılmıştır. Çalışmada sözcük kökleri sözdizimsel özelliklerine göre (isim, fiil vb.) kategorilerine ayrılmıştır. Benzer olarak ekler de alomorfik özelliklerine göre kategorilendirilmiştir. Her bir kök kategorisi için bir FSA oluşturulmuştur. Bu FSA'ların başlangıç durumları bir kök kategorisine karşılık gelirken, takip eden durumlar ise bir ek kategorisine karşılık gelmektedir. FSA'lar kullanılarak Türkçe sözcükler türetilmiştir. Buna ek olarak

Türkçenin yazımsal özellikleri denetimsiz olarak keşfedilmiş ve sözcük türetme için kullanılmıştır. Yaklaşık 3000 biricik sözcük kökünden, 1 milyon civarı sözcük türetilmiştir. Geliştirdiğimiz bu model, %82.36 doğruluk oranıyla sözcük formu türetebilmektedir. Bu çalışmada önerilen model, Fince ve Macarca gibi diğer sondan eklemeli ve zengin morfolojiye sahip diller için de uygulanabilir.

Anahtar Sözcükler: Doğal dil işleme, denetimsiz öğrenme, morfoloji, sözcük türetme, morfolojik üretme, sonlu durum özdevinirleri

ABSTRACT

USING A TURKISH MORPHOLOGICAL ANALYZER FOR WORD GENERATION

Mustafa Burak ÖZTÜRK

Master of Science, Computer Engineering Department

Supervisor: Asst. Prof. Dr. Burcu CAN BUĞLALILAR

August 2016, 66 pages

Generating word forms accordingly with the syntax and semantics is needed by natural language processing applications such as machine translation and question answering. Turkish morphology is rich and productive. A Turkish word can have thousands of different word forms. With these features, Turkish is a challenging and attractive language for natural language processing tasks such as morphological generation.

In this study, a model that generates Turkish words in an unsupervised way is presented. In the study, a supervised Turkish morphological analyzer is used for splitting words into morphemes. We used finite state automatas (FSA) to deal with morphosyntactic features. In the study, stems are categorized based on their syntactic features (i.e. noun, verb, etc.). Similarly, suffixes are categorized based on their allomorphic features. An FSA is built for each stem category. Start states of these FSAs correspond a stem category whereas following states correspond a suffix category. Turkish words are generated by using these FSAs. Additionally, Turkish ortographic features are extracted with an unsupervised approach and these features are used for generating words. From approximately 3000 unique stems, around 1 million words are generated. The model that we developed can generate word forms with an

accuracy of %82.36. The model proposed in this study, can be applied to other agglutinative languages, such as Hungarian and Finnish that are also morphologically rich.

Keywords: Natural language processing, unsupervised learning, morphology, word generation, morphological generation, finite state automatas (FSAs)

TEŞEKKÜR

Çalışmalarım boyunca bilgisinden ve tecrübelerinden faydalandığım, göstermiş olduğu hoşgörü ve sabırla bana her açıdan destek olan danışmanım Sayın Yrd. Doç. Dr. Burcu CAN BUĞLALILAR'a,

Tez savunmam sırasındaki değerli yorumları ve önerileri sebebiyle jüri üyelerim Sayın Prof. Dr. Cem BOZŞAHİN'e, Sayın Doç. Dr. Harun ARTUNER'e, Sayın Yrd. Doç. Dr. Cengiz ACARTÜRK'e, Sayın Yrd. Doç. Dr. Gönenç ERCAN'a,

Çalışmalarım sırasında, sorularıma sıkılmadan cevap veren ve yardımlarını esirgemeyen Columbia Üniversitesi doktora öğrencisi Sayın Mohammad Sadegh Rasooli'ye,

Bu tez çalışmaları sırasında ve tüm hayatım boyunca, desteğini benden hiçbir zaman ve hiçbir konuda esirgemeyen, can yoldaşım, sevgili ablam Özge BORA'ya,

Beni çok seven, her zaman destekleyen ve güvenen, benim için yaptıkları büyük fedakarlıklarla tüm başarılarımın mimarı olan sevgili aileme,

Çalışmalarım sırasında sabırla, sevgiyle bana destek olan, zor zamanlarda benim devam etmemi sağlayan, bana her zaman inanan, her zaman yanımda olan ve geri kalan tüm hayatım boyunca da yanımda olmasını istediğim birtaneme, Yelda YÜCEER'e teşekkürlerimi sunarım.

İçindekiler

Sayfa

ÖZET i
ABSTRACT iii
TEŞEKKÜR v
İÇİNDEKİLER vi
ÇİZELGELER viii
ŞEKİLLER x
SİMGE VE KISALTMALAR x
1. GİRİŞ 1
2. ALAN BİLGİSİ VE ALANYAZIN 6
2.1. Doğal Dil İşleme 6
2.2. Morfoloji 7
2.3. Morfolojik Çözümleme 10
2.4. Morfolojik Türetme 13
3. MODEL 21
3.1. Ön Çalışma ve Motivasyon 21
3.2. Veri Kümesi 24
3.3. Kök Kategorilerinin Bulunması 24
3.4. Ek Kategorilerinin Bulunması 36
3.5. Harf İkilileri 38
3.6. Ortografik (Yazımsal) Kurallar 41
3.7. Sözcük Türetme 43
3.8. Sözcük Türetme Deney ve Sonuçları 46
4. KARŞILAŞTIRMA 53
5. SONUÇLAR VE TARTIŞMA 57
REFERENCES 59

ÇİZELGELER

2.1. Bhavsar'ın tanımladığı kurallar [1]. **Ç:K** - çıkarılacak harf (trimming character, **GNPC** - gnp ve durum özellikleri, **Rev** - ters işlem, **x ve y** - gerçek ekler, **sNoun** - kaynak sözcük kategorisi (isim) ve **tNoun** - hedef sözcük kategorisi . 17

3.1. Örnek girdi, eklerine ayrılmış ve türleriyle beraber işaretlenmiş sözcükler 23
3.2. Üretilmiş sözcüklere örnekler 23
3.4. JS ıraksama metriği için kullanılan sınır değerlerinin oluşan kategori sayısı ve saflık değerleri üzerindeki etkisi..... 29
3.3. JS ıraksama ve Jaccard mesafesi yöntemlerinin 100, 200, 300, 400, 500 sözcüklük veri kümeleriyle gerçekleştirilen deney sonuçları sırasıyla saflık değeri ve son kategori sayılarıyla birlikte verilmiştir..... 29
3.5. Eklerine ayrılmış olan *gerçekleştirdiğinin* sözcüğü için, pencere boyutunun 2 olarak seçilmesi ile oluşan pencereler. 31
3.6. word2Vec modeli ile oluşturulan kök kategorilerinden bazıları 32
3.7. CBOW kümeleme sonuçları 32
3.8. Skip-gram kümeleme sonuçları 32
3.9. Karşılıklı bilgi tabanlı metrik ile elde edilen sonuçlar..... 34
3.10. Köklerin kategorilerine ayrılması işlemi sonucunda oluşan bazı kategoriler ... 35
3.11. Bazı ek kategorileri 38
3.12. Bazı harf ikililier ve onların görülme olaslıkları..... 39
3.13. Son kurallar kümesi 42
3.14. Karşılıklı bilgi tabanlı metrik ile oluşturulan sözcük kökü kategorileri ve onlarla üretilen sözcükler 48
3.15. Jensen-Shannon ıraksama metriği ile birleştirilmeye devam eden kök kategorileri ve üretilen sözcükler 48
3.16. Son olarak elde edilen kök kategorileri ve üretilen sözcükler..... 48
3.17. word2vec modeli ile elde edilen kök kategorileriyle üretilen sözcükler..... 49
3.18. Az görülen ekleri içeren durumların ihmal edildiği sözcük türetme sonuçları. . 51

3.19. Doğru ve yanlış olarak üretilen sözcüklerden örnekler..... 52

4.1. Rasooli'nin [2] ve bizim geliştirdiğimiz modellerin sözcük türetme sonuçları . 53

4.2. İkinci karşılaştırma için kullanılan veri kümesinden bir kesit. Satır başlarındaki sayılar frekansları göstermektedir. Türkçe karakterler büyük harf eşlenikleriyle (ç -> C) değiştirilmiştir..... 54

4.3. Yapılan ikinci karşılaştırma sonuçları. Her iki model de morfolojik çözümleme için Morfessor CAT-MAP'i kullanmıştır..... 55

ŞEKİLLER

Sayfa

1.1. Morfem tabanlı morfolojide dili oluşturan unsurların, küçükten büyüğe sıralanışı 3
1.2. Geliştirilen modelin genel bir gösterimi 5

2.1. Dilin seviyeleri 7
2.2. Kuralların FST ile ifade edilmesi[3] 12
2.3. Paralel olarak derlenen FST'ler ile iki seviyeli morfoloji mimarisini basit bir gösterimi 13
2.4. "uygarlaştıramadıklarımızdanmışsınızcasına" sözcüğü için, morfolojik çözümleme ve üretme 14
2.5. Sabit ek (fixed affix) modeli için FST. **<epsilon>** ek olmadığı durumlarda kullanılır [2] 19
2.6. İkili ek (bigram affix) modeli için FST. **<epsilon>** ek olmadığı durumlarda kullanılır [2] 19

3.1. Hiyerarşik yığınsal kümeleme algoritması [4] 25
3.2. Word2vec modelinde kullanılan mimariler [5] 30
3.3. Örnek bir sonlu durum özdeviniri. S_i bir sözcük kökü kategorisini temsil ederken, M_i, M_j, M_k, M_l ve M_n bir ek kategorisini temsil etmektedir. 43
3.4. İsimler için oluşturulmuş sonlu durum özdevinirinden bir kesit. Burada *a* ve *e*, *A* olarak ve *ı* ve *i* ise *I* olarak temsil edilmiştir 44
3.5. Eylemler için oluşturulmuş sonlu durum özdevinirinden bir kesit. Burada *a* ve *e*, *A* olarak ve *ı* ve *i* ise *I* olarak temsil edilmiştir 45
3.6. Şekil 3.4.'deki sonlu durum özdevinirinden, *miras* sözcüğü için üretilen sözcüklerden bir parça. 46
3.7. Hatalı olarak oluşturulmuş sonlu durum özdevinirinden bir kesit. Burada *a* ve *e*, *A* olarak ve *ı* ve *i* ise *I* olarak temsil edilmiştir 50

SİMGE VE KISALTMALAR

NLP	Doğal dil işleme (Natural language processing)
CL	Hesaplamalı dil bilimi (Computational linguistics)
FSA	Sonlu durum özdeviniri (Finite state automata)
FST	Sonlu durum dönüştürücü (Finite state transducer)
AI	Yapay zeka (Artificial intelligence)
GNP	Cinsiyet, yaş ve şahıs (Gender, number, person)
NLG	Doğal dil üretme (Natural language generation)
ASR	Otomatik konuşma tanıma (Automatic speech recognition)
MT	Makine çevirisi (Machine translation)
SMT	İstatistiksel makine çevirisi (Statitical machine translation)
RBMT	Kural tabanlı makine çevirisi (Rule based machine translation)
LBM	Sözlükbirim tabanlı morfoloji (Lexeme based morphology)
WBM	Sözcük tabanlı morfoloji (Word based morphology)
KL	Kullback-Liebler (ıraksama)
JS	Jensen–Shannon (ıraksama)
CBOW	Devamlı sözcük torbası (Continuous bag-of-words)
MI	Karşılıklı bilgi (Mutual information)

1. GİRİŞ

Morfoloji (morphology) sözcüklerin yapısını inceleyen dilbilimi alanıdır [6]. Morfolojik türetme, yüzeysel formdaki sözcüklerin üretilmesi işlemidir. Morfolojik türetme, birçok doğal dil işleme (NLP - natural language processing) uygulamasında önemli bir yer tutmaktadır. Otomatik konuşma tanıma (ASR - automatic speech recognition) ve makine çevirisi (MT - machine translation) gibi dilin üretilmesini içeren uygulamalarda, morfolojik üretmeye ihtiyaç duyulur.

ASR, bilgisayar tabanlı, konuşulan dilin gerçek zamanlı yazılı metne dönüştürülmesini sağlayan uygulamaları geliştirmeyi amaçlar [7]. Konuşulan dilin analiz edilmesinden sonra yazıya dönüştürülmesi aşamasında morfolojik üretme çalışması yapılacaktır. Bir doğal dilden başka bir dile çeviri yapan bilgisayar yazılımlarını geliştirmeyi amaçlayan makine çevirisi, hesaplamalı dilbilimi ve doğal dil işleme alanlarının bir alt alanıdır. Çevirisi yapılacak dilin morfolojik olarak türetilmesi gerekmektedir. Hem kural tabanlı, hem istatistiksel hem de hibrid MT sistemleri dilin çözümlenme ve üretilme aşamalarını içerir.

Günümüzde dil üretme çalışmaları, işaretlenmiş geniş veri kümeleri kullanarak, dilleri modellemeyi amaçlar. ASR'de, eşleştirilmiş ses kayıtları ve bunlara ait metinler eğitim verisi olarak kullanılır. Benzer bir şekilde, makine çevirisinde de kaynak dilden hedef dile tercüme edilmiş ve eşleştirilmiş metin ikilileri (bilingual text) kullanılarak sistemin dili öğrenmesi, modellemesi sağlanır. Banko ve Brill [8], kullanılan daha büyük veri kümesinin bu gibi uygulamaların başarısını artıracağını ortaya koymuştur.

İşaretlenmiş veri kullanılması ve bu verinin boyutunun artırılması gelişen internet ile birlikte kolaylaşmıştır. Özellikle İngilizce gibi dillerde, işaratlenmiş veri kümelerine bile doğrudan, kolayca erişim sağlanabilmektedir [2]. Denetimli (supervised) NLP uygulamalarında, ihtiyaç duyulan işaretlenmiş veri miktarı fazladır. Özellikle kaynak sayısı az olan dillerde, kullanılabilecek işaretlenmiş veri miktarı kısıtlı olabilmekte ve bu da ilgili NLP uygulamasının başarısını etkileyebilmektedir. Yeterli sayıda işaretlenmiş veri bulunmayan diller için sonradan işaretleme yapılması da ciddi bir iş gücü gerektirmektedir.

Türkçe, Fince gibi karmaşık morfolojiye sahip diller için geliştirilen denetimli morfoloji öğrenme uygulamalarında sözcüklerin doğru analizleri yanında, birçok kural da sisteme elle

dahil edilebilmektedir. Türkçe, düzenli bir morfolojiye sahip olmasına rağmen, bütün yazımsal (orthographic) ve biçim dizgesi (morphotactics) kuralları, eklerin alomorfik özellikleriyle birlikte kullanılmalıdır. Buna ek olarak, dillerin dinamik (zamanla değişebilme) yapısı gereği, işaretlenmiş veri kümeleri, sık sık yeniden gözden geçirilmeye ihtiyaç duyarlar.

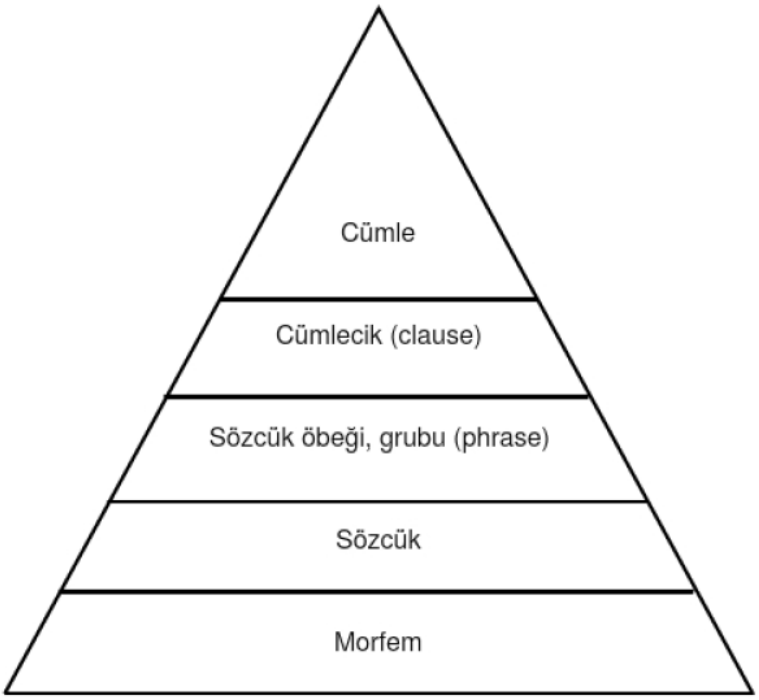
Kaynak sayısının az olması, dillerin karmaşık morfolojik yapıları ve dillerin dinamik olması gibi nedenler denetimsiz (unsupervised) NLP çalışmalarına olan ilgiyi artırmıştır. Denetimsiz öğrenme, işaretlenmiş veri kullanmadan, çoğunlukla, istatistiksel ve olasılıksal hesaplamalarla öğrenmeyi gerçekleştirir. Örneğin MT açısında bakılırsa, çeviri ikili metin kullanılmadan dil modellenerek morfolojik çözümleme ve üretme çalışmaları gerçekleştirilir. Birçok doğal dil işleme ve hesaplamalı dilbilimi alanında denetimsiz öğrenme uygulanmıştır [2, 9–13].

Yukarıda anlatılan nedenlerden dolayı, bu çalışmada denetimsiz olarak sözcük türetme modeli geliştirilmiştir. Çalışmada, eklerine ayrıştırılmış sözcükler dışında hiçbir işaretlenmiş veri kullanılmamıştır. Bu ayrıştırma işleminden sonra elde edilen morfemler (morpheme) hakkında hiçbir tür ve alomorfik yapı bilgisi bilinmemektedir. Modelin, Türkçenin yazımsal ve biçim dizgesi kurallarını kendisinin keşfetmesi amaçlanmıştır.

Morfoloji terimi ilk olarak 1859 yılında, Alman dilbilimci August Schleicher tarafından ortaya atılmıştır [14]. Morfoloji, sözcüklerin iç yapısını, sözcüklerin alt birimlerinin nasıl bir araya geldiğini inceler. Morfoloji hem hesaplamalı dilbilim hem de doğal dil işleme alanında önemli bir yer tutar. Dilin üretilmesi ve anlaşılması aşamasında hesaplamalı morfoloji hem dilbilimciler hem de bilgisayar bilimciler tarafından kullanılmaktadır [15].

Morfem tabanlı morfolojide, sözcüklerin anlam içeren, en küçük yapı birimi morfemlerdir [16, 17]. Morfemlerin anlam içermesi, sözcük olarak tek başına anlam ifade edebilmesi veya dilbilgisi açısından görev olarak bir anlam ifade edebilmesi olarak açıklanır. Sözcük içerisindeki ek(ler) ile sözcük kökü birer morfemdir. Örneğin *kitaplarım* sözcüğündeki *kitap* sözcük kökü ile *lar* ve *ım* ekleri birer morfemdirler. *Kitap* sözcüğü tek başına anlamlıyken, *lar* morfemi eklendiği sözcüğe çoğul anlamı katar.

Sözcükler morfemlerin kurallı bir şekilde bir araya gelmesiyle oluşur. Sözcük öbekleri de sözcüklerin bir araya gelmesiyle meydana gelirler. Dili oluşturan unsurların sıralaması Şekil 1.1.'de gösterilmektedir.



ŞEKIL 1.1.: Morfem tabanlı morfolojide dili oluşturan unsurların, küçükten büyüğe sıralanışı

Sözlükbirim tabanlı morfoloji ise (LBM - lexeme based morphology), morfem tabanlı morfolojiden farklı olarak sadece sözlükbirimler ve serbest morfemler en küçük dilbilgisel birimlerdir. Çekimlenen sözcükler bir sözcükbirime ait sözcük formları olarak adlandırılır [18]. Örneğin *ipekler*, *ipekten*, *ipeği* sözcükleri *ipek* sözlükbirimine ait sözcük formları olarak ifade edilir.

Sözcük tabanlı morfolojide (WBM - word based morphology), sözcük ve paradigmaları kullanır. Yeni oluşturulacak bir sözcük, daha önce üretilmiş bir sözcüğün kurallarının uygulanmasıyla oluşturulur. Üretilen yeni sözcük bir sözlüksel kategori içerisinde yer alır [19]. Sözcük tabanlı morfoloji, morfem tabanlı yaklaşımdan farklı olarak morfemlerin bir araya geldiği kuralları oluşturmak yerine, çekimsel paradigmalar arasındaki ilişkiyi genelleştirmeye çalışır.

Bu çalışmada morfoloji, morfem tabanlı yaklaşım ile ele alınacaktır. Ekler sözcük sonlarına eklenerek, birden fazla ek içeren çekimsel sözcük formları oluşturabilirler. Bu durum sözcükleri LBM ile ele almayı zorlaştırır. İngilizce gibi ek sayısı sınırlı olan dillerde, sözcüklerin çekimsel formları da sınırlı olacaktır. Fakat Türkçede, bir sözcüğün çekimsel formları için

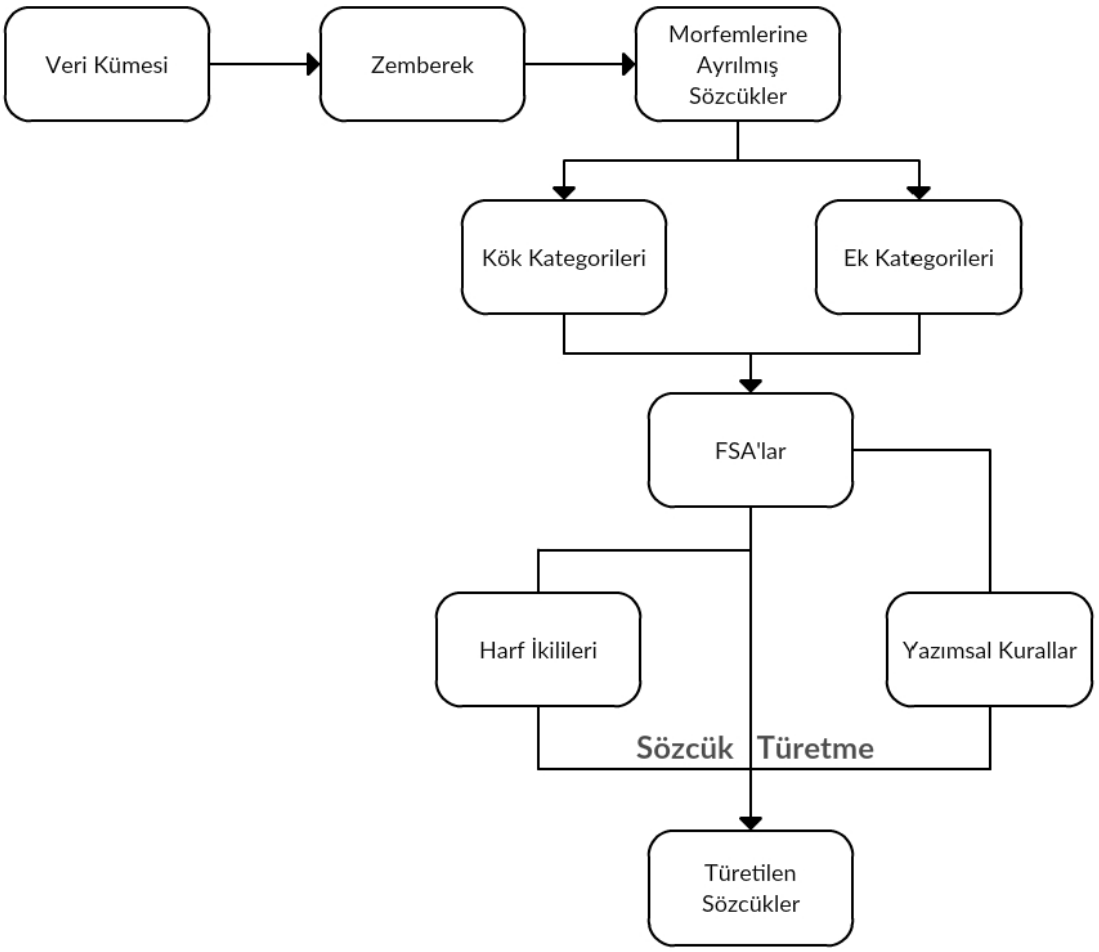
teorik bir sınır yoktur [20]. Benzer olarak, Türkçenin karmaşık morfolojisi, sözcük paradigmalarının genelleştirilmesini zorlaştırır. Bu durum da dil üretme uygulamalarını WBM yaklaşımı ile gerçekleştirmeyi zorlaştırır. Türkçenin çok sayıda ek içeren morfolojisi, yazımsal (orthographic) ve biçim dizgesi (morphotactics) kurallarının çeşitliğini, morfolojik üretme çalışmalarının morfem tabanlı morfoloji ile ele alınmasını kolaylaştırır.

Çalışmanın motivasyonu benzer ekleri alan sözcük köklerinin benzer olabileceği ve benzer ekler almaya devam edebileceğidir. Bu öngörü ile ilgili çalışmalar ve çıkarımlar Model bölümünde ayrıntılı bir şekilde anlatılmıştır.

Bu tez çalışmasında sözcükleri morfemlerine ayrıştırmak için Zemberek [21] kullanılmıştır. Zemberek, açık kaynak kodlu, Türkçe doğal dil işleme kütüphanesidir. Java programlama dili ile kodlanmış olan Zemberek, morfolojik analiz dışında yazım denetimi, hatalı sözcükler için öneri, heceleme gibi işlevleri de yerine getirebilmektedir. Çalışmada sözcüklerin morfemlerine ayrıştırılması işlemi Zemberek tarafından denetimli olarak gerçekleştirilmiştir. Zemberek dışında, denetimli veya denetimsiz, başka bir morfolojik çözümleyici de kullanılabilir. Çalışmanın diğer adımları morfemlerine ayrılmış sözcükleri kullanacağı için diğer adımların her biri morfolojik analiz yapan sistemin başarı oranından etkilenecektir. Özetle çalışmanın devamında verilen tüm doğruluk ve saflık oranları gibi değerler, Zemberek'in morfolojik çözümleme başarısından etkilenmiştir ve sonuçlarda Zemberek-göreli doğruluk oranları verilmiştir.

Bu çalışmada, Türkçenin morfem tabanlı morfolojisi denetimsiz olarak öğrenilmeye çalışılmıştır. İşaretlenmemiş bir metinde yer alan sözcükler, Zemberek [21] yardımı ile eklerine ayrıştırılacaktır. Daha sonra öğrenilen morfoloji kullanılarak, bir morfolojik üretme çalışması yapılacak ve yeni sözcükler türetilecektir. Bu türetme aşamasından önce, sözcük kökleri tür (isim, fiil) özelliklerine göre kategorilere ayrılacaktır. Daha sonra benzer olarak ekler, alomorfik özelliklerine göre kategorilerde toplanacaktır. Denetimsiz olarak gerçekleştirilen tüm bu işlemlerden sonra Türkçenin yazımsal ve biçim dizgesi kuralları keşfedilmeye çalışılacaktır. En son olarak bu bilgilerle, sonlu durumlu özdevinirler kullanılarak sözcük türetme işlemi gerçekleştirilecektir. Modelin genel bir gösterimi Şekil 1.2.'de görülmektedir.

Tez metninin ana hatlarıyla yapılanması şu şekildedir: İkinci bölümde, tez için ihtiyaç duyulan alan bilgisine ve konuyla ilgili geçmişte yapılan çalışmalara yer verilmiştir. Üçüncü



ŞEKIL 1.2.: Geliştirilen modelin genel bir gösterimi

bölümde sözcük türetme modeli anlatılmıştır. Model anlatılırken sırasıyla ilk olarak ön çalışma ve motivasyon verilmiş ve daha sonra sözcük köklerinin ve eklerin kategorilerinin bulunması çalışmalarına yer verilmiştir. Ardından sözcük türetme çalışmaları için gerekli olan harf ikilileri ve yazımsal kuralların keşfedilmesi çalışmalarına değinilmiştir. Modelin son bölümünde sözcük türetme çalışmaları ve bu çalışmalara ait sonuçlar verilmiştir. Tezin dördüncü bölümünde çalışmamız başka bir çalışma ile karşılaştırılmıştır. Tez metninin beşinci ve son bölümünde sonuçlar ve tartışma sunulmuştur.

2. ALAN BİLGİSİ VE ALANYAZIN

2.1. Doğal Dil İşleme

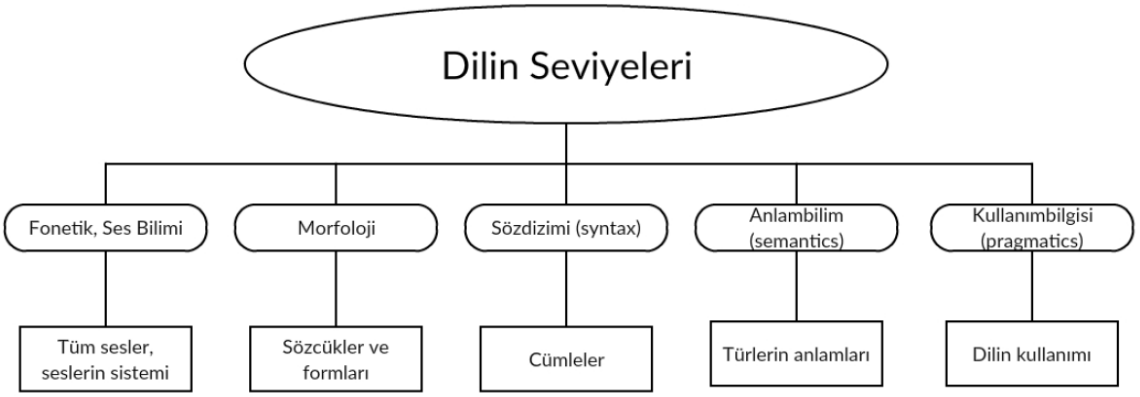
Bir dil, her biri sonlu uzunlukta ve sonlu bir üyeler kümesinde oluşturulan (sonlu ya da sonsuz) cümleler kümesidir [22]. Doğal dil, insan zekası tarafından, önceden tasarlanmadan, doğal bir şekilde gelişen ve kullanılan dillerdir. Genellikle konuşma, yazı ve işaret aracılığıyla iletişim için kullanılır [23]. İngilizce, Türkçe gibi günlük hayatta kullanılan diller, doğal dillere örnektir.

Doğal dil işleme, insan dillerini (doğal diller) analiz eden, onları anlamaya çalışan veya onları üretmeye yarayan bilgisayar ve hesaplamalı dilbilimi alanıdır [24]. Doğal dil işleme, doğal dillerin kurallı yapısını çözümleyerek dilin yeniden üretilmesini, anlaşılmasını veya yorumlanmasını sağlar. Doğal dil uygulama alanlarının bir kısmı şu şekilde özetlenebilir:

- Metin özetleme (Automatic text summarization)
- Makine çevirisi (Machine translation)
- Soru cevaplama (Question answering)
- Konuşma tanıma (Speech recognition)
- Döküman özetleme / kümeleme (document summarization / clustering)
- Bilgi edinme (Information retrieval)

Hesaplamalı dilbilimi, (CL - computational linguistics), dilbilimi işlemlerinin hesaplamalı bir bakış açısıyla ele alınmasıdır. CL, dilin edinimini, üretilmesini ve anlaşılmasını inceleyen bilimsel disiplindir. Tarihte, dilbilimcileri tarafından ele alınan bu alan, daha çok bilgisayar bilimcileri tarafından ele alınan NLP ile farklı alanlardı. Bilgisayar ve İnternet kullanımının artmasıyla beraber bu iki alan birleşmeye ve ortak konularla ilgilenmeye başladılar. Dillerin ele alınma seviyeleri Şekil 2.1.'de gösterilmektedir. Hesaplamalı dilbiliminin ele aldığı problemlerin bir kısmı aşağıdadır:

- Morfolojik bölümleme (Morphological segmentation)



ŞEKIL 2.1.: Dilin seviyeleri

- Morfolojik türetme (Morphological generation)
- Tür etiketleme (Part-of-speech tagging - POST)

Çok daha eski bir tarihe sahip olan doğal dil işleme araştırmaları, günümüzdeki anlamıyla 1950'lerde başlamıştır [25]. 1950 yılında Alan Turing, yayınladığı çalışmada makinelerin düşünebileceğini ve insanları taklit edebileceğini göstermeye çalışmıştır. Turing testi olarak kullanılmaya devam eden yöntem ile makinelerin zekası ölçülmeye çalışılmaktadır [26]. Doğal dil işleme çalışmaları ilk olarak makine çevirisi üzerine odaklanmıştı. 1954 yılında, sınırlı sayıdaki Rusça cümle, İngilizce'ye otomatik olarak çevrildi. Daha sonraki yıllarda yapay zeka (AI - artificial intelligence) tabanlı NLP uygulamaları kullanılmaya başlandı [25]. 1990'lı yıllarla beraber istatistiksel doğal dil işleme çalışmaları popüler olmaya başladı [27].

Günümüzde ise yukarıdaki alanlarda ve çok daha fazlasında NLP çalışmaları devam etmektedir. İnternetin gelişmesi ve doğru orantılı olarak erişilebilen veri sayısının artmasıyla birlikte denetimsiz (unsupervised) doğal dil işleme araştırmaları, bu alandaki, son yıllardaki en çekici başlıklardan biri haline gelmiştir [10]. İşaretlenmemiş veriler kullanılarak, makine öğrenmesi ve istatistiksel yaklaşımlar gibi yöntemlerle doğal dil işleme uygulamaları geliştirilebilmektedir.

2.2. Morfoloji

Morfoloji, sözcüklerin iç yapısını ve bu yapının kurallı oluşumunu inceler. Morfemlerin kurallı birliktelikleriyle sözcükler meydana gelir. Sözcükler, bir dili oluşturan anlamlı en küçük

birimlerdir. Bir sözcük, dil içerisindeki tek başına anlam ifade edebilen, cümlelerin yapı taşı olarak kabul edilebilir [28]. Fakat bazı sözcükler tek başlarına anlam ifade etmelerine karşın tek başlarına kullanılmazlar (örneğin Türkçede *ve*, *veya* sözcükleri ile İngilizcedeki *the*, *of* sözcükleri) [29]. Sözcükler, yazıda, dile özgü alfabe içerisinde yer alan çeşitli karakterlerle ifade edilir. Bu ifadelerin sesler (fonem) ile ifade edilmesiyle de konuşmaya aktarılırlar.

Morfemler, serbest morfem ve bağımlı (bound) morfemler olarak ikiye ayrılır. Serbest morfemler tek başlarına bir sözcük olabilirler. Bağımlı morfemler ise tek başlarına kullanılamazlar. Sözcük kökleri (stem) serbest veya bağımlı morfem olabilir. Çoğunlukla, sözcük kökleri serbest morfemlerdir. Fakat İngilizcede, aşağıdaki gibi bazı sözcük kökleri bağımlı morfemlerdir [30]:

receive (teslim almak) $->$ ceive
cranberry (kızılcık) $->$ cran
lukewarm (ılık, ilgisiz) $->$ luke
inept (beceriksiz) $->$ ept

Ekler ise bağımlı morfemlerdir. Sözcüklere eklenerek kullanılabilirler. Fakat Arapça gibi dillerde ekler tek başlarına bulunabilirler. Türkçede ise, bir sözcük içerisinde en az bir kök bulunur ve bu sözcük kökünün alabileceği ek sayısında teorik olarak bir sınır yoktur.

Ekler, sözcük içerisindeki bulundukları yerlere göre ön ek (prefix), iç ek (infix) ve son ek (suffix) olarak adlandırılır. Sondan eklemeli bir dil olan Türkçede eklerin büyük çoğunluğu son ektir. *Çiçeklerden* sözcüğü *çiçek* sözcük köküne, *ler* ve *den* son eklerinin eklenmesiyle oluşmuştur. İngilizcede, *sleeping (uyku)* sözcüğündeki *ing* eki, *sleeper (uykucu)* sözcüğündeki *er* eki son eklere örnektir. Türkçede daha çok yabancı dillerden geçmiş ön ekler mevcuttur [31]. *Namert* sözcüğündeki *na* olumsuzluk eki Türkçedeki ön eklere örnek olarak gösterilebilir. İngizlizcedeki *replay* sözcüğündeki *re* tekrar anlamı taşıyan ön ek için bir örnektir. Türkçe ve İngilizcede standard bir iç ek kullanımı mevcut değildir. Arapçadaki *ktb (yazma ile ilgili)* kök morfeminden türetilmiş *katab (o yazdı)* sözcüğündeki *a-a* ve *kaAtib (yazar)* sözcüğündeki *aA-i* iç eklere örnektir [32].

Görevleri açısından ekler ikiye ayrılır: Çekim ekleri ve yapım ekleri. Morfoloji üç farklı şekilde ele alınır: Çekimsel (inflectional) morfoloji, türetimsel (derivational) ve bileşik (compounding) morfoloji [33]. Çekimsel morfolojide ekler, eklendiği sözcüğün türünü değiştirmezler, yeni bir sözcük türetmezler. Çekim ekleri, yeni üretilen sözcüğün cinsiyet, yaş ve

şahıs (GNP - gender, number, person) gibi dilbilgisel özelliklerini değiştirir [1]. Örneğin *koşuyorlarmış* sözcüğü içerisinde yer alan *uyor*, *lar* ve *mış* ekleri çekim ekleridir. Türetimsel morfolojide ise, yapım ekleri eklendiği sözcüğün anlamını ve/veya türünü (part of speech) değiştiriler. Örneğin *tuz* sözcük köküne, *luk* eki eklenerek yeni anlamda bir sözcük (*tuzluk*) türetilmiştir. *Dur* (türü fiil) sözcük köküne ise *gun* eki eklendiğinde, oluşan *durgun* (türü isim) sözcüğünde hem anlam hem de tür değişimi olmuştur. Birleşimsel morfolojide ise sözcük içerisinde birden fazla kök bulunur. Türkçedeki *sivrisinek (sivri ve sinek kökleri)* ve *Anıtkabir (anıt ve kabir kökleri)* birleşimsel morfolojideki birleşik sözcüklere örnektir. İngilizcede ise *database (veritabanı)* sözcüğü birleşik bir sözcüktür. Birleşik sözcükler içinde yer alan sözcük kökleri ek alabilir. *Demiryolu* sözcüğünde *yol* sözcük kökü *u* çekim ekini alarak birleşik sözcük içerisinde yer almıştır.

Alomorflar

Morfemler görünüm olarak farklı olsalar da sözcük içerisinde aynı görevde olabilirler. Örneğin çoğul ekleri Türkçede *lar* ve *ler*, İngilizcede ise *s* ve *ies* şeklinde görünümlere sahiptirler. Bu morfemlere alomorf (allomorph) denir. Alomorflar, bazı dillerde, ünlü uyumu gibi kurallardan ötürü sıkça görülmektedir. Ünlü uyumu sözcük içerisindeki ünlü harflerin birbirlerine adapte olma durumudur.

Türkçede ünlü uyumu önemli bir yer tutmaktadır. Bu sebepten ötürü dil içerisinde birçok alomorf oluşmuştur. Örneğin, sözcüğün bulunma durumu (ismin -i hali), ünlü uyumundan dolayı birçok farklı şekilde yer alabilir: [*ı*], [*i*], [*u*], [*ü*].

ı *baraj* - *baraj-ı*
i *kent* - *kent-i*
u *koyun* - *koyun-u*
ü *üzüm* - *üzüm-ü*

Türkçede alomorfların oluşmasına sebep olan tek durum ünlü uyumu değildir. Sözcük sonunda bulunan sert (unvoiced) ünsüzler, bir sonraki ekin ilk harfine göre değişime uğrarlar. Bu duruma ünsüz değişmesi denilmektedir. Sözcük sonunda yer alan sert ünsüzler (örneğin *p*, *ç*, *t*, *k*), bir sonraki ekin ünlü bir harf olması durumunda değişime (yumuşama) uğrayarak yumuşak ünsüz harflere dönüşürler (örneğin p:*b*, ç:*c*, t:*d*, k:*ğ*):

- Belirtme durumu: *kitap-lı$\underline{k}$-ı* $->$ *kitap-lı$\underline{ğ}$-ı*
- Belirtme durumu: *ağa$\underline{ç}$-ı* $->$ *ağa$\underline{c}$-ı*
- Yönelme durumu: *şara$\underline{p}$-a* $->$ *şara$\underline{b}$-a*
- Yönelme durumu: *kağı$\underline{t}$-a* $->$ *kağı$\underline{d}$-a*

Buna ek olarak, Türkçede, ünsüz uyumu ile sözcük sonundaki sert ünsüz ile takip eden ekteki ilk ünsüz harf arasında bir uyuşma meydana gelir. Örneğin, *para-dır* sözcüğünde *dır* morfemi, *kağıt-tır* sözcüğündeki *tır* morfemine dönüşür. Bu sebepten ötürü *dır* ve *tır* birbirlerinin alomorfudur.

Ünlü uyumu, ünsüz uyumu ve ünsüz değişmesiden dolayı Türkçe, birçok alomorf örneğini bünyesinde barındırır. Bazı alomorflar çok farklı görünümlerde olabilir (örneğin *dık* ve *tüğ*). Alomorfların incelenmesi ve sınıflandırılması birçok NLP uygulamasında kullanılabilir. Özellikle olasılıksal modellerde bu alomorfların tek bir morfem türüne karşılık geldiği bilgisi önemli bir bilgidir. Morfemlerin genelleştirilebilmesi için, bazı Türkçe morfolojik çözümleyiciler, alomorflar için büyük harf gösterimi kullanmıştır [34]. Örneğin *A*, *a* ve *e* harflerini ortak olarak göstermek için kullanılabilir. Burada *a* ve *e* harfleri birbirlerinin allofanlarıdır. Allofan, fonetik olarak benzer ses kesişim dizisini içeren bir ses birim setidir. Allofanlar fonetik olarak benzer olan ses birimleridir [35]. Bu gösterimle çoğul ekleri *lar* ve *ler* ortak bir görünüme (*lAr*) sahip olurlar.

2.3. Morfolojik Çözümleme

Morfolojik çözümleme (veya morfolojik analiz), bir sözüğün kök ve eklerinin tanımlanması işidir. Morfolojik çözümleme ile sözcüğün iç yapısı anlaşılmaya çalışılır. Sözcüğün anlamsal ve sözdizimsel rolü keşfedilir [6]. Morfolojik çözümleme, sözcüğün morfemlere ayrıştırılması (morphological segmentation) ve morfemlerin türlerinin belirlenmesi işlerini içerir. Bu sayede, dilin biçim dizgesi kuralları (morphotactics) da belirlenir. Biçim dizgesi kuralları, morfemlerin sözcük içerisinde hangi sıra ile bulunabileceği, hangi morfem sınıfının hangi morfem sınıfı tarafından takip edilebileceğini açıklar [36].

Yüzeysel ve Sözcüksel Form

Yüzeysel seviye (surface level), sözcüğün gerçek yazımsal şeklidir. Birkaç Türkçe ve İngilizce sözcüğün yüzeysel formları:

- geliyorum, kitabım, runs (koşar), dogs (köpekler)

Sözlüksel seviye (lexical level) ise sözcüğün işlevsel parçalarının yapısal gösterimidir [36]. Yukarıdaki sözcüklerin sözlüksel gösterimleri:

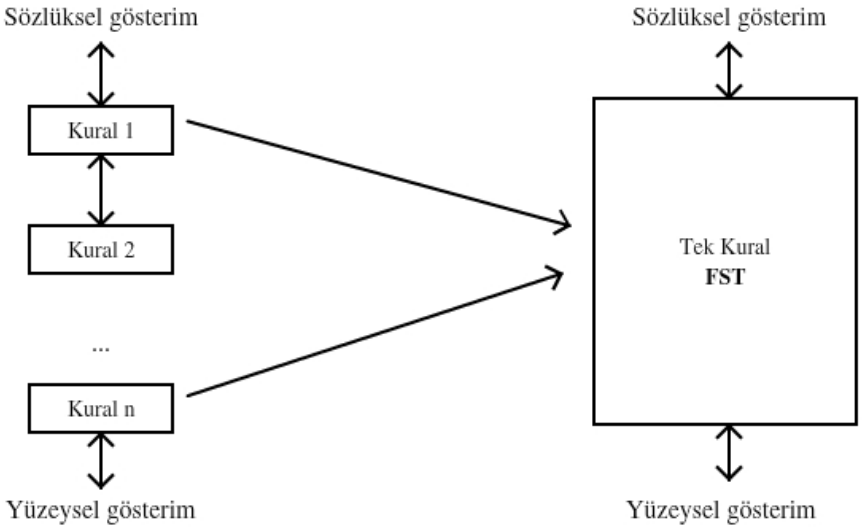
- gel + PROG + 1SG (PROG= şimdiki zaman, 1SG= birinci tekil şahıs)
- kitap + P1SG (P1SG= birinci tekil şahıs sahiplik)
- run + AOR (AOR= geniş zaman)
- dog + PLU (PLU= çoğul)

Yüzeysel formdan sözlüksel forma dönüşüm morfolojik tanıma (morphological recognition) ve sözlüksel formdan yüzeysel formun elde edilmesi de morfolojik türetme olarak adlandırılır.

İki Seviyeli Morfoloji

İki seviyeli morfoloji (two-level morphology), sözlüksel ve yüzeysel seviye arasındaki bağlantıyı ifade eder. İlk olarak 1983 yılında Kimmo Koskenniemi tarafından tanıtılmıştır [37].

Koskenniemi, Kaplan ve Kay'ın [38] çalışmasını geliştirerek iki seviyeli modeli tanımlamıştır [3]. Kaplan ve Kay, sözlüksel form ile yüzeysel form arasındaki dönüşüm kurallarının tek bir sonlu durum özdeviniri (veya dönüştürücü) ile ifade edilebileceğini ortaya koymuştur (bkz. Şekil 2.2.). Basamaklandırılmış kuralların tek bir sonlu özdevinir dönüştürücü (FST - finite state transducer) ile ifade edilmesi, daha etkin ve uygulanabilirdir. Ayrıca FST kullanımı ile çift yönlü dönüşüm yapılabilmektedir. Fakat burada, birleştirilen kuralların boyutu fazla olabilir. Özellikle zengin ve karışık morfolojilere sahip, Türkçe, Fince gibi dillerde birleştirilen kuralların boyutu oldukça büyük olabilir [37].



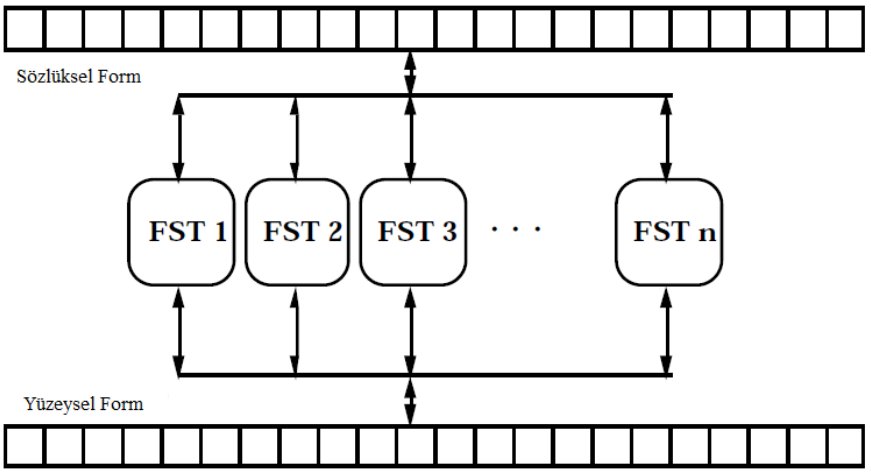
ŞEKIL 2.2.: Kuralların FST ile ifade edilmesi[3]

Koskenniemi, tek bir FST yerine, kuralları paralel olarak olarak işleten, iki seviyeli birden fazla FST önermiştir (bkz. Şekil 2.3.).

Koskenniemi'nin kısıt tabanlı kuralları şu şekilde ifade edilir:

- s:y $\Rightarrow$ LC _ RC. Bağlam kısıt kuralı. Sol bağlam (LC - left context) ve sağ bağlam (RC - right context) verildiğinde, sözlüksel form *s*, yüzeysel form *y* olarak ifade edilir. Ama her zaman bu durum gerçekleşmek durumunda değildir.
- s:y $\Leftarrow$ LC _ RC. Yüzeysel zorlama kuralı. Bu bağlamda *s* sözlüksel formu *y* yüzeysel formu olarka ifade edilir. Fakat sadece bu bağlam içerisinde gerçekleşmek zorunda değildir.
- s:y $\Leftrightarrow$ LC _ RC. Birleşik kural. *s* sözlüksel formu sadece bu bağlam içerisinde *y* yüzeysel formuna karşılık gelir.
- s:y / $\Leftarrow$ LC _ RC. Dışlanım kuralı. Verilen bağlam içerisinde, *s* sözlüksel formu hiçbir zaman *y* yüzeysel formuna karşılık gelmez.

Bu kurallar sonlu durum dönüştürücüleriyle paralel olarak derlenir. Bu FST'ler sözlüksel form ile yüzeysel form arasındaki ilişkiyi konrol eder. Herhangi bir FST tarafından reddedilen ilişki, geçersiz bir ilişki olarak kabul edilir [36]. Paralel olarak derlenen bu model Şekil 2.3.'te gösterilmektedir.



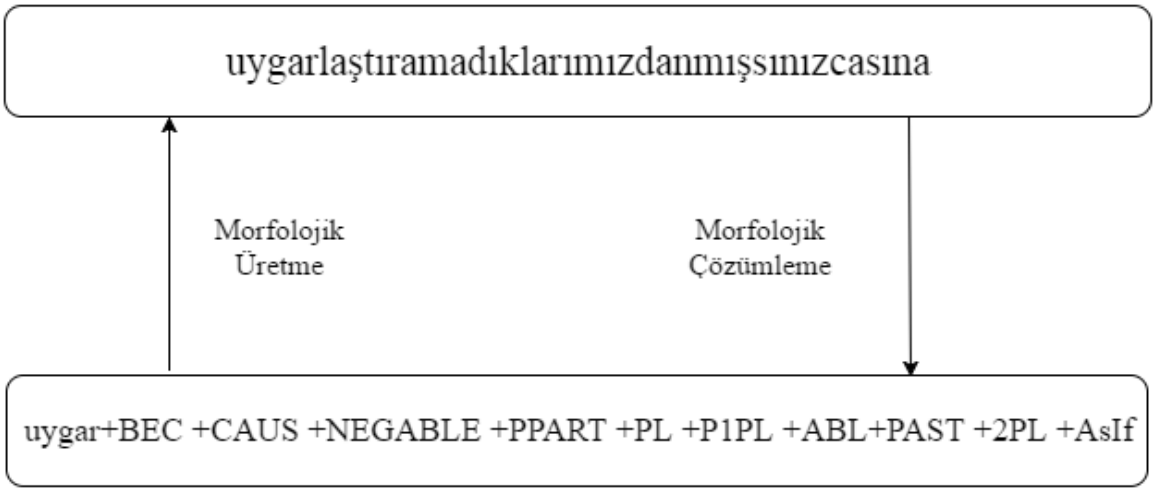
ŞEKIL 2.3.: Paralel olarak derlenen FST'ler ile iki seviyeli morfoloji mimarisini basit bir gösterimi

İki seviyeli morfoloji modeli, birçok farklı dilin çözümlenmesi için kullanılmıştır [39–43]. Oflazer [36], çalışmasında, Türkçenin iki seviyeli modelini açıklamıştır. Bu çalışmada, Türkçedeki ünlü ve ünsüz harf grupları (fonetik özelliklerine göre), iki seviyeli kurallar ve sözcük kök türleri ayrıntılı olarak anlatılmıştır. İsimsel sözcük kökleri ve fiilsel sözcük köklerinin biçim dizgesi kuralları için ayrı ayrı sonlu durum makineleri tanımlanmıştır.

Hankamer [44], çalışmasında benzer olarak sonlu durum yaklaşımını kullanmıştır. *Keçi* ismini verdiği çalışmasında morfolojik ve fonolojik kuralları ayrı ayrı ele almaya çalışmıştır. FST'deki kök durumlar, bir sözcük kategoriyle ifade edilmiştir. Tanımlanan isim, fiil, sıfat ve benzer kategorilerle, hangi ek sınıfının ekleneceği belirlenmiştir. Bu şekilde bir kategoriye eklenen ekten sonra yeni bir kategoriye geçilir.

2.4. Morfolojik Türetme

Doğal dil üretme (NLG - natural language generation), bir doğal dilin bilgisayar tabanlı gösteriminden, yüzeysel formdaki gösterimin elde edilmesi işlemidir. Birçok NLG sistemi, morfolojik türetme ve sözdizimsel doğrusallaştırma (syntactic linearization) işlemlerini ayrı ayrı ele alır [45]. Sözdizimsel doğrusallaştırma veya sözdizimsel doğrulayıcı, üretilen metni dilbilgisi kuralları açısından (örneğin özne-yüklem uyumu) kontrol eder. Morfolojik türetme ise dili oluşturan sözcüklerin morfoloji kurallarına uygun olarak üretilmesidir. Birçok morfolojik türetme sistemi Koskenniemi'nin iki seviyeli morfolojisi üzerine geliştirilmiştir [46].



ŞEKIL 2.4.: "uygarlaştıramadıklarımızdanmışsınızcasına" sözcüğü için, morfolojik çözümleme ve üretme

Morfolojik türetme, morfolojik çözümlemenin tam tersidir (bkz. Şekil 2.4.). Morfolojik türetme, sözlüksel formdan, yüzeysel formdaki sözcüklerin türetilmesi işlemidir. Morfolojik çözümleme ve türetme birçok NLP uygulamasında kullanılmaktadır. Özelikkle makine çevirisinde çözümleme ve türetme ortak olarak kullanılmakdatırlar. Çevirisi yapılacak dilin çözümlemesi yapıldıktan sonra çevirilecek dilin üretilmesi yapılır. Makine çevirisi, bir dilin çözümlenerek diğer bir dilin üretilmesidir. İstatistiksel makine çevirisi (SMT - statitical machine translation) veya kural tabanlı makine çevirisi (RBMT - rule based machine translation) sistemlerinin tamamı çözümleme ve üretme işlemini gerçekleştirmektedir [47].

Oflazer [36] Türkçe için iki seviyeli morfoloji tanımlamış ve PC-KIMMO [48] platformunuyla uygulamıştır. Türkçe için biçim dizgesi kuralları ve yazımsal kurallar denetimli (supervised) olarak tanımlanmıştır. Haşim Sak [49], Türkçe için diğer NLP uygulamalarında kullanılabilecek bir takım kaynak ve araçlar sunmuştur. FST tabanlı morfolojik ayrıştırıcı (parser), morfolojik belirsizlik giderici (disambiguator) ve İnternet üzerinden erişilebilen geniş bir veri kümesi sunmuştur. Morfolojik ayrıştırıcı, çözümlediği sözcük için, anlamsal belirsizlikten dolayı birden fazla sonuç üretebilir. Örneğin *kalemleri* sözcüğü için oluşan farklı çözümlemeler aşağıdaki gibidir:

kalem[Noun]+lAr[A3pl]+SH[P3sg]+[Nom] (onun birden fazla kalemi)
kalem[Noun]+lAr[A3pl]+[Pnon]+YH[Acc] (o kalemleri, işaret)
kalem[Noun]+lAr[A3pl]+SH[P3pl]+[Nom] (onların birden fazla kalemleri)
kalem[Noun]+[A3sg]+lArH[P3pl]+[Nom] (onların kalemi)

Bu ve benzer belirsizlikleri ortadan kaldırmak için morfolojik belirsizlik giderici uygulamalar kullanılmaktadır. Haşim Sak [49] ve arkadaşları, bir ortalama algılayıcı (averaged perceptron) tabanlı morfolojik belirsizlik giderici geliştirmiş.

Her iki çalışmanın da (Oflazer [36], Sak [49]) asıl amaçları morfolojik çözümleme olmasına karşın, morfolojik türetme için de kullanılabilirler.

Arapça için birçok morfolojik türetme çalışması yapılmıştır. Arapça, diğer Sami (Semitic) diller gibi morfolojik açıdan karmaşık ve sondan eklemeli olmayan bir dildir [47]. Bu da morfolojik analiz ve çözümleme araştırmalarını ilgi çekici ve zorlu kılmıştır. Arapçada kökler tek başlarına bulunmaz. Sesli harfler (melodi seslileri) ile birlikte bir örüntü oluştururlar [50, 51]. Örneğin Arapça kök *katab* (o yazdı), *ktb* kök morfemi ile *a-a* melodi morfeminin birleşiminden oluşmaktadır. Burada oluşan örüntü CVCVCV (C=sessiz harf, V=sesli harf) şeklinde olmuştur [50].

Beesley [51], KIMMO benzeri iki seviyeli kuralları tanımlamıştır. FST tabanlı sistem, morfolojik çözümleme ve üretme için kullanılmaktadır. Kiraz [52], çift bantlı iki seviyeli morfoloji modelini geliştirmiştir. Bu morfolojik çözümleme modelinde klasik sonlu durum dönüştürücüleri, çift bantlı yardımcı versiyonları (AFST) ile değiştirilmişlerdir.

Cavalli [50], Arapçanın karmaşık morfolojisini basite indigeyerek ihtiyaç duyulan kural sayısını önemli miktarda azaltmıştır. Karmaşık morfolojiyi basitleştirebilmek için ayırt etme ağaçları (discrimination trees) ve düzenli ifadeler kullanılarak kök içerisindeki değişimlerle, diğer çekimleme kuralları birbirlerinden ayrı tutulmuştur. Cavalli ve arkadaşları, yaptıkları gözlemde, birkaç fiil türü dışında, kök içerisinde bu değişimlerle, sözcüklere ön ek ve son ek ekleme işlemlerinin birbirlerini çok az etkilediklerini gözlemlemişlerdir. Bu yüzden, bu iki işlemin birbirinden ayrı tutulabileceğini ortaya koymuşlardır. Arapça için birleşimsel bir yaklaşım kullanılarak morfolojik türetme gerçekleştirilmiştir.

Soudi ve Cavalli [53], önceki çalışmalarına [50] benzer olarak karmaşık Arapça morfolojisi basite indirgeyerek kural sayısını küçük bir sayıda tutmaya çalışmışlardır. Bu çalışmalarında, sözlükbirim (lexeme) tabanlı morfoloji modeli ile sözcük içerisindeki değişimleri, ek ekleme işlemlerinden ayrı tutmuşlardır. Soudi [53], karmaşık yapıdaki fiilsel kökleri morfolojik olarak form olarak kabul edip kök içerisindeki değişimleri diğer değişimlerden (ön ek ve son ek ekleme) ayırmışlardır.

Habash [54], çalışmasında Arapça için *Aragen* ismini verdiği büyük ölçekli, sözlükbirim tabanlı denetimli bir morfolojik türetme sistemi geliştirmiştir. Bu çalışmada, Buckwalter Arapça morfolojik çözümleyici sistemi [55] tersine çevrilerek morfolojik türetme için kullanmıştır. Buckwalter çözümleyicinin veri kümesi sözlükbirim ve bu sözükbirimlerin özelliklerini içeren özellik kümelerini kullanabilecek şekilde genişletilmiştir. Özellikler arasında numara, kişi ve durum çekimlemeleri mevcuttur.

Habash, Ranbow ve Kiraz [32], Arapçanın konuşulan lehçeleri için bir morfolojik çözümleyici ve üretici geliştirmişlerdir. Çalışmalarında lehçelerin standart Arapça ile olan ilişkileri ele alınmıştır. Lehçeler daha çok konuşma dilinde kullanıldıkları ve yazımsal olarak bazı farklılıklar görülebildiği için *MAGEAD* ismini verdikleri sistem hem yazımsal hemde sessel kuralları modellemeyi amaçlamıştır. Buna ek olarak konuşma sırasında insanların birden fazla lehçeyi ve standart Arapçayı birlikte kullandıkları göz önünde bulundurularak birden fazla morfolojik veriyi kullanmışlardır.

Shaalan ve ekibi [56], görev temelli interlingua tabanlı konuşma diyalogları için kural tabanlı bir Arapça morfolojik üretici geliştirmişleridir. Interlingua, Uluslararası Yapay Dil Derneği (IALA) tarafından geliştirilmiş, basit bir dilbilgisine sahip, farklı dilleri konuşan insanlar tarafından kolayca anlaşılabilen ve çoğunlukla biyolojik adlandırılmalarda kullanılan bir yapay dildir [57]. Shaalan, çalışmalarını, NESPOLE! [58] (NEgotiating through SPOken Language in E-commerce) isimli konuşmadan konuşmaya makine tercümesi yapan bir sistemin Arapça ayağını gerçekleştirebilmek için geliştirmişlerdir. NESPOLE, literal anlam yerine konuşmacının niyetini baz alan bir makine tercümesi sistemidir. Çalışmada Arapça sözcüklerin, NESPOLE'nin interlingua tabanlı gösterimi ile ifade edilmesi aşamsında karşılaşın sorunları ortaya koymuş ve çözüm önerilerini getirmişlerdir.

Selçuk Köprü [47], sonlu durum makinelerini, birleştirme kabileyeti ile zenginleştirerek, Arapça morfolojik çözümleyici ve üretici geliştirmişler. Özellikle, isim - fiil türetimsel morfolojinin nasıl ele alındığı da açıklanmıştır. Bu çalışmada FST'ler özellik yapılarıyla birleştirilerek geliştirilmiştir. Standart iki seviyeli sistemlerle bu çalışma arasındaki fark, morfem kategorilerinin kullanılmasıdır. Bu morfem kategorileri tanımlanan kurallar içerisinde yer almaktadır. Geliştirilen sistem hem istatistiksel hem de kural tabanlı makine çevirisi sitemlerinin bir bileşeni olarak kullanılabilecektir.

Ek Tipi	Morfem Tipi	Sözlüksel Kategor	Kategori Gösterimi	Anlamsal Özellik	Ek	Ç. H.	GNPC	Rev
Son ek	Çekim eki	sNoun	Nn.CmnNn.p	Canlandırma	x	-1	MS*D	T
		tNoun	Nn.CmnNn.p	Canlandırma	y	-1	FS*D	

ÇİZELGE 2.1.: Bhavsar'ın tanımladığı kurallar [1]. **Ç:K** - çıkarılacak harf (trimming character, **GNPC** - gnp ve durum özellikleri, **Rev** - ters işlem, **x ve y** - gerçek ekler, **sNoun** - kaynak sözcük kategorisi (isim) ve **tNoun** - hedef sözcük kategorisi

Goyal ve Lehal [6], Hintçe için morfolojik çözümleyici ve üretici geliştirmişlerdir. Çalışma, Hintçeden Pencap diline (Punjabi) makine çevirisi sisteminin bir parçası olarak geliştirilmiştir. Geliştirilen sistemin amacı arama süresini azaltmak ve sonuçların doğruluğunu arttırmaktır. Bu amaç için, Hint dilinde sıkça kullanılan sözcük kökleri için bu köklerin tüm formlarını (çekimsel ve yapımsal) kullandıkları veritabanında saklanmıştır. Bu şekilde ciddi bir depolama alanı ihtiyacı doğmuş olsa da, sözcük türleri için kural veya paradigmaları kullanan sistemlere göre daha kısa cevap süresi ve doğruluk oranı elde etmişlerdir.

Bhavsar [1], verilen bir sözcük için kural tabanlı morfolojik türetme sistemi geliştirmiştir. Hintçeden, Marati (Marathi) dile makine çevirisi sistemi için geliştirilen sistem, belirlenen sözcük köklerinin çekimsel ve türetimsel formlarını üretmeyi amaçlamıştır. Tanımlanan kurallar, kaynak sözcüğün sözlüksel kategorisi (isim, fiil vb.), kelime sonu işaretleri, cinsiyet, sayı ve kişi (gnp - gender, number, person) özellikleri, durum, fiiller için zaman, görünüm ve kip (tam - tense, aspect, mood) özellikleri, anlamsal özellikler, çıkarılacak harfler, eklenecek ön ek ve son ekler ve hedef sözcük için gnp, tam, durum gibi özellikleri içerir. Kuralların basit bir gösterimi Çizelge 2.1.'de gösterilmektedir. Tanımlanan kurallar diğer diller ve farklı uygulamalar için kullanılabilecek şekilde kolay bir yapıda hazırlanmıştır. Kural tabanlı bu sistem morfolojik çözümleme için de kullanılabilmektedir.

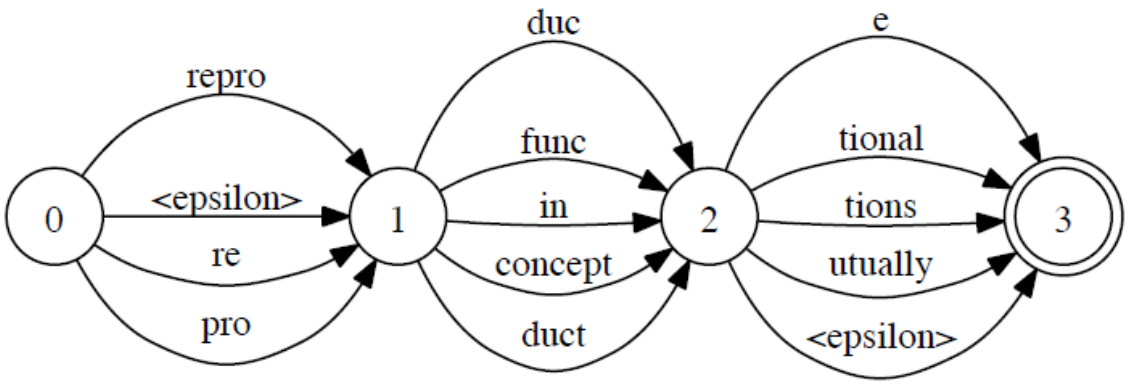
Morfolojik türetmenin en sık kullanıldığı alanlardan biri makine çevirisidir. Minkov [59], olasılıksal model ile, makine çevirisi için karmaşık morfolojik türetme çalışması gerçekleştirmiştir. Birçok diğer makine çevirisi sistemi, kaynak ve hedef dillerdeki sözcüklerin yüzeysel formlarını ayrı ayrı, birbirinden bağımsız olarak, modellemeye çalışmışlardır. Bu ve benzeri sözcük tabanlı sistemler veri seyrekliği problemlerine karşı dayanıksız olabilmektedir. Minkov'un çalışmasında ise kaynak dilden hedef dile örnek çeviri çiftleri kümesi ve bu her iki dildeki çeviri cümleleri için morfoloji ve sözdizimi bilgileri kullanılarak üretilecek sözcükler tahmin edilmeye çalışılmıştır. Deneylerini İngilizce-Rusça ve İngilizce-Arapça çeviri yaparak gerçekleştirmişlerdir. Rusça ve Arapça morfolojik olarak İngilizceye göre daha

zengin dillerdir. Morfolojik olarak fakir dillerden, zengin dillere makine çevirisi daha zorlu bir iştir [60]. Bu çalışmada, kaynak-hedef dilleri arası çeviri çiftleri ve her iki dil için kök bulma (stemming), çekimleme ve morfolojik çözümleme işlemlerini destekleyen sözlükler kullanılmıştır. Öncelikle kökleri bulunan sözcüklerden çeviri yapılacak dil için çekimlenmiş sözcük formları tahmin edilmeye çalışılmıştır. Bu işlem için Maksimum Entropi Markov modeli [61] kullanılmıştır. Çalışma sonunda kısıtlı kaynaklarla bile, dili modelleyerek çeviri yapan sistemlere göre daha yüksek doğruluk oranları yakalamışlardır.

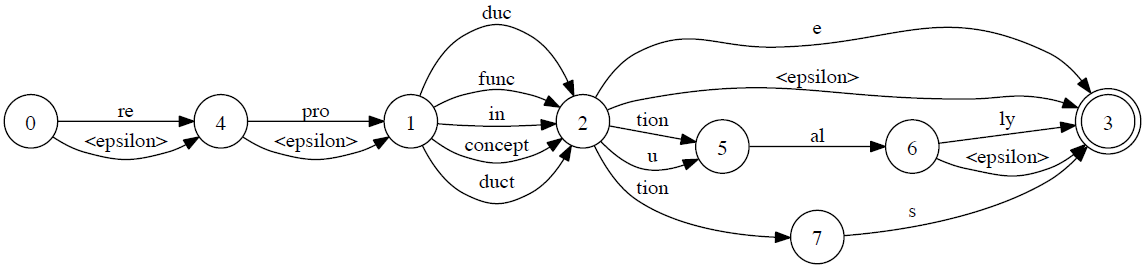
Buraya kadar denetimli morfolojik türetme çalışmalarından bahsedildi. Rasooli [2], kaynak veri miktarı az olan diller için, eldeki kısıtlı veri ile bir morfolojik türetme sistemi ihtiyacını ortaya koymuş ve tamamen denetimsiz bir morfolojik türetme çalışması yapmıştır. Hiçbir işaretlenmiş veri kullanılmadan, dilin morfolojik yapısı ve sözcüklerin türleri (POS - part of speech) hakkında hiçbir varsayım yapılmadan, sınırlı bir veri kümesi kullanılarak morfolojik türetme çalışması gerçekleştirilmiştir. Çalışmada sonuçlar yedi sınırlı kaynaklı, dil ile değerlendirmiştir: Asamez (Assamese), Bengalce (Bengali), Afganca (Pashto), Farsça (Persian), Tagalogca (Tagalog - Filipincenin temelini oluşturan bölgesel bir dil), Türkçe ve Zuluca (Zulu).

Rasooli'nin [2] çalışması üç adımdan oluşmaktadır. Öncelikle, işaretlenmemiş veriler denetimsiz olarak morfemlerine ayrılmıştır. Bu amaç için yeni bir çalışma yapılmamış ve Creutz ile Lagus'un [13], Morfessor CAT-MAP (v. 0.9.2) isimli, denetimsiz morfolojik çözümleyici kullanılmıştır. Bir sonraki adımda, ağırlıklı sonlu durum dönüştürücüleri (WFST - weighted finite state tranducer) kullanılarak sözlük genişletme (veri kümesinden bulunmayan sözcükleri üretme) gerçekleştirilmiştir. Son olarak üretilen çok sayıdaki sözcüğü sınırlandırmak için yeniden derecelendirme (reranking) çalışması yapılmıştır.

Rasooli [2], sözcük üretme çalışması sırasında iki farklı model kullanmıştır. Sabit ek modeli olarak adlandırılan ilk modelde, ön ekler ve son ekler kendi aralarında birleştirilmiştir. Sözcük köküne kadar olan tüm ön ekler bir bütün olarak ele alınmış ve benzer şekilde sözcük kökünden sonraki tüm son ekler de tek bir bütün şeklinde kabul edilmiştir. Örneğin *re + pro + duc + e* olarak morfemlerine ayrılmış, İngilizce, *reproduce* (yeniden üretmek) sözcüğünde *repro* ve *e* birer bütün olarak ele alınmıştır. Türkçedeki *çiçeklerimden* sözcüğündeki tüm son ekler, *lerimden* şeklinde FST'de yer almıştır (bkz. Şekil 2.5.). Diğer model olan, ikili ek modelinde, tüm ekler ayrı ayrı ele alınmıştır (bkz. Şekil 2.6.). Özellikle Türkçe gibi morfolojik olarak zengin dillerde bu modelin daha başarılı sözcükler üretmesi beklenmektedir.



ŞEKIL 2.5.: Sabit ek (fixed affix) modeli için FST. **<epsilon>** ek olmadığı durumlarda kullanılır [2]



ŞEKIL 2.6.: İkili ek (bigram affix) modeli için FST. **<epsilon>** ek olmadığı durumlarda kullanılır [2]

Rasooli'nin [2] çalışmasında, üçüncü adım olarak, aşırı üretimleri engellemek için, oluşturulan WFST'ler yeniden derecelendirilmiştir. Bu iş için dört farklı sınır yaklaşım kullanılmıştır. İlk derecelendirme ayarı, hiç derecelendirmenin yapılmamasıdır. İkinci ayar ise, trigraf tabanlı yeniden ağırlıklandırmadır. Bu ayarda, üçlü sıralı harf gruplarının olasılıkları baz alınmıştır. Örneğin *çiçeklridn* sözcüğündeki *klr* ve *idn* üçlülerinin görülme olasılığı oldukça düşük olacağı için bu yanlış sözcüğünde üretilme olasılığı azalacaktır. Diğer bir ayar da trigraf tabanlı yeniden derecelendirmedir. Bu ayarda, ana WFST ile trigraf FST birleştirilmektedir. Yeni üretilen FST'de, üçlü harf gruplarının görülme olasılıklarıyla üretilecek yeni sözcüklerin olasılıkları eşitlenmiş olur. Son ayar olan morfem sınırları ile yeniden derecelendirme de, morfem sınırları yakınındaki trigraflar dikkate alınır. Diğer ayarlara göre daha az sayıda trigraf olasılığı kullanılır.

Rasooli'nin [2], bu çalışmasında fazla ve yanlış türetmeler oluşmuştur. Her bir ek aynı özellikte kabul edilmiş ve her bir ekin başka diğer bir eki alabileceği kabul edilmiştir. Hem kök hem de ek kategorilerinin bulunması için bir çalışma yapılmamış olmasından dolayı bu fazla

üretimler meydana gelmiştir. Morfolojik türetme alanındaki en yakın zamanda yapılan çalışmalardan biri olduğundan, bu tezde geliştirilen model, bu çalışma ile karşılaştırılacaktır.

3. MODEL

Sözcük türetme modelinin amacı, bir veri kümesini işleyip bu küme içerisinde yer almayan sözcüklerin türetilmesidir. Türkçe dilbilgisine uygun sözcüklerden oluşan bir yapı, öğrenme verisi olarak kullanılacak ve buradan yeni sözcükler türetilecektir.

Diller morfolojik yapılarına göre birkaç ana başlık altında incelenir [62]. Çözümleyici (analytic) diller ekler konusunda fakir dillerdir. Çoğunlukla hiç çekimlenmezler veya çok az çekimlenirler. İngilizce ve Çince bu grupta yer alırlar. Birleşimsel (synthetic) dillerde ise sözcükler içerisinde yer alan morfem sayısı daha fazladır. Bireşimsel dillerden, çok çekimlenebilen ve eklerin sözcüklere eklenerek yeni sözüklerin türetildiği dillere ise sondan eklemeli (agglutinative) diller denir [63]. Türkçe, Japonca, Fince ve Macarca gibi diller eklemeli dillerdir.

Sondan eklemeli dillerde sözcüklerin yapısı genel olarak *ön ek + sözcük kökü + son ek* şeklinde olmaktadır. Ön ek ve son ek bir sözcük içerisinde yer almak zorunda değildir. Türkçede az da olsa, Latince ve Farsça gibi dillerden geçmiş ve kullanımı olan ön ekler mevcuttur [31]. Çoğunlukla sondan eklemeli bir dil olan Türkçede, bir sözcük kökünden sonra gelebilecek ek sayısında, teorik olarak, bir sınır yoktur. Bununla birlikte gelen ekler anlamları ve görevleri çerçevesinde belli bir sıra ile gelirler. Bu sıra dilbilgisi kapsamında kurallarla belirlenmiştir [20, 64]. Örneğin, *çekim ekinden sonra yapım eki gelemez* kuralı bu kurallardan sadece biridir. Sözcük türetmek için bu tür kuralları bilmek gerekir. Yazımsal (ortografik) kurallar olarak adlandırılan bu kurallar, gözlenen ekler ve bunların birbiri ardına gelmeleri durumları incelenerek tespit edilebilir.

3.1. Ön Çalışma ve Motivasyon

Çalışmadaki ilk motivasyon benzer ekleri alan sözcüklerin benzer olabileceği yani benzer ek almaya devam edebileceğiydi.

Veri kümesinde *çiçek + çi*, *sepet + çi* ve *çiçek + lik* sözcükleri olduğunu varsayarsak, bu kümeden yola çıkarak *sepet + lik* sözcüğü de türetilebilir. *Çiçek* ve *sepet* sözcük kökleri, *çi* yapım ekini almışlardır. Bu durum, iki sözcüğün benzer olabileceğini gösterir. Bu öngörüden yola çıkarak sepet sözcüğünün *lik* eki alabileceği tahmin edilebilir. Bu tahmin sadece sözcük kökleri için değil ekler için de yapılabilir. Veri kümesinde bulunan *kalem + lik + ten* sözcüğünden faydalanılarak *sepet + lik + ten* sözcüğünün de türetilebileceği öngörülebilir.

Bu öngörü bir takım eksiklikler barındırmaktadır. En büyük eksikliği eklerin yazımsal olarak (surface form) benzerliğine dayanıyor olmasıdır. Türkçede bir ek kendisinden önce gelen sözcükle ses uyumuna göre uygun bir yapı alır. Örneğin yukarıdaki örnekte yer alan *çi* eki ses uyumlarına göre sözcüklerin sonuna *çı*, *cı* ve *ci* şeklinde de gelebilir. *Kitap* sözcüğünden sonra Türkçedeki ses kuralları çerçevesinde *çı* eki gelir. Yukarıdaki örnekte yer alan öngörüye göre *çı* ve *çi* eklerinin aynı (aynı anlam ve görev) ek olup birbirinin alomorfu olduğu bilinirse, birbirlerinin aldığı ekleri alabilecekleri tespit edilebilir.

Tespit edilen diğer önemli eksiklik ise, bu ön çalışmanın benzer anlam ve görevdeki sözcük köklerinin veya alomorf olmayan eklerin benzer ekleri alabileceği durumunu ele alamamasıdır. Örneğin *çi* ve *lik* eklerinin ikisi de isimden isim türeten yapım ekidir. Yani görev olarak aynıdır. Doğal olarak bu ekler birbirlerinin alabileceği ekleri alabilirler. Aynı durum sözcük kökleri için de geçerlidir. Bu sefer aynı türe ait sözcüklerin aynı ekleri alabilmesi söz konusudur. *Çiçek* ve *sepet* sözcükleri isim oldukları için benzer ekleri alabilirler.

Bütün bu gözlemler sonucunda sözcük türetmek için şu iki çıkarım yapıldı:

- Aynı türdeki sözcük kökleri aynı ekleri alabilir.
- Aynı görev veya türdeki ekler de aynı ekleri alabilir.

Bu öngörüyü doğrulayabilmek için bir çalışma yapıldı. Özel olarak seçilmiş, birkaç sözcük eklerine ayrılarak bu kökler ve ekler türleriyle birlikte işaretlendi (bkz. Çizelge 3.1.). Eklerine ayırma ve işaretleme işlemi, Türkçe için doğal dil işleme kütüphanesi olarak geliştirilen, açık kaynak koduna sahip Zemberek[21] kullanılarak gerçekleştirildi. Zemberek bir sözcüğün eklerine ayrılmış hali için birden fazla çıktı üretir. Bunun sebebi bir ekin aynı fonetik yapıda olmasına karşın farklı görevlerinin olabilmesidir. Örneğin *i* eki hem tamlama eki hem de belirtme eki olarak kullanılabilir. Zemberek'ten elde edilen çıktılar kullanılarak aynı türden sözcük kökleri veya ekleri, aynı ekleri alabilirler fikriyle yeni sözcükler türetildi. Sonuçta üretilen 101 biricik sözcükten 10 tanesi yanlış çıktı ve %90 oranında bir doğruluk hesaplandı. Üretilen sözcüklerden bir kesit Çizelge 3.2.'de verildi.

çiçek*ISIM_KOK+çi*ISIM_ILGI_CI
çiçek*ISIM_KOK+lik*ISIM_BULUNMA_LIK+çi*ISIM_ILGI_CI
sepet*ISIM_KOK+çi*ISIM_ILGI_CI
dönüş*FIIL_KOK+tür*FIIL_ETTIRGEN_TIR+me*FIIL_DONUSUM_ME+ye*ISIM_YONELME_E
dönüş*FIIL_KOK+tür*FIIL_ETTIRGEN_TIR+me*FIIL_OLUMSUZLUK_ME+ye*FIIL_ISTEK_E
dön*FIIL_KOK+üş*FIIL_BERABERLIK_IS+tür*FIIL_ETTIRGEN_TIR+me*FIIL_DONUSUM_ME+ye*ISIM_YONELME_E
yelek*ISIM_KOK+ten*ISIM_CIKMA_DEN+miş*IMEK_RIVAYET_MIS
kitap*ISIM_KOK+tan*ISIM_CIKMA_DEN

ÇİZELGE 3.1.: Örnek girdi, eklerine ayrılmış ve türleriyle beraber işaretlenmiş sözcükler

Yanlış çıkan sözcüklere bakıldığında karşılaşılan en büyük problemin yumuşama gibi ses olaylarının ele alınmıyor olması olarak gözlemlenmiştir. Ses olaylarını tespit edip sözcük türetme aşamasında kullanabilmek için yazımsal (ortographic) kurallar bulunmaya çalışılmıştır. Çizelge 3.2.'de görüldüğü üzere *düşeceğ* sözcüğünde son harfteki yumuşama ses olayı yanlış yerde oluşmuştur.

sepet	Doğru
sepetçi	Doğru
sepetlik	Doğru
sepetlikçi	Doğru
sepetlikten	Doğru
sepetliktenmiş	Doğru
düş	Doğru
düşüş	Doğru
düşüşten	Doğru
düşüştenmiş	Doğru
düşüşçi	Yanlış
düşeceğ	Yanlış
düşeceğiz	Doğru

ÇİZELGE 3.2.: Üretilmiş sözcüklere örnekler

Çalışmadaki sözcük sayısı artırılıp sonuçlar değerlendirildiğinde bir başka problem daha gözlemlenmiştir. *Kitapçılar* sözcüğünün veri kümesinde yer aldığı varsayıldığında, *çı* eki *lar* eki aldığı için aynı kategoride yer alması beklenen *çi* ekinin de aynı eki yani *lar* ekini alması beklenir. Fakat Türkçedeki ünlü ses uyumundan dolayı alması gereken doğru ek *ler* (örneğin *çiçekçiler*) olacaktır. Anlatılan ünlü ses uyumu problemi yanlış olarak türetilmiş *düşüşçi* sözcüğünde de gözlemlenmiştir (bkz Çizelge 3.2.).

Benzer olarak Türkçedeki ses olaylarından biri olan ünsüz benzeşmesinde de sert ünsüzle biten bir sözcükten sonra sert ünsüz, yumuşak ünsüzle biten bir sözcükten sonra ise yumuşak ünsüz harf ile başlayan bir ek gelir. Örneğin ikisi de isim olan *kitap* ve *okul* sözcük köklerinin aynı ekleri alması beklenir. Fakat ayrılma/çıkma hal ekini *kitap* sözcüğü *tan* olarak alırken *okul* sözcüğü *dan* olarak alacaktır.

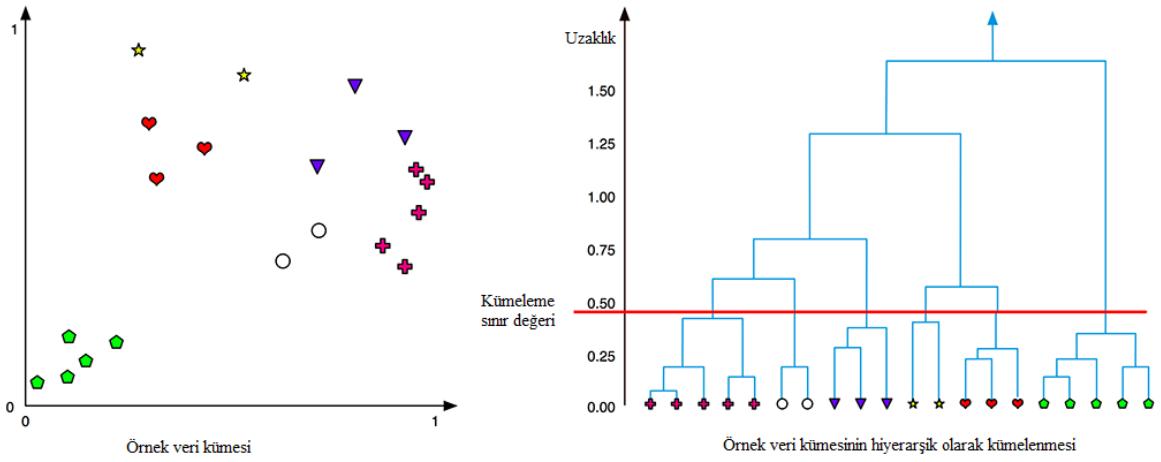
Buraya kadar, yapılan ön çalışma anlatılmıştır. Bu ön çalışma ile kök ve ek kategorilerinin ayrı ayrı modellenmesi gerektiği ve ses olaylarının da ayrıca ele alınması gerektiği anlaşılmıştır. Bu amaçla sözcük türetebilmek için öncelikle veri kümesindeki sözcük köklerini türlerine ayırmaya çalıştık. Bunu denetimsiz (unsupervised) olarak bulmaya çalıştık. Daha sonra aynı işlemi benzer olarak ekler için de gerçekleştirdik. Sözcük türetme modelini hayata geçirebilmek için öncelikle bu iki çalışma anlatılmıştır. Daha sonra sözcükleri türetirken kullanılacak harf ikilileri ve yazımsal kurallar anlatılmıştır.

3.2. Veri Kümesi

Tez çalışması sırasında Morpho Challenge 2009 derleminden elde ettiğimiz bir metin, eğitim verisi olarak kullanıldı. Bu metin 1 milyon adet cümleden oluşmaktadır. Biz buradan sadece 5000 cümle alarak çalışmalarımızı gerçekleştirdik. Bu 5000 cümlelik veriden elde edilen sözcükler, Zemberek yardımıyla eklerine ayrıştırıldı. Sonuç olarak eklerine ayrılmış 83618 sözlüksel (lexical) formda sözcük elde edildi. Zemberek'in bir sözcük için birden fazla çıktı üretebildiği ve üretilen sözlüksel formların biricik olmadığı belirtilmelidir. Bundan sonraki deneyler gerçekleştirilirken, bu veri kümesinin tamamı veya bir kısmı kullanıldı.

3.3. Kök Kategorilerinin Bulunması

Bu çalışma sonunda beklenen, sözcük köklerinin isim, fiil, sıfat ve benzeri türlerine ayrılmasıdır. Bağlaç, edat gibi sözcükler, yeni sözcük türetme açısından zengin olmadıkları için ihmal edildi. Sıfat, zarf ve zamirler ise aldıkları eklerden ötürü isimlerle benzer yapıya sahiptir. Bu yüzden bu çalışmada amaç, son durumda iki ana kategorinin oluşmasıdır: İsimsel (nominal) kökler ve fiil (verbal) kökler. Sözcük köklerini kategorilere ayırabilmek için öncelikle olasılık yaklaşım, bilgi teorisi yöntemleri ve word2vec modeli sırasıyla uygulandı.



ŞEKIL 3.1.: Hiyerarşik yığınsal kümeleme algoritması [4]

Olasılıksal Yöntem

Daha önceki bölümlerde bahsedildiği üzere, Türkçede aynı türdeki sözcük kökleri aynı ekleri alabilirler. Bu bilgiye tersten bakarsak, bir sözcüğün aldığı ek veya ekler o sözcüğün türü hakkında bize bilgi verir. Yani bir sözcüğün kendisinden sonra gelen eklerine bakarak o sözcüğün türünü tespit edebiliriz. Bu bölümde sözcük köklerini kümelemek için olasılık tabanlı uzaklık metrikleri olan ıraksama (divergence) ve benzer olarak Jaccard mesafesi kullanıldı.

Bu çalışmada sözcük köklerinin aldığı ekler üzerine olasılık dağılımları hesaplandı. Her kök için oluşan bu dağılımlar karşılaştırıldı. Olasılık dağılımları benzer olan köklerin, başka bir deyişle birbirine yakın olasılık dağılımı gösteren köklerin aynı tür sözcük kökleri olduğu çıkarımı yapıldı.

Çalışmada hiyerarşik yığınsal (agglomerative) kümeleme algoritması olarak adlandırılan yöntem kullanıldı (bkz. Şekil 3.1.). Algoritmadaki kategorilerin benzerliklerinin hesaplanması aşağıdaki metrikler kullanılarak gerçekleştirildi. Bu yöntemde, ilk durumda, sözcük köklerinin teker teker ayrı bir kategoride olduğu varsayılır. Her iterasyonda iki kategori birleştirilir. Bu iterasyonlarda kategorilerin birbirlerine göre benzerlikleri hesaplanır. Sıra gözetmeksizin birbirine en çok benzeyen iki kategori birleştirilir. Kategoriler birleştirildikten sonra, birleşen kategorilerin aldığı ekler de birleştirilir, kaynaştırılır. Daha sonra bu işlem belirlenen sınır değerine ulaşılıncaya kadar devam eder.

Kullback-Leibler ıraksama (KL divergence) iki olasılık dağılımı arasındaki mesafeyi, ilişkiyi ölçer [65]. Ölçülen uzaklık simetrik değildir. Yani P olasılık dağılımının Q olasılık dağılımına uzaklığı, Q olasılık dağılımının P dağılımına uzaklığına eşit olmayabilir. KL ıraksama yöntemi $D_{KL}(P\|Q)$ olarak gösterilir. Bu yöntemde çoğunlukla P doğru ya da olması beklenen olasılık dağılımı olarak kullanılırken Q test dağılımı olarak kullanılır. Okunurken de Q'dan P'ye olan uzaklık şeklinde ifade edilir. Aşağıdaki formül ile hesaplanır.

$$D_{KL}(P\|Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \tag{1}$$

Bu çalışmada sözcük köklerinin olasılık dağılımları birbirleriyle karşılaştırıldı. Başlangıçta kök türlerinin olasılık dağılımları hakkında bir bilgi mevcut değildir. Yapılacak karşılaştırma, beklenen bir dağılıma göre değil iki kökün birbirlerine göre dağılımları arasında yapıldı. Bu amaçla Jensen–Shannon ıraksama (JS divergence) metriği, köklerin benzerliğini tespit etmek için kullanıldı. Jensen–Shannon ıraksama metriği KL ıraksama metriğini temel alarak sonlu ve simetrik bir ölçüm sunar. Jensen–Shannon uzaklığı olarak da isimlendirilen metrik, basitçe KL ıraksama metriğinin her iki yönden aritmetik ortalamasını verir. Bu değerin küçük çıkması dağılımların benzer olduğunu gösterir. KL değeri negatif değer almaz ve JS ıraksama metriği de negatif değer almayacaktır. Ölçülen uzaklığın 0 değerini alması, olasılık dağılımların birbirlerini aynısı olduğunu anlamına gelir.

Bu yöntemde sözcük köklerinin aldığı ekler üzerine oluşan olasılık dağılımları JS ıraksama metriği kullanılarak karşılaştırıldı. JS metriği ile uzaklık hesaplanırken sözcük kökünden sonra gelen, sadece ilk ek göz önüne alındı. Bunu sebebi kökün türü hakkındaki en net bilgiyi kendisinden sonra gelen ilk ekin vermesidir. Bu durumu şu şekilde açıklayabiliriz: *su* sözcüğünün türü isimdir. *la* eki ise isimden fiil yapan yapım ekidir. Yani bu ek sadece isim köklü (nominal) bir sözcüğe gelebilir ve bu sözcüğün türünü fiil olarak değiştirir. Çalışmanın başında bahsedilen çıkarıma göre *la* ekini alan sözcüğün türünü isim olduğu söylenebilir. Örneğimizdeki *su* sözcüğü *la* ekini aldıktan sonra oluşan *sula* sözcüğü artık fiil türünde bir sözcüktür. Yani bu aşamadan sonra gelebilecek ekler fiillere gelebilecek olan eklerdir. *Sula* sözcüğü daha sonra *mak* mastar ekini alabilir ki mastar ekleri sadece fiil türündeki sözcükler tarafından alınabilir. Örnekte görüldüğü üzere *mak* eki *su* sözcük kökü açısından bir bilgi vermemektedir.

Bu çalışmada kullanılacak diğer bir metrik ise Jaccard uzaklığıdır. Bu yöntem ile iki sözcük kökü arasındaki benzerlik aldığı ekler ile hesaplanmaya çalışıldı. Jaccard mesafesi, Jaccard benzerlik katsayısının birden çıkarılmasıyla elde edilir. Sözcük köklerinin aldığı ekler birer küme olarak düşünülüp bu kümelerin benzerlikleri hesaplanmaya çalışıldı. JS ıraksama yönteminde olduğu gibi hesaplamada sözcük kökünden sonra gelen ilk ekler ele alındı.

Sözcük köklerini kategorilerine ayırma işlemi yapılırken bir veya daha az çeşit ekle görülmüş kökler ihmal edildi. Bu sayede, aldığı ekler açısından, kendisi hakkında yeterli bilgi vermeyen sözcüklerin gürültü oluşturmasının önüne geçildi.

S_1 ve S_2 diye iki sözcük kökünün benzerliği JS uzaklık metriği ile hesaplanırken ilk olarak aldıkları ekler ortak bir kümede toplandı. Veri kümesinde yer alan tüm eklerin S_1 ve S_2 sözcük kökü için olasılığı hesaplandı. $P(S_m)$, S ekinin m ekini alması olasılığı olarak ifade edilir. Hesaplanırken S sözcük kökünün aldığı tüm eklerin görülme sıklıkları toplandı. Ve m ekini aldığı örnek sayısı bu toplama bölündü. Eğer S sözcüğü m ekini hiç almamışsa bu olasılık 0 çıkacaktır. Eğer veri kümesi içerisinde S sözcüğü sadece m eki ile birlikte görülmüşse de olasılık değeri 1 olacaktır. $P(S_m)$ değeri, karşılaştırılan tüm kökler ve bu köklerin aldığı tüm ekler için hesaplandı. Karşılaştırılan iki sözcük kökü için bu işlem sadece aldığı eklerin birleşim kümesindeki ekler için yapıldı. Diğer eklerin olasılığı o iki kök için 0 olacağından dolayı hesaplanmadı. Daha sonra her ek için hesaplanan P değerleri ile KL ıraksama formülü uygulandı.

$$D_{KL}(S_1||S_2) = \sum_{m_i \in M} p_{S_1}(m_i) \frac{p_{S_1}(m_i)}{p_{S_2}(m_i)} \tag{2}$$

Burada $M = M_1 \cup M_2$ olarak ifade edilir. M_1, S_1 sözcük kökleriyle beraber görülen ekler kümesiyken, M_2 ise benzer şekilde S_2 kökleriyle beraber görülen eklerin kümesi olarak ifade edilir. Daha sonra da JS ıraksama metriği, KL ıraksama metriğinin aritmetrik ortalaması şeklinde hesaplanır.

$$JSD(S_1 \| S_2) = \frac{1}{2} D_{KL}(S_1 \| S_2) + \frac{1}{2} D_{KL}(S_2 \| S_1) \tag{3}$$

Yukarıdaki formüle bakıldığında P_{S_2} değerinin 0 gelebilmesi mümkündür. S_2 kökünün m_i ekini hiç almadığı durumlarda bu değer 0 olarak hesaplanır. Payda yer alan 0, sonucu tanımsız kılacağı için istatistikte kullanılan bir yöntem olan düzeltme (smoothing) uygulandı.

En basit haliyle yapılan düzeltmede pay ve paydaya oldukça küçük bir değer eklendi. Çalışmada bu değer 10^{-4} olarak seçildi. Buna ek olarak payda yer alan olasılık değeri de yine 0 olarak hesaplanabilir. Burada $\log 0$ tanımsız olur. Fakat formülün başında paydaki değer çarpım şeklinde yer alacağı için formül sıfır olarak hesaplandı.

Jaccard mesafesi ise hesaplanırken karşılaştırılacak sözcüklerin aldığı ekler birer küme olarak kabul edilir. Bu kümelerin kesişim ve birleşimleri oluşturulur. Jaccard indeksinin birden çıkarılmasıyla elde edilen ve dj olarak ifade edilen Jaccard mesafasi, $S1$ ve $S2$ kökleri için aşağıdaki gibi hesaplanır:

$$d_J(S_1, S_2) = 1 - J(S_1, S_2) = \frac{|S_1 \cup S_2| - |S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{4}$$

Her iki metrik kullanılarak oluşturulan kategorileri karşılaştırmak için veri kümemizden belli sayıda sözcük kökünü kategorize etmeye çalıştık. Burada kategorilerin birbirlerine göre üstünlüklerini saflık (purity) değer ile ölçmeye çalıştık. Elde edilen kategorilerin saflık değerleri hesaplanırken karakteristik özellikleri daha belirgin olan isim, eylem ve sıfat türleri baz alınmıştır. Doğru sözcük kökü kümelerinin (gold clusters) oluşturulmasında sözcük köklerinin türlerini de verdiği için Zemberek [21] kullanılmıştır. Saflık değerinin hesaplanmasında aşağıda verilen formül uygulanmıştır:

$$\sum purity(S; C) = \frac{1}{N} \sum_k max_j |S_k \cap C_J| \tag{5}$$

S uyguladığımız kümeleme algoritması sonucunda elde edilen kök kümelerini ifade ederken, C ise Zemberek tarafından elde edilen doğru sözcük kökü kümelerini ifade etmektedir. Böylece her sonuç kümesi doğru kümelerden hangisiyle en fazla ortak köke sahipse o doğru kümeye eşleştirilmektedir.

Çizelge 3.3.'te JS ıraksama ve Jaccard uzaklığı metrikleriyle oluşturulan kategorilerin saflık değerleri verilmiştir. Burada 100, 200, 300, 400, 500 biricik sözcük kökünden oluşan veri kümeleri kullanıldı. Deneyler gerçekleştirilirken JS ıraksama metriği için sınır değeri 2, Jaccard uzaklığı için ise 0.8 olarak belirlendi. Çizelge 3.3. incelendiğinde JS ıraksama metriği kullanılarak oluşturulan kümelerin saflık değerlerinin Jaccard uzaklığına göre daha iyi olduğu gözlemlendi. Saflık değerleri hesaplanırken sadece 3 sözcük türü ile karşılaştırılma yapıldığından, mükemmel durumda oluşması beklenen son kategori sayısı da 3 olacaktır. Fakat

Sınır Değeri	Saflık	Kategori Sayısı
1	0.98	94
1.2	0.98	94
1.5	0.97	82
1.8	0.97	74
2	0.97	66

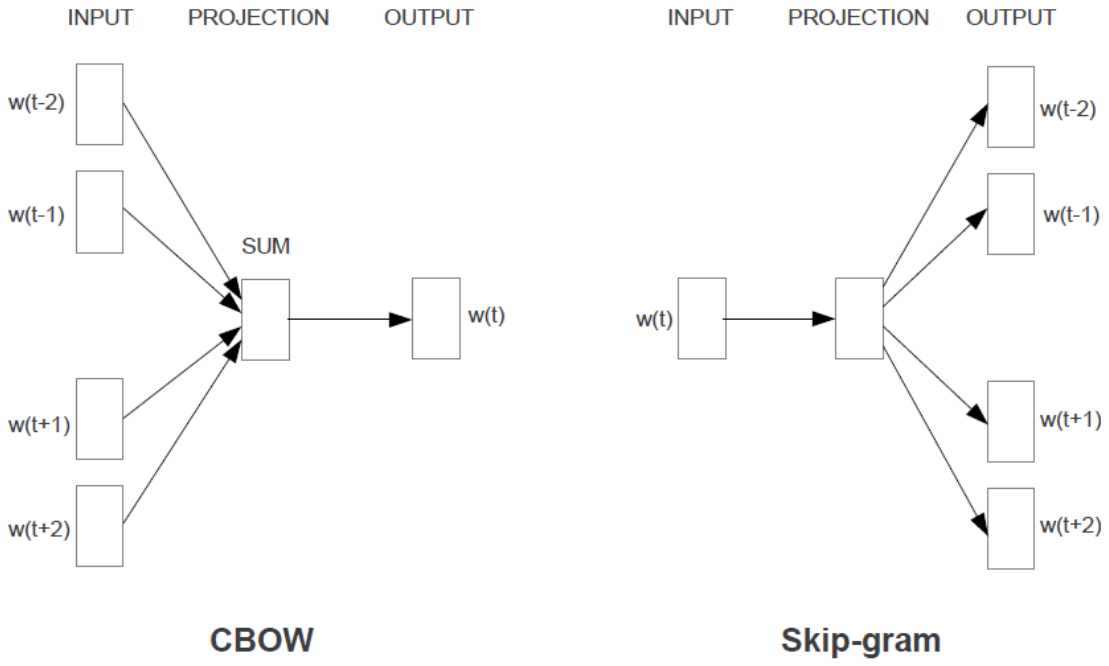
ÇİZELGE 3.4.: JS ıraksama metriği için kullanılan sınır değerlerinin oluşan kategori sayısı ve saflık değerleri üzerindeki etkisi.

son kategori sayılarının bu değerden oldukça yüksek olduğu gözlemlendi. Bir metriğin başarılı sayılabilmesi için saflık değeri en önemli kriter olacaktır. Fakat buna ek olarak oluşan son kategori sayısı da metrik hakkında bilgi vermektedir. Bu açıdan bakıldığında ise Jaccard mesafesi metriğinin daha başarılı olduğu gözlemlenmektedir.

	Jensen-Shannon Iraksama		Jaccard Mesafesi	
100	0.97	66	0.95	42
200	0.96	109	0.95	63
300	0.96	163	0.936	72
400	0.96	209	0.93	87
500	0.956	228	0.93	92

ÇİZELGE 3.3.: JS ıraksama ve Jaccard mesafesi yöntemlerinin 100, 200, 300, 400, 500 sözcüklük veri kümeleriyle gerçekleştirilen deney sonuçları sırasıyla saflık değeri ve son kategori sayılarıyla birlikte verilmiştir.

Yukarıda bahsedilen sınır değerleri, her iki yöntem için de, ne kadar yüksek olursa kategoriler o kadar çok birleşebilecek ve daha az sayıda, içerisinde daha çok sözcük bulunduran kategoriler oluşacaktır. Fakat oluşan kategorilerin büyük olması, saflık değerlerinde olası bir düşüşü de beraberinde getirebilecektir. Bu öngörüyü doğrulayabilmek için JS ıraksama metriğinin sınır değerleri değiştirilerek deneyler yapıldı. Çizelge 3.4.'te bu deneylerin sonuçları gözlenmektedir. Burada 100 biricik sözcük kökünden oluşan veri kümesi kullanıldı. Sınır değeri 1, 1.2, 1.5, 1.8 ve 2 olarak 5 farklı değer aldı. Bu değerlere baktığımızda saflık değerlerinde büyük bir değişiklik gözlemlenmedi. Buna karşın oluşan son kategori sayıları ise öngördüğümüz ölçüde değişti.



ŞEKIL 3.2.: Word2vec modelinde kullanılan mimariler [5]

Word2Vec Modeli ile Sözcük Köklerinin Kümelenmesi

Word2vec sözcüklerin vektör uzayında çok az boyutlu olarak ifade edilmesini sağlayan bir modeller bütünüdür [5, 66]. Word2vec, geniş bir metni girdi olarak alır ve bu metin içerisindeki her bir sözcük, çıktı olarak üretilen vektör uzayına yerleştirilir. Bu uzayda, anlamsal olarak benzer sözcükler birbirlerine yakın bir şekilde konumlanırlar. Vektörler arasında uzaklık kosinüs benzerliği ile hesaplanır. Word2vec bu anlamsal ilişkikeri iki ayrı mimari ile kurar [5, 66]. Devamlı sözcük torbası (CBOW - continuous bag-of-words) mimarisi, sözcüğü, etrafındaki konsepte bakarak tahmin eder. Diğer mimari olan skip-gram ise hedef konsepti sözcüğe bakarak tahmin eder (Şekil 3.2.). Birçok doğal dil işleme uygulamasında kullanılabilen word2vec, bu çalışmada sözcük köklerinin kümelenmesi için kullanıldı [67]. Google'dan Tomas Mikolov ve ekibi tarafından geliştirilen word2vec modeli için, yine aynı ekip tarafından geliştirilmiş, açık kaynak kodlu uygulaması kullanıldı [67].

Çalışmada kullanılan word2vec uygulamasının Türkçeye özgü karakterler (*ü, ö, ı, ç, ş, ğ*) ile uyumlu çalışmadığı gözlemlendi. Türkçe dilinin fonetik (sesbilgisel) özelliklerine uygun

olarak bu karakterler yerine bunların allofan (eşses) karşılıkları büyük harf ile yazıldı [36]. Örneğin *öğrenci* sözcüğü *OGrenci*, *çalışması* sözcüğü *CalISmasI* şeklinde kullanıldı.

Vektör olarak ifade edilen sözcükler k-means kümeleme algoritması ile kümelerine ayrıştırıldı.

Dağılımsal modellerdeki önemli parametrelerden biri kullanılacak pencere boyutudur. Pencere boyutu, özellik vektörüne dahil edilecek olan sözcüklerin özellik vektörü çıkarılan sözcüğün ne kadar ileri ve gerisine bakarak elde edileceğini belirleyen bir parametredir (Çizelge 3.5.). Bu çalışma kapsamında her pencerede bir kökün aldığı ekler yer almaktadır.

	gerçek leş tir diğ i nin				
CBOW (2-grams)	gerçek leş	leş tir	tir diğ	diğ i	...
skip-gram (1-skip-2grams)	gerçek tir	leş diğ	tir i	diğ nin	

ÇİZELGE 3.5.: Eklerine ayrılmış olan *gerçekleştirdiğinin* sözcüğü için, pencere boyutunun 2 olarak seçilmesi ile oluşan pencereler.

Veri kümesinde sözcüklerin aldığı eklerin ortalama sayısı $1,35$ olarak hesaplandı. Yani bir sözcüğün türünü kendisinden sonra gelen ekler ile tespit edebilmemiz için pencere boyutunun en az bu sayı kadar olması gerekir. Bu çalışma sırasında pencere boyutu olarak 4 ve 5 pencere boyutları test edildi. Her kök için bu pencere boyutlarında yer alan ekler alınarak köklerin özellik vektörleri oluşturuldu. Türkçede ön ek (prefix) kullanımı fazla olmadığı için pencerelerde daha çok son eklerin (suffix) bulunduğu sözcüklerin sağ tarafı dikkate alındı.

Kümeleme algoritmalarındaki önemli parametrelerden bir diğeri de oluşacak kategorilerin sayısıdır. Bu sayı k-means kümeleme algoritmasındaki k değerine denk gelmektedir. Elde edilecek kök türlerinin sayısı isim, sıfat ve fiil gibi sadece temel kategoriler düşünüldüğünde oldukça azdır.

Ancak modelde ek ve kökler arasında bir ayrım yapılmadan hepsi bir arada kümelendirildi. Bu yüzden $k = 50$ olarak belirlendi. word2vec ile elde edilen kök kategorilerinden bazıları Çizelge 3.6.'da verildi.

rüşvet, esaret, hasret, plebisit
seyyare, enfiye, turnike, muahede, beriki, fonem
ye, ip, ti, ki, in, m, i, dik, sin, si, de, ne
ağ, aydın, sağ, tutuk, kira, av, anlatı, boya, ıska, pompa
utan, yumuşa, damla, bağda, boşal, daral, fırla, hatırla, tanı
ısıt, taşın, yararlan, çoğal, azal, yansı, payla, ula, hızlan
diploma, parola, sigara, tanrı, yumurta, tartışma, acı, anı
gelin, evren, vali, evli, gebe, üste, rehber, yüksek, amir

ÇİZELGE 3.6.: word2Vec modeli ile oluşturulan kök kategorilerinden bazıları

Deneylerde CBOW ve skip-gram modelleri ayrı ayrı seçilerek farklı pencere boyutları ile sözcük köklerinin kategorize edilmesi sağlandı. Sözcüklerin başlangıç ve bitişlerini belirten sınır karakteri de eklenerek de gerçekleştirilen deneylerde maksimum öbek sayısı 50 olarak belirlenmiş ve daha önceden seçilen sözcük kökleriyle son kümeler oluşturuldu.

Pencere	Küme Sayısı	Saflık
4	47	0.9
5	46	0.91

ÇİZELGE 3.7.: CBOW kümeleme sonuçları

Yapılan deneylerde karşılaştırma için saflık değeri ve oluşan son kategori sayısı baz alındı. İlk olarak CBOW modeli ile deneyler yapıldı. Bu deneyler sonunda saflık değeri en fazla 0.91 olarak bulunmuştur (bkz. Çizelge 3.7.). Pencere sayısının 5 olarak ayarlandığı deney en yüksek saflık değerini vermiştir. Aynı deneyler skip-gram modeli seçilerek tekrarlandığında saflık değerleri CBOW modeline göre daha yüksek çıkmıştır (bkz. Çizelge 3.8.). CBOW modelinden farklı olarak, skip-gram modelinde pencere sayısını azalması daha olumlu sonuç vermiştir.

Pencere	Küme Sayısı	Saflık
4	50	0.93
5	50	0.92

ÇİZELGE 3.8.: Skip-gram kümeleme sonuçları

Karşılıklı Bilgi Tabanlı Yöntem

Sözcük köklerini kategorilerine ayırmak için uygulanan son yöntemde Baek ve arkadaşlarının [68] geliştirdiği, karşılıklı bilgi (MI - mutual information) tabanlı bir benzerlik metriği kullanıldı. Karşılıklı bilgi, rastgele bir değişkenin diğer başka bir değişken hakkında verdiği bilgiyi ölçer. Örneğin, yıl içerisinde rastgele bir günün hava sıcaklığı, o günün hangi ayda olduğunu kesin olarak göstermez ama bir fikir verir. Buna benzer olarak ayın bilinmesi o günün sıcaklığını net bir şekilde vermez ama bir ipucu verir. X ve Y rastgele değişkenleri için $I(X;Y)$ olarak ifade edilen karşılıklı bilgi aşağıdaki formül ile hesaplanır.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{6}$$

Burada $p(x,y)$, x ve y birlikte görülme olasılıkları, bileşik olasılıkları olarak ifade edilirken, $p(x)$ ve $p(y)$ ise x ve y'nin toplam görülme olasılıkları olan marjinal olasılıklarıdır.

Bu çalışmada morfemlerin (anlambirim) birbirlerine olan benzerliklerini, diğer morfemlere bakarak hesaplayan bir metrik kullanıldı. Bu metrik ile sözlük (lexicon) içerisindeki diğer morfemlere göre karşılıklı bilgiler hesaplanarak morfemler arasındaki benzerlikler keşfedilir. Bizim çalışmamızda ise bu metrik özelleştirilerek, iki sözcük kökünün benzerliği, aldıkları eklere bakılarak hesaplanan karşılıklı bilgi ile ölçülmeye çalışıldı. Özetle ilk motivasyon olan, benzer ekleri alan köklerin benzer olabileceği fikrine göre burada da, yine kökler aldıkları ekler ile türlerine ayrıştırılmaya çalışılmış oldu.

$$Sim(s_1;s_2) = \frac{\sum_{m_i \in M_1 \cup M_2} min(MI(s_1,m_i), MI(s_2,m_i))}{\sum_{m_i \in M_1 \cup M_2} max(MI(s_1,m_i), MI(s_2,m_i))} \tag{7}$$

Burada s_1 ve s_2, benzerlikleri hesaplanacak olan köklerdir. M_1, s_1 sözcük kökü ile birlikte görülen ek kümesi ve M_2, s_2 kökü ile birlikte görülen eklerin kümesidir. Burada iki kökün benzerliği, diğer eklerle arasındaki karşılıklı bilgiye bakılarak hesaplandı. Özetle kullanılan benzerlik metriği, iki karşılıklı bilgi değeri arasındaki benzerliği ölçmektedir.

Metrik hesaplanırken, karşılaştırılacak sözcük köklerinin ekleri bir kümede toplandı. Her kökün her eki ile teker teker karşılıklı bilgileri yukarıdaki formüle göre hesaplandı. Bu formül hesaplanırken de bir s sözcük kökü ile m ekinin birlikte görülme olasılığı (birleşik olasılığı)

$p(s, m)$ hesaplandı. Daha sonra $p(s)$ ve $p(m)$ olasılıkları, yani s kökünün ve m ekinin veri kümesinde bulunma olasılıkları ayrı ayrı hesaplandıktan sonra yukarıdaki formül uygulandı. Teker teker hesaplanan MI değerleri ana formülde yerine koyuldu. Kategori oluşturma için, daha önce de kullanılan hiyerarşik yığınsal kümeleme algoritması kullanıldı.

Daha önceki verilerle yapılan deney sonuçları Çizelge 3.9.'da verilmiştir. Sonuçlara bakıldığında diğer yöntemlere göre daha az kategori sayısı oluşturulmuştur. Oluşan kök kümeleri saflık açışından biraz geride kalmışsa da daha az sayıda kategori oluşturmuştur.

	Küme Sayısı	Saflık
100	30	0.86
200	45	0.865
300	56	0.843
400	66	0.862
500	75	0.862

ÇİZELGE 3.9.: Karşılıklı bilgi tabanlı metrik ile elde edilen sonuçlar

Sözcük Kök Kategorilerini Bulmada Uygulanan Yöntem

Yukarıda uygulanan yöntemler sonucunda oluşan kategoriler arasında en yüksek saflık değeri ıraksama metriği ile elde edildi. Fakat bu yöntem oldukça fazla sayıda sözcük kökü kategorisi üretti. Buna benzer olarak word2vec modeli de fazla kategori üreten bir yöntem oldu. MI tabanlı metriğin kullanıldığı yöntem ise saflık konusunda diğer yöntemlere göre geride kalmış olsa da en az sayıda kategoriyi oluşturdu. Fakat sonuç olarak baktığımızda uygulanan yöntemlerin hiçbirisi tek başına, büyük bir farkla ön plana çıkmadı. Bu sonuçlara bakıldığında melez (hibrid) bir algoritmanın kullanılmasının uygun olacağı düşünülebilir.

Olasılıksal yöntemde veri kümesinin alt kümeleri de kullanılarak deneyler yapılmıştı. Bu aşamadan sonra yapılan testlerde, kullanılan veri kümesinin tamamı ile çalışıldı. Sonuç olarak 3049 biricik sözcük kökü kategorilerine ayrıştırılmaya çalışıldı. İlk olarak MI tabanlı yöntemi kullanan metrik ile sözcük kökleri kategorilere ayrıştırıldı ve 152 farklı kategori oluştu. Bu yöntemin üstüne oluşturulan kategoriler ıraksama metriği kullanılarak birleştirilmeye devam edildi. Sınır değerine ulaşıldıktan sonra kümeleme işlemi sonlandı ve sonunda 89 farklı kategori elde edildi.

bilim, fizik, küre, cin, gelenek, birey , evren, istatistik, matematik, zihin, beden
görüş, yürüt, öngör, öldür, sök, bük, güt, büyült, öv, götür, sürdür, düşür
kaçak, ırk, dans, kalıp, sanat, miras, balık
kur, dur, sor, uy, kor, yor, ok, hoş, otur, kaybol, sok, koştur, kop, uyuştur, vur, korkut, kavur, dokun, duyur, durdur, duy, doldur, boz, unut, bulundur, kon, tut, bulun, sun
uygun, boş, bozuk, uzun, mutlu, yoğun, memnun, buz, imparator, durgun, yolcu, tutuk, yolsuz, olumsuz

ÇİZELGE 3.10.: Köklerin kategorilerine ayrılması işlemi sonucunda oluşan bazı kategoriler

Sonuç olarak elde edilen 89 kategori halen daha yeterince iyi değildi. Bu durum analiz edilerek sorun tespit edilmeye çalışıldı. Tespit edilen ilk sorun eklerin fonetik farklılıklarıydı. Örneğin çoğul eki olan *lar* ve *ler* ekleri, tamamen aynı görevde olmalarına karşın fonetik olarak birbirlerinden farklıdırlar. Türkçedeki ünlü uyumu kuralından ötürü de *lar* ve *ler* ekleri farklı sözcükler tarafından alınabilir. Başka bir deyişle bir sözcük hem *lar* hem de *ler* ekini alamaz. Böyle olunca da fonetik olarak farklı ekleri alan sözcük kökleri farklı kategorilerde yer aldılar. Örneğin *kitap* ve *kalem* sözcükleri isim olmasına karşın kitap sözcüğü *lar*, *ı*, *a* gibi ekleri alabilirken kalem sözcüğü ise aynı görevdeki *ler*, *i* ve *e* eklerini alabilir. Bu da kullanılan algoritma açısında bakıldığında bu iki sözcüğün farklı kategorilerde yer almasına sebep olmaktadır.

Yukarıda anlatılan durumu aşabilmek için sesli harfler alafonlarıyla beraber ortak bir ifade ile yer değiştirdiler [36]. *a* ve *e A* ile gösterildi. Aşağıda görüldüğü gibi bu şekilde bir değişiklik yapıldığında *lar* ve *ler* eklerinin yeni görünümü *lAr* oldu. Bu da bu iki ek arasındaki fonetik farklılığı ortadan kaldırarak iki ekin birbirinin aynısı olarak ele alınmasını sağladı. *kitap* ve *kalem* sözcükleri için yukarıda verilen örneğe göre aldıkları ekler ortak olarak, *lAr* oldu. Bu işlem sadece eklere uygulandı. Ve daha önce anlatıldığı şekilde önce MI tabanlı yöntem ardından ise ıraksama metriği kullanan yöntem ile sözcükler türlerine ayrıştırıldı ve sonuçta 33 biricik kategori oluştu (bkz. Çizelge 3.10.).

3.4. Ek Kategorilerinin Bulunması

Bu bölümde birbirleriyle aynı göreve ve anlama sahip fakat ses olarak farklı olan alomorflar kümelenmeye çalışıldı. Türkçe ekler tamamen aynı işleve sahip olmalarına karşın harf olarak farklı görünümlerde kullanılabilirler. Bunun sebebi de daha önce de anlatıldığı üzere sesli ve sessiz harf uyumlarına göre çeşitlilik göstermeleridir. Örneğin ünlü uyumundan dolayı çoğul eki *ler* ve *lar* olarak iki farklı şekilde sözcüklere eklenir. İlgi eki olan *CI* ise hem ünlü uyumu hem de ünsüz uyumundan ötürü *cı*, *ci*, *cu*, *cü*, *çı*, *çi*, *çu* ve *çü* olmak üzere tam sekiz farklı şekilde sözcüklere ek olarak eklenmektedir. Alomorfların bulunması için önce sesli harfler için bir kümeleme işlemi daha sonra da sessiz harfler için kümeleme yapıldı ve ekler bu özellikleriyle beraber kategorize edilmiş oldu.

Sesli Allofanların Kümelenmesi

Sesli allofanları bulabilmek için eklerin kendilerinden sonra gelen ekleri sesli harflerinden arındırıldı. Ve daha sonra bu ekler üzerindeki olasılık dağılımları kullanılarak kümelendirilmeye çalışıldı:

ir: {d(i), l(e)r, s(e), m(i)ş, vb.}
ır: {d(ı), l(a)r, s(a), m(ı)ş, vb.}

ir ve *ır* ekleri (alomorfları) birbirleriyle benzer ekler (sesli harfleri çıkartılmış) almıştır. Bu sesli harfleri çıkartılmış ekler üzerinde bir Multinomial-Dirichlet dağılımı tanımlanmıştır. Bu tanımlama tüm ekler için teker teker yapılmıştır. Sesli harflerinden arındırılmış ekler üzerindeki θ parametresiyle tanımlanan Multinomial dağılım $M_{fol} = \{m_1, \ldots, m_N\}$:

$$m_i|\theta \sim Multinomial(\theta) \tag{8}$$

Ve önsel (prior) olasılık dağılımı, β hiperparametreleriyle beraber Dirichlet dağılımı tarafından aşağıdaki gibi tanımlanmıştır:

$$\theta|\beta \sim Dirichlet(\beta) \tag{9}$$

Sesli harfleri çıkartılmış bu ekler üzerindeki birleşik olasılığı elde etmek için θ integralde yok edildi:

$$p(M_{fol}|\beta) = \frac{\Gamma(B)}{\Gamma(N+B)} \prod_{i=1}^{K} \frac{\Gamma(n_k + \beta_k)}{\Gamma(\beta_k)} \tag{10}$$

Burada K alomorf küme sayısı, $B = \sum_k^K \beta_k$ ve $N = \sum_k^K n_k$. Her küme için simetrik hiperparametreler kullanılmıştır.

Sessiz Allofanların Kümelenmesi

Sessiz allofanları kümelemek için sesli harflerden arındırılmış önceki ekler üzerindeki olasılık dağılımları kullanılmıştır. Örneğin *lik* ve *liğ* alomorf, *k* ve *ğ* harfleri de allofanlardır:

lik {c(i), l(i), (i)ş, s(i)z, (i)c(i), vb.}
liğ {c(i), l(i), (i)c(i), vb. }

Burada *lik* ve *liğ* eklerinin kendilerinden önce gelen ve sesli harflerinden arındırılmış eklerine bakıldığında benzer bir dağılım gösterdikleri görülmektedir. Benzer olarak, bu sesli harfleri çıkartılmış ekler üzerinde bir Multinomial-Dirichlet dağılımı tanımlandı ve bu işlem tüm kümelendirilmeye çalışılan ekler için uygulandı. Sesli harflerinden arındırılmış ekler üzerinde tanımlanan Multinomial-Dirichlet dağılımı $M_{pre} = \{s_1, \ldots, s_N\}$:

$$\begin{aligned} s_i|\gamma &\sim Multinomial(\gamma) \\ \gamma|\alpha &\sim Dirichlet(\alpha) \end{aligned} \tag{11}$$

Sesli harfleri çıkartılmış bu ekler üzerindeki bileşik olasılığı elde etmek için integrali alınır:

$$p(M_{pre}|\alpha) = \frac{\Gamma(A)}{\Gamma(M+A)} \prod_{i=1}^{L} \frac{\Gamma(n_k + \alpha_k)}{\Gamma(\alpha_k)} \tag{12}$$

Burada L alomorf küme sayısı, $A = \sum_l^L \alpha_l$ ve $M = \sum_l^L n_l$. Burada da benzer olarak, her küme için simetrik hiperparametreler kullanıldı.

Sonuç olarak elde edilen bazı ek kategorileri Çizelge 3.11.'de görülmektedir.

ler, lar
cik, cığ, cık, cük, cüğ
luk, lık, liğ, lüğ, lığ, lik, lük, luğ
dükçe, dikçe, dıkça, dukça
tüğ, tuk, tik, tuğ, tığ, tiğ, tük, tık
dığ, düğ, dık, diğ, dük, duk, duğ, dik

ÇİZELGE 3.11.: Bazı ek kategorileri

3.5. Harf İkilileri

N-gram, terim olarak ard arda gelen elemanlar olarak ifade edilebilir. Bigram, n-gram terimindeki, $n = 2$ olduğu durumdur. Birçok doğal dil işleme konusunda kullanılan bigram, Türkçe ikili olarak çevrilebilir. Bu ikililer harfler veya sözcükler olabilir. Bu çalışmada da harf ikilileri kullanıldı. Ses kurallarını istatistiksel olarak tespit edebilmek için ikililer kullanıldı. Sözcük türetme işlemi yapılırken ek seçimi sırasında doğru eki seçebilmek için bu ikililerden faydalanılacaktır.

Çalışma kapsamında 3 tane ikili gurubu oluşturuldu. Bunlardan birincisi sesli uyumunu ele alabilmek için oluşturuldu. Bu ikili grubu oluşturulurken kökten eke ve ekten eke olan geçişlerdeki harflere bakıldı. Sözcüğün son sesli harfi ile bir sonraki ekin ilk sesli harfi ile birlikte bir ikili oluşturuldu.

Örneğin *kalem* + *in* sözcüğünden çıkarılan ikili *e* ve *i* oldu. İfade edilirken de *e:i* olarak, sıra önemli bir şekilde yazıldı. Başka bir örnekte ise *koş* + *acak* sözcüğünden üretilen sesli harf ikilisi *o:a* şeklinde oldu. Buradaki *a* harfinin *acak* ekindeki ilk *a* harfi olduğu unutulmamalıdır.

Diğer bir ikili ise sessiz harf uyumu için kullanıldı. Türkçede sessiz harf benzeşmesi olarak adlandırılan kural çerçevesinde sert sessizle biten bir sözcük yine sert bir sessiz harf ile

p:t - 0.00117	p:d - 0
s:t - 0.000584	s:d - 0
a:d - 0.00878	a:t - 0.0018
e:k - 0.00369	e:ğ - 0
e:i - 0.12	a:i - 0.0053
u:u - 0.05	u:ü - 0.000113
a:a - 0.075	a:e - 0.0022

ÇİZELGE 3.12.: Bazı harf ikililier ve onların görülme olaslıkları

başlayan bir ek alabilir. Bunun tersi de doğrudur. Örneğin *kitap* sözcüğünün son harfi olan *p* harfi sert sessiz bir harftir. Bu sebepten ötürü yönelme hal eki olan *tan* ekini alabilirken, dengi olan *dan* ekini alamaz. *t* harfi sert bir ünsüzken, *d* harfi değildir. Bu gibi durumları ele alabilmek için yine geçişlerdeki ilk ve son harf arasında ikililer oluşturuldu.

Örneğin *kitap* + *lar* + *dan*, eklerine ayrılmış sözcüğünden üretilen ikililer, *p:l* ve *r:d* olacaktır. Bu şekilde ünlü benzeşmesi ve benzeri ses olay ve kuralları ele alınabilmektedir.

Diğer ikili grubu ise son harf ikilisidir. Bu grupta ise sözcüklerin son harfleri dikkate alınmıştır. Örneğin *kitap+lar+dan* eklerine ayrılmış sözcüğünden üretilen son harf ikilisi *n:$* olacaktır. *$* karakteri sözcüğün sonu anlamında kullanılmıştır. Burada son harfi sözcük sonunda yer alamayacak eklerin, sözcük sonuna gelmesi önlenmektir. Örneğin son harfinde yumuşamaya uğramış *lığ* ekinin sözcük sonunda yer almaması bu ikililer sayesinde sağlanabilmektedir.

Bazı eklerine ayrılmış sözcükler ve elde edilen harf ikilileri:

kita**p + ta + dır** $->$ p:t (son-ilk harf); a:a ve a:ı (son-ilk sesli harf); r:$ (son harf)
kale**m + ler** $->$ m:l (son-ilk harf); e:e (son-ilk sesli harf); r:$ (son harf)
gi**t + miş + ti** $->$ t:m, ş:t (son-ilk harf); i:i (son-ilk sesli harf); i:$ (son harf)

Bütün bu ikili grupları görülme sıklıkları, frekanslarıyla beraber kaydedildi. Daha sonra seçim yapılırken en yüksek olasılığa sahip ikiliye uygun ek seçildi (bkz. Çizelge 3.12.).

Bu ikililer kullanılarak ek seçimi yapılırken ilk olarak ikinci ikili grubu olan son - ilk harf ikilileri kullanılır. Burada da son harfin sesli, ilk harfinin sessiz olduğu (VC) ve son ve ilk harfin sessiz olduğu (CC) durumlar dikkate alınır. Son harfin sessiz, ilk harfin ise sesli olduğu durumda (CV) bu ikili grubu kontrolü yapılmadan devam edilir. Son ve ilk harflerin ikisinin

de sesli harf olması durumunda ise seçim yapılmayacak ve bu aşamada sözcük türetilmeyecektir. Bu ikili grubunda frekansı en yüksek olan ekler bir sonraki aşamada elenmek üzere seçilir. Son - ilk harf ikilileri ile yapılan örnek seçimler şu şekilde olacaktır:

1. kap**ı**, Ek kümesi: **d**an, **t**an, **d**en, **t**en $->$ Seçilen ek(ler): **d**an, **d**en
2. çiçekli**k**, Ek kümesi: **ç**i, **c**i, **ç**ı, **c**ı, **ç**ü, **c**ü, **ç**u, **c**u $->$ Seçilen ek(ler): **ç**i, **ç**ı, **ç**ü, **ç**u
3. so**r**, Ek kümesi: **i**, **ı**, **u**, **ü** $->$ Seçilen ek(ler): **i**, **ı**, **u**, **ü** (Eleme yapılmadan devam edildi)
4. sepe**t**, Ek kümesi: **l**ik, **l**ık, **l**iğ, **l**ığ $->$ Seçilen ek(ler): **l**ik, **l**ık, **l**iğ, **l**ığ (Eleme yapıldı, sonunda hiçbir ek elenmedi)
5. ok**u**, Ek kümesi: **a**, **e** $->$ Ek seçimi yapılamadı ve bu durumda sözcük türetilmedi. FSA'da bu durumdan sonrasına ilerlenmedi.

Daha sonra kalan ekler ile sözcük arasında sesli harf uyumuna bakılır. Yine en yüksek frekanstaki ekler seçilir. Sesli uyumunu elen alan ikililer kullanılarak yapılan seçim şu şekilde devam edecektir:

1. kap**ı**, Ek kümesi: d**a**n, d**e**n $->$ Seçilen ek(ler): d**a**n
2. çiçekl**i**k, Ek kümesi: ç**i**, ç**ı**, ç**ü**, ç**u** $->$ Seçilen ek(ler): ç**i**
3. s**o**r, Ek kümesi: i, ı, u, ü $->$ Seçilen ek(ler): **u**
4. sep**e**t, Ek kümesi: l**i**k, l**ı**k, l**i**ğ, l**ı**ğ $->$ Seçilen ek(ler): l**i**k, l**i**ğ

Bu durumda çoğunlukla tek bir ek kalmış olacaktır. Örnek 4'te de olduğu gibi, birden fazla ek kalmışsa iki ek ile de sözcük türetilmeye devam edilecektir. Son olarak sözcük türetilirken üçüncü ikili grubu tarafından frekansı en yüksek ek seçilir. Üçüncü ikili grubuna bakıldığında sözcük sonuna gelecek harfin frekansı 0 olarak bulunmuşsa o sözcük türetilmez. Son harf ikilileri ile yapılan seçim, örnek 4 için şu şekilde gerçekleşir:

4. sepet, Ek kümesi: li**k**, li**ğ** $->$ Seçilen ek(ler): li**k**

Harf ikilileri kullanılarak yapılan ek seçimi sonucunda üretilen sözcükler aşağıdadır:

1. kapıdan
2. çiçeklikçi
3. soru
4. sepetlik

3.6. Ortografik (Yazımsal) Kurallar

Harflerin düşmesi, yeni bir harf türetilmesi veya harflerin başka bir harfe dönüşmesi gibi ses olayları bu bölümde ele alındı. Örneğin *kitap* sözcüğü *ın* eki aldığında yeni oluşan sözcük *kitabın* oldu. Görüldüğü üzere burada *p* harfi *b* harfine dönüşmüştür. Çalışmada bu ve benzer değişimleri ele alabilmek için, yazımsal kurallar bulunmaya çalışıldı.

Bu kurallar Oflazer ve arkadaşları [69] çalışmasına benzer olarak aşağıdaki gibi tanımlanmıştır.

$$\text{i} -> \text{s} \;||\; \text{Sol bağlam} \_ \text{Sağ bağlam}$$

Burada i ilk durumu s son duruma, sağ bağlam sol bağlamdan hemen önce gediğinde dönüşür. Yukarıdaki tanım *kitabın* sözcüğü için aşağıdaki gibi gösterilir. Tanımın okunuşu, *p* harfinden sonra *ı* harfi ile başlayan ek gelirse *p* harfi *b* harfine dönüşür şeklinde olacaktır. Gösterimde harf düşmesi durumunda ise, düşen harf yerine 0 yazılarak gösterilecektir. Bir takım aday kurallar aşağıda verilmiştir:

kita**p** + ın= kita**b**ın : p $->$ b || p _ ı

çiçe**k** + i= çiçe**ğ**i : k $->$ ğ || k _ i

al**ı**n + ı= alnı : ı $->$ 0 || n _ ı

Çalışmada bu tarz kurallar otomatik olarak bulunmaya çalışıldı. Veri kümesindeki sözcüklerin ilk halleri ile Zemberek yardımıyla eklerine ayrılmış halleri yardımıyla yeni sözcük oluşumundaki yazımsal kurallar bulunmaya çalışıldı. Bir sözcük incelenirken tespit edilen ses olayının hangi koşullarda meydana geldiği başta bilinmemektedir. Bu yüzden öncelikle

aday kurallar belirlendi. Oluşan aday kurallar, yüzeysel (surface) form *kitabın* karşılığı olan *kitap+ın* sözlüksel (lexical) formu için aşağıdaki gibi tanımlandı:

p − > b || p _ ı
p − > b || ap _ ı
p − > b || p _ ın
p − > b || ap _ ın

Burada sözcük kökünün sonu için 2 karakterlik sınır koyuldu. Yani yukarıda sol bağlam olarak *tap*, *itap* vb. bir örnek oluşturulmadı. Sağ bağlam için herhangi bir sınır koyulmadı.

İlk olarak aday kurallar çıkartıldı. Oluşan aday kurallar sayıldı ve bu sayı kuralların skoru olarak saklandı. Daha sonrasında kurallar kategorilerine ayrıldı. Kategoriler ses olaylarına göre belirlendi. Örneğin *p* harfinin *b* harfine dönüşmesi olayını ele alan tüm kurallar bir yerde toplandı. Her kategori içerisinde skoru yani görülme sayısı en yüksek olan kural, son kural olarak seçildi. Aynı skora sahip kuralların aynı kategoride bulunması durumunda ise en genel kural, son kural olarak seçildi. Genel kural seçimi yapılırken de sağ bağlamı en geniş olan yani daha çok karakter barındıran kural tespit edildi. Sonuç olarak halen daha birden fazla kural varsa bu sefer de sol bağlamı daha geniş olan kural son kural olarak seçildi. Yukarıdaki aday kurallardaki son kural aralarında en genel kural olarak tespit edildi.

Son kuralların seçilmesinden sonra bu kurallar genelleştirildi. Örneğin sesli harflerden sonra değişim meydana geldiyse kuraldaki bağlam ona uygun olarak tüm sesli harfler olarak genişletildi. Sonuç olarak oluşan son yazımsal kurallar aşağıdaki gibi oluşmuştur.

Son yazımsal kurallar
k − > ğ || "nk" _ "i"
k − > ğ || "nk" _ "e"
t − > d || "t" _ Vowel
k − > ğ || "k" _ Vowel
ç − > c || "ç" _ Vowel
p − > p || "p" _ Vowel
a − > 0 || "a" _ "Uyor"
e − > 0 || "e" _ "iyor"

ÇİZELGE 3.13.: Son kurallar kümesi

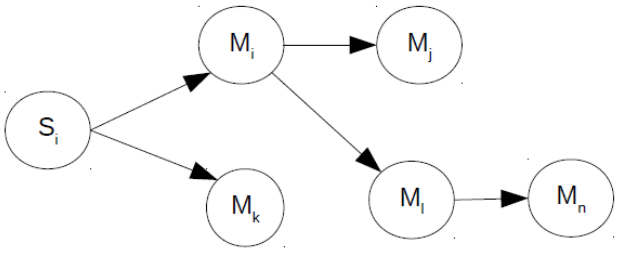
3.7. Sözcük Türetme

Sözcük türetme modeli için gerekli olan, sözcük kök kategorileri, ek kategorileri, harf ikilileri ve yazımsal kurallar oluşturuldu. Bu aşamadan sonra sonlu durum özdevinirleri ile sözcük türetme işlemi yapılacaktır.

FSA Kurma

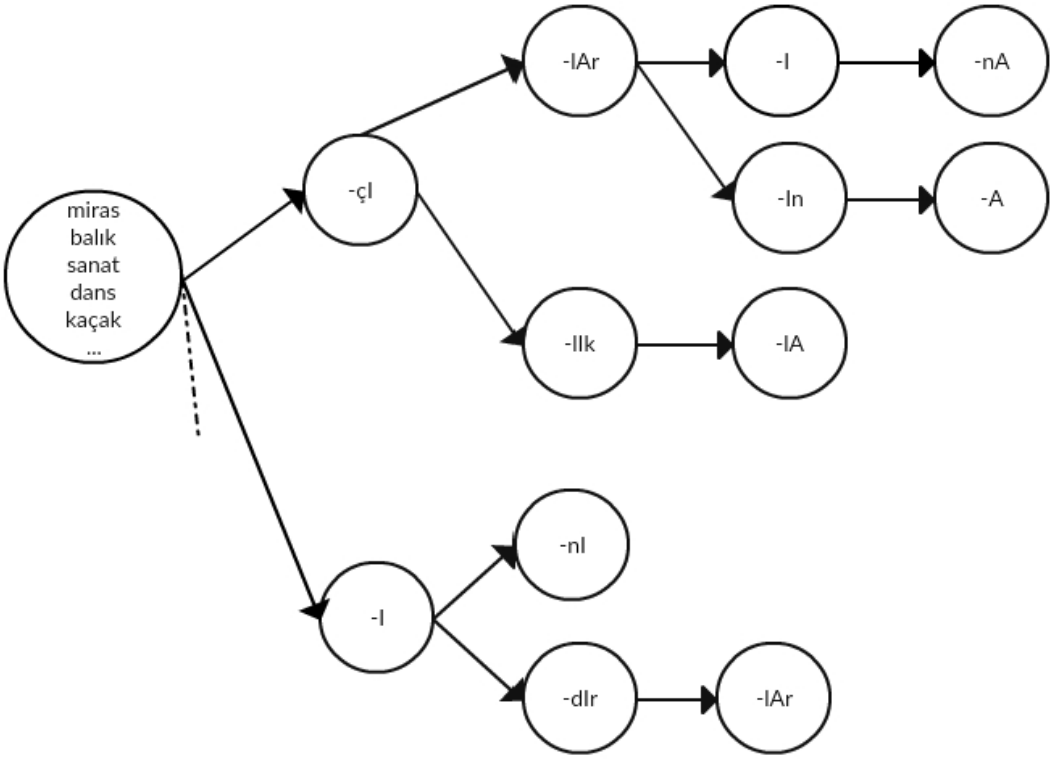
Kök ve ek kümeleri kullanılarak FSA'lar oluşturuldu. FSA'daki her bir durum, bir kök kategorisi S_i veya bir ek kategorisi M_i ile temsil edilmektedir. Her kök kategorisi için bir FSA tasarlandı. Bu kök kategorileri oluşan FSA'ların başlangıç durumları olarak atandı. Kök durumu kendisinden sonra çeşitli ek durumları ile bağlandı. Son olarak ek kategorileri de kendilerinden sonra gelen diğer ek durumlarına bağlandı. Bir sonlu durum özdeviniri 5 katmanlı olarak tanımlanır:

$$X = (A, I, f, S_0, F)$$



ŞEKIL 3.3.: Örnek bir sonlu durum özdeviniri. S_i bir sözcük kökü kategorisini temsil ederken, M_i, M_j, M_k, M_l ve M_n bir ek kategorisini temsil etmektedir.

- A durumlar kümesi, A, kök kategorisi S_i veya ek kategorisi M_i'ye ait olabilir.
- I alfabe, $I = M \cup S$ (M tüm ekler, S tüm kökler)
- f geçiş fonksiyonu, S_1'den diğer bir durum M_1'e veya M_1'den diğer bir durum M_2'ye



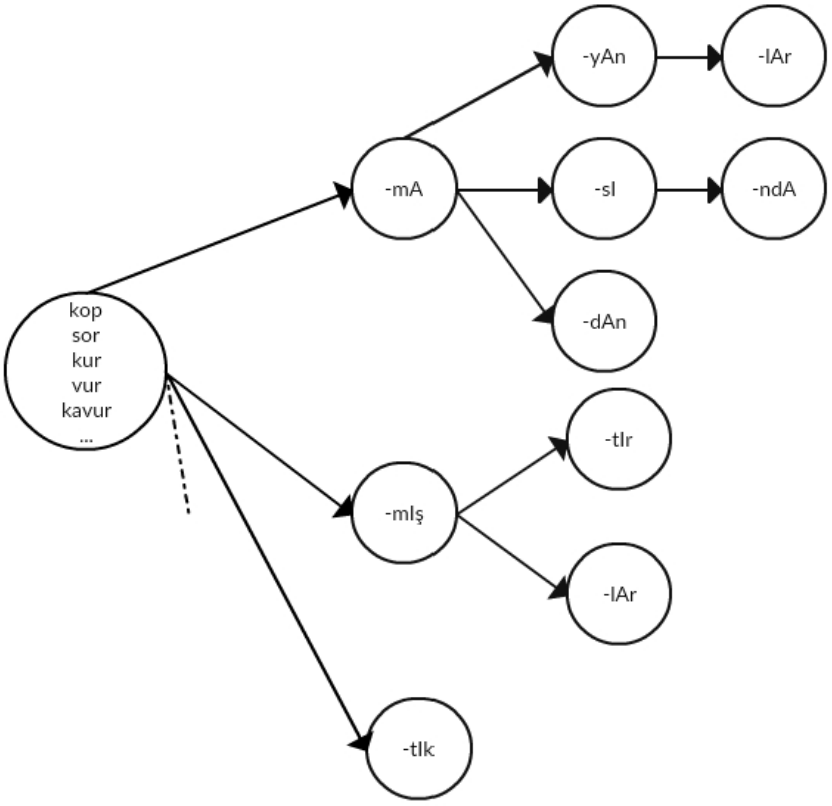
ŞEKIL 3.4.: İsimler için oluşturulmuş sonlu durum özdevinirinden bir kesit. Burada *a* ve *e*, *A* olarak ve *ı* ve *i* ise *I* olarak temsil edilmiştir

- S_0 ilk durum, A'nın bir elamanı başlangıç durumu
- F ise A'nın bir alt kümesi, son durumların kümesi. Kök kategorisi S_1 veya ek kategorisi M_1 F'nin bir elamanı olabilir.

İsimler için oluşturulmuş bir FSA'dan kesit Şekil 3.4.'te ve eylemler (fiiller) için oluşturulmuş bir FSA'dan kesit ise Şekil 3.5.'te verilmiştir.

FSA Kullanılarak Sözcük Türetilmesi

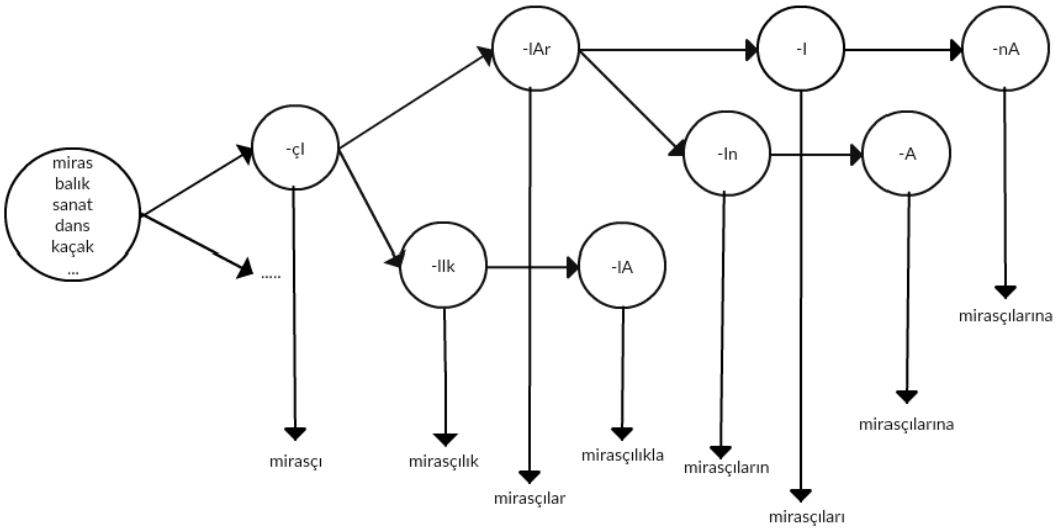
Tüm FSA'lar kurulduktan sonra her bir FSA derin öncelikli arama algoritmasına uygun olarak gezildi. Gezilen her durum sonunda bir sözcük türetildi. Kök durumlarından ek almamış tek bir kök oluşurken, ek durumlarında ise ek almış, kök + ek(ler) formunda yeni bir sözcük



ŞEKIL 3.5.: Eylemler için oluşturulmuş sonlu durum özdevinirinden bir kesit. Burada *a* ve *e*, *A* olarak ve *ı* ve *i* ise *I* olarak temsil edilmiştir

türetildi. Üretilen sözcükler aldığı eklerle beraber bir sonraki duruma aktarıldı. Yani gelen ekler birbiri ardına gelerek yeni sözcüklerin türetilmesine devam edildi. FSA'daki gezme işlemi kök durumda yer alan kök kategorisindeki tüm sözcükler için teker teker uygulandı. Ek durumları için ise her geçişte uygun bir ek seçimi yapılmaya çalışıldı. Şekil 3.4.'te FSA kesitinde yer alan *miras* sözcük kökü için üretilmiş sözcük örnekleri Şekil 3.6.'da verildi.

FSA'daki ek durumlarında ek seçimi yapılırken, harf ikili olasılığı en yüksek ek seçilir. Eğer tek bir ek seçilemezse, yani aynı olasılığa sahip birden fazla ek seçimi yapılırsa sözcük türetmek için bir sonraki duruma birden fazla sözcük ile devam edilir. Bu birden fazla seçilen ekten sözcük sonunda yer almaya uygun olan ekler son harf ikilileri ile tespit edilerek, türetilen sözcükler arasına aktarılır. Eğer tek bir ek seçilmişse de FSA'daki bir sonraki duruma bu ek ile türetilen sözcük ile devam edilir. Seçilen tek ek yine son harf ikili olasılığına göre karşılaştırılır ve sözcük sonunda yer almayan bir harf barındırmışsa, yani son harf ikili olasılığı 0 olarak belirlenmişse yeni sözcük türetilmeden bir sonraki duruma geçilir. Örneğin



ŞEKIL 3.6.: Şekil 3.4.'deki sonlu durum özdevinirinden, *miras* sözcüğü için üretilen sözcüklerden bir parça.

Türkçede *ğ* harfi sözcük sonunda yer almaz. Hesaplanan olasılığın da 0 olaması beklenir.

Harf ikili olasılıklarına göre ek seçimi yapıldıktan sonra yazımsal kurallar uyguldı. Gerekli bir değişiklik varsa yapıldı ve yeni sözcük bu şekilde türetildi. Örneğin, Şekil 3.4.'te, *balık* sözcüğünden sonra *ı* ekinin seçimi yapıldıysa yazımsal kurallar uygulandıktan sonra yeni üretilen sözcük *balığı* oldu.

3.8. Sözcük Türetme Deney ve Sonuçları

Çalışma boyunca öncelikle sözcük kökleri kategorilerine ayrıldı. Daha sonra ekler benzer şekilde kümelendi. Oluşan kök ve ek kategorileri sonlu durum özdevinirlerindeki durumlara yukarıda anlatıldığı gibi yerleştirildi. Durumlar arası geçişlerde ek seçimlerinde harf ikilileri kullanıldı. Ardından da yeni oluşan sözcükler üzerindeki muhtemel değişiklikler yazımsal kurallar ile ele alındı. Bütün bu adımlardan sonra yukarıdaki algoritma ile yeni sözcükler türetildi. Burada ise yapılan sözcük türetme çalışmaları ve bu çalışmaların sonuçlarından bahsedilecektir.

İlk olarak sözcük köklerinin kategorilerine ayrılması sonucunda üretilen kategorilerle yapılan sözcük türetme çalışması ve sonuçları verilmiştir. Sözcük türetme sonuçlarının başarısı doğruluk (accuracy) oranıyla ifade edilmiştir. Doğruluk, bu çalışma kapsamında, sonuç olarak

Algorithm 1 Sözcük türetme algoritması

Girdi: tüm FSA'lar $\mathbf{X} = \{X_1, X_2, \dots\}$, harf ikilileri, yazımsal kurallar

for each: X^i in **X do**
  for each: $s_i \in S_0^i$ **do**
    $w \leftarrow s_i$
    $A_j^i \leftarrow S_0^i$
    while A_j^i takip eden durum **do**
      bir sonraki duruma git A_{j+1}^i
      if bir ek m_i, A_{j+1}^i içerisinden seçildi **then**
        $w \leftarrow w + m_i$
        w için yazımsal kuralları uygula (eğer varsa)
      else
        break
      end if
      $A_j^i \leftarrow A_{j+1}^i$
    end while
    Ekle w, **W** içerisinden
  end for
end for

Çıktı: W

üretilen sözcüklerin doğru olanlarının üretilen tüm sözcüklere oranıdır. Üretilen sözcüklerin doğru olup olmadığına Zemberek [21] yardımıyla karar verilmiştir.

$$Accuracy = \frac{|D|}{|T|} \tag{13}$$

Burada D doğru olarak türetilen sözcük kümesini, T ise üretilen tüm sözcük kümesini ifade eder ve D, T'nin alt kümesidir ($D \subseteq T$). En iyi durumda ise yani üretilen tüm sözcüklerin doğru olması durumunda bu iki küme birbirinin aynısı olacaktır.

Kökler kümelenirken kullanılan ilk yöntem olarak karşılıklı bilgi (MI) tabanlı yöntem incelendi. Sözcük kökü sayısı 100, 200, 500, 1000, 2000 ve son sözcük kökü sayısı olan 3049 belirlendi. Bu sayılar belirlenirken kök kümeleme işlemi bu sayılara ulaşıldığında sonlandı. Bu durum göz önüne alındığında artan sözcük kökü sayısının kümülatif olarak arttığı ve bir üst veri kümesinin bir alttaki sözcük köklerinin hepsini içerdiği belirtilmelidir. Çizelge 3.14.'te her kök sayısı için üretilen kategori sayısı, üretilen sözcük sayısı ve doğruluk oranları verilmiştir.

Kök sayısı	Kök kategorisi sayısı	Üretilen sözcük sayısı	Doğruluk
100	30	7.491	%76.03
200	45	13.093	%72.67
500	75	37.041	%78.59
1000	95	80.569	%76.14
2000	114	172.881	%77.65
3049	152	237.219	**%76.79**

ÇİZELGE 3.14.: Karşılıklı bilgi tabanlı metrik ile oluşturulan sözcük kökü kategorileri ve onlarla üretilen sözcükler

Kök sayısı	Kök kategorisi sayısı	Üretilen sözcük sayısı	Doğruluk
100	24	10.737	%76.07
200	30	26.017	%73.11
500	44	104.745	%77.47
1000	53	243.097	%76.14
2000	68	527.170	%75.50
3049	89	915.288	**%74.02**

ÇİZELGE 3.15.: Jensen-Shannon ıraksama metriği ile birleştirilmeye devam eden kök kategorileri ve üretilen sözcükler

Kök sayısı	Kök kategorisi sayısı	Üretilen sözcük sayısı	Doğruluk
3049	33	1.516.769	**%73.19**

ÇİZELGE 3.16.: Son olarak elde edilen kök kategorileri ve üretilen sözcükler

Çalışmada karşılıklı bilgi tabanlı metrik ile bulunan kategoriler, Jensen-Shannon ıraksama metriği kullanılarak birleştirilmeye devam edildi. Sonuçlar Çizelge 3.15.'te verilmiştir.

Çalışmada bu iki aşama sesli harflerin allofanlarıyla birlikte ortak bir karakter ile ifade edilmesinden sonra tekrar edildi. Özetle allofanlarla ilgili değişiklikler yapıldıktan sonra, önce karşılıklı bilgi metriği daha sonra da JS ıraksama metriği ile sözcük kökleri kategorize edildi. Bu aşamada ise sadece son veri kümesi (3049 biricik sözcük kökü) ile çalışıldı. Sonuçlar Çizelge 3.16.'da verilmiştir.

Yukarıdaki sonuçları değerlendirirken üretilen sözcük sayısı ve doğruluk oranları dikkate alınmıştır. Karşılıklı bilgi tek başında kullanıldığında son durumda $\%76.79$'luk bir doğruluk oranı sağlamıştır. Ama ürettiği 237.219 sözcük diğer yöntemlere bakıldığında oldukça az kalmıştır. Karşılıklı bilgi metriğine ek olarak ıraksama(divergence) benzerlik metriği ile yapılan çalışmada doğruluk oranında küçük bir düşüş yaşanmıştır ($\%74.02$). Fakat oluşan sözcük sayısı yaklaşık 4.5 kat artarak 915.288 olmuştur. Son olarak uygulanan yöntemde ise

Yöntem	Pencere	Kök kategorisi sayısı	Üretilen sözcük sayısı	Doğruluk
CBOW	4	47	641.255	%70.76
CBOW	5	46	628.082	%70.31
skip-gram	4	50	500.308	**%76.07**
skip-gram	5	50	528.227	%74.52

ÇİZELGE 3.17.: word2vec modeli ile elde edilen kök kategorileriyle üretilen sözcükler

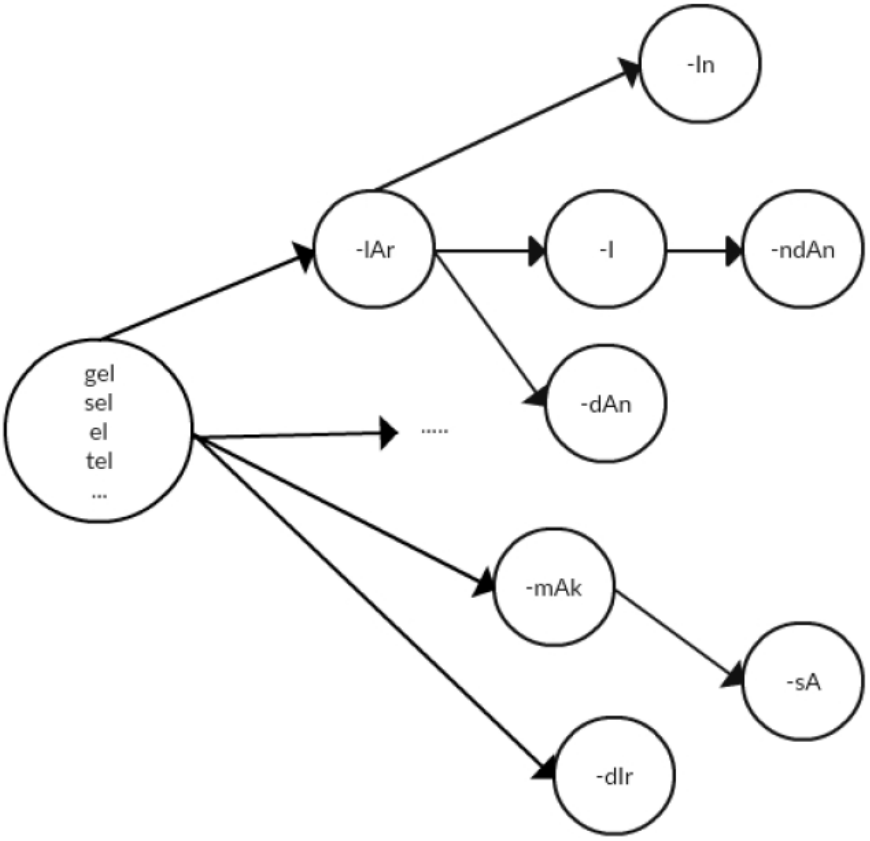
$\%73.19$ doğruluk oranıyla birlikte üretilen $1.516.769$ sözcük ile en yüksek başarı oranı sağlandığı söylenebilir. Bu üç sonuç karşılaştırıldığında kullanılan sözcük kök kategori sayısı ile üretilen sözcük arasında ters orantı olduğu gözlemlenmiştir. Aynı şekilde üretilen sözcük sayısı arttığında doğruluk oranlarında da düşüş yaşanmıştır. Fakat üretilen sözcük sayısındaki artışın, düşen doğruluk oranına göre daha büyük olduğu söylenebilir.

Bu çalışmalara ek olarak word2vec modeli ile üretilen sözcük kök kategorileriyle de sözcük türetme çalışmaları yapılmıştır. Çizelge 3.7. ve 3.8.'de elde edilen kök kategorileriyle gerçekleştirilen sözcük türetme sonuçları 3.17.'de verilmiştir. Sonuçlar son veri kümesi ile elde edilmiştir.

word2vec modeli kullanılarak elde edilen kategorilerle üretilen sözcüklere bakıldığında skip-gram mimarisinin daha yüksek doğruluk oranı elde ettiği gözlemlenmiştir. En yüksek doğruluk oranı $\%76.07$ ile pencere boyutunun 4 olarak seçildiği skip-gram mimarisi ile elde edilmiştir. Fakat daha yüksek bir sözcük sayısına ulaşan model son sözcük türetme modeli olarak seçilmiştir.

Tüm bu çalışmalar sonucunda elde edilen doğruluk oranları yaklaşık olarak %73 ile %76 arasında değiştiği gözlemlenmiştir. Bu sonuçları artırabilmek için ek bir çalışma daha yapılmıştır.

Son yöntem olarak kabul ettiğimiz modelde yaklaşık bir buçuk milyon sözcük üretilmiş ve $\%73.19$'luk bir doğruluk oranı yakalanmıştır. Üretilen sözcükler detaylı olarak incelenmiş ve oluşan gürültüler tespit edilmeye çalışılmıştır. Algoritma gereği başlangıç durumunda bulunan her bir sözcük kökü için, FSA bir kez gezilerek sözcükler türetilmiştir. Başlangıç durumunda yer alan her bir kökün aldığı ekler birleştirilerek FSA oluşturulmuştur. Bu aşamada bazı yanlış veya eksik kategorize edilmiş köklerin aldığı ekler FSA'da istenmeyen yolların oluşmasına neden olmuştur. Bir FSA içerisinde tek bir kök, birçok yanlış sözcüğün üretilmesine sebep olmuştur.



ŞEKIL 3.7.: Hatalı olarak oluşturulmuş sonlu durum özdevinirinden bir kesit. Burada *a* ve *e*, *A* olarak ve *ı* ve *i* ise *I* olarak temsil edilmiştir

Şekil 3.7.'de hatalı bir sözcük kategorisi kökü ile oluşturulmuş bir FSA örneği görülmektedir. Burada *el*, *sel* ve *tel* isim kökler iken *gel* sözcüğünün türü fiildir. *Gel* sözcüğü yanlış bir şekilde bu kategori içerinde yer almıştır. FSA incelendiğinde *mAk* (*mak*, *mek*) mastar ekinin durumlar içerinde yer aldığı görülmektedir. Mastar ekleri sadece fiiller tarafından alınabilen bir ek türüdür. Burada *gel* sözcüğünden ötürü FSA içerinde yer aldığı kolayca tahmin edilebilir. *mAk* durumunun FSA'da yer alması, başlangıç durumundaki her kökün bu durumdan ilerleyerek sözcük türetmesi anlamına gelmektedir. Sonuç olarak *selmek*, *telmekse* gibi yanlış sözcüklerin üretilmesi söz konusudur. Buna ek olarak *gel* sözcüğünden üretilen *geller*, *geldir*, *gellerden* gibi yanlış sözcükler de elde edilir.

Bu durumu engellemek için az sayıda görülen eklerin yer aldığı düğümler sözcük üretilme aşamasında ihmal edildiler. Şekil 3.7.'teki örnekte *mAk* eki diğer sözcük kökleri tarafından alınmadığı için, başka bir deyişle sadece tek bir kök tarafından alındığı için, bu ekin yer aldığı

Az görülme limiti	Üretilen sözcük sayısı	Doğruluk
1	845.467	%80.11
2	744.061	**%82.36**

ÇİZELGE 3.18.: Az görülen ekleri içeren durumların ihmal edildiği sözcük türetme sonuçları.

durum ihmal edilecektir. Bu da diğer sözcük köklerinin bu yanlış yoldan ilerlemesini engelleyerek birçok yanlış sözcüğün üretilmesini önleyecektir. Fakat *gel* sözcüğünün de bu yoldan ilerlemesi engelleneceği için, bir takım doğru sözcükler de üretilemeyecektir. Bununla ilgili yapılan deneylerde bir ekin, aynı kategoride yer alan sözcük kökleri tarafından sadece 1 kere alındığı ve 2 veya daha az alındığı iki farklı durum ele alınmıştır. Çizelge 3.18.'de yapılan çalışmalara ilişkin sonuçlar verilmiştir. Az görülme limiti alanındaki 1 sayısı, sadece 1 kere görülen eklere ait durumların ihmal edildiği anlamına gelirken, 2 sayısı ise 2 veya daha az görülen eklere ait durumların ihmal edildiği anlamına gelmektedir.

Son yapılan çalışmanın sonuçlarına bakıldığında az görülme limitinin 2 olarak seçildiği durumda doğruluk oranı %82.36 olarak hesaplanmıştır. Sözcük sayısı son çalışmaya göre yarı yarıya (1.516.769 sözcükten 744.061 sözcüğe düşmüştür) azalmıştır. Fakat doğruluk oranında yaklaşık olarak %10'luk (%73.19'dan %82.36'ya) bir artış meydana gelmiştir. Sözcük sayısındaki düşüşe rağmen doğruluk oranındaki bu artış, son durumda üretilmeyen sözcüklerin büyük bir kısmının yanlış sözcüklerden olduğu şeklinde yorumlanmıştır. Üretilen sözcük örnekleri Çizelge 3.19.'da verilmiştir.

Doğru üretilmiş sözcükler	Yanlış üretilmiş sözcükler
bağımsız	indirimindeyi
bağımsızlık	indirimlerindeyi
bağımsızlığa	indirimki
indirilebilir	indirti
indirilebilirlik	indirtiler
indirilebilirliği	giyilebilerek
ürettik	giyemeyereksiniz
ürettikler	eğilimindeyi
ürettiğimiz	eğilimlerindeyi
doktoralı	hamamdıra
doktoralılar	hamamdırım
doktoralılığımızdan	hamamdırıl
doktoralamayacaklarını	toplantıyına
pazarlıklılığımızdan	toplantısa
pazarlıklayamamıştır	sayfadama
konuşmak	sayfadamama
konuşmaktadırlar	sayfata
konuşulabilmektedir	aylıklaşılabilerek
çatışmalarımızın	aylıktırıyor
çatışmamaktadırlar	aylıktırmış

ÇİZELGE 3.19.: Doğru ve yanlış olarak üretilen sözcüklerden örnekler

4. KARŞILAŞTIRMA

Çalışmamızda denetimsiz olarak sözcük türetmeye çalıştık. Konuyla ilgili alan incelendiğinde birçok çalışmanın denetimli olarak gerçekleştirildiği gözlemlenmektedir. Geliştirdiğimiz modelle karşılaştırabileceğimiz tek model Rasooli ve arkadaşlarının çalışmasıdır [2].

Rasooli ve arkadaşları [2], BabelGum adını verdiği sistemini yedi farklı dil ile test etmiştir. Bu dillerden biri de Türkçe'dir. Biz de bu çalışmayı ilk olarak kendi veri kümemiz ile test ederek iki sistemi karşılaştırmaya çalıştık.

Öncelikle Rasooli'nin çalışmasında sadece eklerin birleştirilerek ele alındığı modeli kendi veri kümemiz üzerinde çalıştırdık. Bu modelde ilk ve son ekler, kendi aralarında birleştirilerek, tek bir ek gibi ele alınmaktadır. Bu model ile bizim geliştirdiğimiz sistem arasındaki bir diğer fark ise morfolojik analizdir. Biz çalışmamızda morfolojik analiz işleminin yapıldığını kabul edip bu iş için Zemberek'i [21] kullandık. Rasooli ise çalışmasında bu işlemi Morfessor CAT-MAP (0.9.2) [13] isimli denetimsiz morfolojik ayrıştırıcıyı kullanmıştır. Biz bu iki sistemi karşılaştırırken Rasooli'nin sistemine, Zemberek tarafından morfemlerine ayrılmış sözcüklerden oluşan bir girdi vererek, morfolojik analiz işleminin yapıldığı aşamadan sonrası için çalıştırdık.

Çizelge 4.1.'de Rasooli'nin ve bizim sistemimizin sözcük türetme sonuçları görülmektedir. BabelGum, daha çok sözcük türetmesine karşın doğruluk oranı oldukça düşük çıkmıştır. Bunun en önemli sebeplerinden biri BabelGum'ın son ekleri bir bütün olarak ele almasıdır. Diğer bir problem ise her ekin her sözcük tarafından alınabilir olarak kabul edilmesidir. Rasooli çalışmasında sözcük kökleri arasında bir fark olmadığını ve ileride bir tür işaretleme sisteminin kullanılarak bu eksiğin kapatılabileceğinden bahsetmiştir.

Karşılaştırmada morfemlerine ayrılmış sözcüklerin kullanılmasıyla iki sistem arasında bu analiz işleminden kaynaklı farklılıkların oluşması önlenmiştir.

	Üretilen sözcük sayısı	Doğruluk oranı
Bizim Model	744.061	%82.36
BabelGum	967.909	%19.24

ÇİZELGE 4.1.: Rasooli'nin [2] ve bizim geliştirdiğimiz modellerin sözcük türetme sonuçları

Yapılan ikinci karşılaştırmada ise, geliştirdiğimiz modelde morfolojik çözümleyici olarak Morfessor CAT-MAP [13] kullanıldı. BabelGum varsayılan morfolojik çözümleyici olarak Morfessor CAT-MAP kullanmaktadır. Biz de eşit şartlarda iki modeli karşılaştırabilmek için modelimizde Zemberek yerine Morfessor CAT-MAP kullandık. Sözcük türetme aşamalarında kullanılan, sözcük kök kategorileri, ek kategorileri, harf ikililer, yazımsal kurallar ve sonlu durum özdevinirleri, Morfessor CAT-MAP'in ürettiği morfemlerine ayrılmış sözcükler kullanılarak üretildi. Karşılaştırma yaparken her iki sistemin de Morfessor CAT-MAP'i aynı ayarlarla çalıştırdığından emin olundu. Böylece kendi sistemimiz de tamamen denetimsiz olarak değiştirildi.

Veri kümesi olarak Morpho Challange 2010 Türkçe sözcük listesi (wordlist) kullanıldı ve bu veri kümesinden frekansı yüksek olan (frekansı 50'den fazla) sözcükler seçildi. Toplam sözcük sayısı 26230 oldu. Daha saf, gürültüden uzak bir veri kümesi oluşturmak için böyle bir daraltma işlemine başvuruldu. Kullandığımız veri kümesinden bir kesit Çizelge 4.2.'de verilmiştir.

56 CIkIn
187 CIkabilecek
433 CIkabilir
51 CIkabilmektedir
67 CIkacaGI
84 CIkacaGInI
541 CIkacak
116 CIkacaktIr

ÇİZELGE 4.2.: İkinci karşılaştırma için kullanılan veri kümesinden bir kesit. Satır başlarındaki sayılar frekansları göstermektedir. Türkçe karakterler büyük harf eşlenikleriyle (ç -> C) değiştirilmiştir.

BabelGum, için gerekli olan eğitim (train) ve geliştirme (development) verileri de yine Çizelge 4.2.'de bir kesiti verilen kümeden elde edilmiştir. Frekansı 50'den fazla olan 26230 sözcüğün 13115'i eğitim için, diğer 13115'i ise geliştirme için kullanılmıştır. Eğitim kümesinde 4822610, geliştirme kümesinde ise 4487088 biricik olmayan sözcük (token) vardır. Tür (type) tabanlı ve sözcük (token) tabanlı olmak üzere BabelGum üzerindeki deneyler iki farklı şekilde yapılacaktır.

	Üretilen sözcük sayısı	Doğruluk oranı
Bizim Model	36648	%31.58
BabelGum Basit Model & Tür-tabanlı	982082	%9.47
BabelGum Bigram Model & Tür-tabanlı	960221	%9.63
BabelGum Basit Model & Sözcük-tabanlı	981556	%9.46
BabelGum Bigram Model & Sözcük-tabanlı	880447	%9.41
BabelGum Basit Model Büyük Eğitim	977777	%10.28
BabelGum Bigram Model Büyük Eğitim	925487	%10.54

ÇİZELGE 4.3.: Yapılan ikinci karşılaştırma sonuçları. Her iki model de morfolojik çözümleme için Morfessor CAT-MAP'i kullanmıştır.

Geliştirdiğimiz model, Morfessor CAT-MAP'den elde edilen morfemlerine ayrıştırılmış sözcükler kullanılarak, 965 biricik sözcük kökü ile sözcük türetme işlemine başladı. 44294 toplam frekans ile 381 farklı harf ikilisi üretildi. Ekler arası hiçbir yazımsal kural tespit edilemedi. Bunu sebebi Morfossor CAT-MAP'in bölme işlemi yaparken hiçbir yazımsal değişiklik yapmamış olmasıdır. 965 sözcük kökü kullanılarak 53 kök kategorisi oluşturuldu. Bunun dışında oluşturulan ek kategori sayısı ise 548 oldu. Sözcük türetme işleminden sonra 36648 sözcük %31.58 doğruluk oranıyla türetildi (bkz. Çizelge 4.3.). Kullanılan morfolojik çözümleyicinin başarısının bizim modelimiz üzerindeki etkisi bu doğruluk oranıyla net bir şekilde ortaya koyulmuştur. Denetimsiz bir sistem olan Morfessor CAT-MAP'in, denetimli bir sistem olan Zemberek'e göre morfemlerine ayırma işleminde daha başarısız olması beklenen bir durumdur.

Çizelge 4.3.'te BabelGum sistemi ile gerçekleştirilen deneylerin sonuçlarına yer verilmiştir. Bu çizelgede yer alan BabelGum basit model, eklerin toplu olarak ele alındığı modeldir. Bigram model ise eklerin ayrı ayrı ele alındığı modeldir. Tür tabanlı (type-based) ve sözcük tabanlı (token-based) sonuçlara yer verilmiştir. BabelGum ayrıca daha büyük bir eğitim verisi kullanılarak da test edilmiştir. Daha büyük eğitim verisinin kullanıldığı deneyde 26230 toplam sözcükten 20887 tanesi eğitim verisi içerisinde yer almıştır. Büyük veri kümesinin kullanıldığı deneyler varsayılan tür tabanlıdır. Sonuçlara bakıldığında eğitim veri kümesinin büyümesi doğruluk oranlarında bir artışa neden olmuştur.

Çizelge 4.3.'te verilen sonuçlara göre bizim modelimiz, doğruluk oranı olarak daha yüksek bir başarı göstermiştir. Çizelge 4.3.'te yer alan BabelGum basit model, eklerin toplu olarak

ele alındığı modeldir. Bigram model ise eklerin ayrı ayrı ele alındığı modeldir. BabelGum bu deneyde de daha fazla sözcük türetmiştir. Ama doğruluk oranı %10 ve altında kalmıştır. Daha önce anlatılan nedenlerden ötürü geliştirdiğimiz model daha yüksek bir başarı sağlamıştır

Sonuç olarak karşılaştırdığımız tek sistem olan BabelGum bizim sistemimize göre oldukça düşük doğruluk oranı elde etmiştir. Kök ve eklerin kategoriler altında toplanması bu iki sistemin başarısı arasındaki en büyük farklılığı oluşturmaktadır.

5. SONUÇLAR VE TARTIŞMA

Bu tez çalışmasında, Türkçe sözcükleri denetimsiz olarak türetebilmeyi amaçlayan bir model geliştirilmiştir. Bu model sözcük kök ve eklerini ayrı ayrı kategorilere ayırmaya çalışmaktadır. Bu kategoriler sonlu durum özdevinirlerinde (FSA) kullanılarak sözcük türetme işlemi gerçekleştirilmiştir. Çalışmada, sözcükler, açık kaynak kodlu Türkçe doğal dil işleme kütüphanesi olan Zemberek [21] kullanılarak morfemlerine ayrıştırılmıştır. Sözcükler, bunun dışında hiçbir işaretlenmiş veri kullanılmadan türetilmiştir.

Benzer türdeki (isim, fiil sıfat vb.) sözcüklerin benzer ekleri alabileceği öngörüsünden yola çıkılarak, öncelikle, sözcük kökleri türlerine ayrıştırılmaya çalışılmıştır. Bu aşamada birden fazla metod denenmiştir. Son olarak, sözcük köklerinin benzerliklerini hesaplamak için iki metrik sırayla kullanılmıştır. Jensen-Shannon ıraksama metriği ile Baek [68] tarafından geliştirilmiş, morfemlerin benzerliğini hesaplayan bir metrik sırayla kullanılmıştır. Her iki metrik de kökleri arası benzerliği hesaplarken, köklerin aldığı eklere bakılmıştır. Benzer ekleri alan sözcükler aynı kategoride toplanmıştır. Ekler de sözcük kökleri gibi kümelendirilmiştir. Eklerin alomorfik özellikleri dikkate alınarak kümeleme işlemi gerçekleştirilmiştir.

Sözcük türetmek için sonlu durum özdevinirleri kullanılmıştır. Her sözcük kökü kategorisi için bir FSA oluşturulmuş ve başlangıç durumu dışındaki durumlar ek kategorileriyle doldurulmuştur. Her bir durumdan geçilirken bir sözcük türetilmiştir. Sözcük türetme aşamasında Türkçedeki harf uyumlarını ele alabilmek için harf ikilileri (bigram) istatistikleri kullanılmıştır. Bu harf ikilileri veri kümesinde yer alan sözcüklerin morfem sınırlarından elde edilmiştir. Türkçedeki yazımsal kurallar da benzer şekilde veri kümesinden çıkarılmaya çalışılmış ve sözcük türetme aşamasında ses olaylarını (yumuşama, ses düşmesi vb.) ele alabilmek için kullanılmıştır.

Sözcük türetme çalışmasında yaklaşık 85 bin sözcükten 3049 biricik sözcük kökü tespit edilmiş ve bu kökler kategorilerine ayrıştırılmıştır. Model, 3049 sözcük kökünden toplamda 744.061 sözcük türetilebilmiştir. Türetilen bu sözcüklerin $\%82.36$'sı doğru olarak belirlenmiştir.

Bu tezde, Türkçe için morfolojik türetmenin denetimsiz olarak gerçekleştirilebileceği gösterilmiştir. Sadece 3049 sözcük kökünden yaklaşık 750 bin sözcük, $\%82.36$ doğruluk oranı ile türetilmiştir. Bu çalışma ile morfemlerine ayrılmış veri dışında hiçbir işaretlenmiş, büyük boyutlu, veri kullanılmadan Türkçenin morfolojik yapısının öğrenilebileceği gösterilmiştir.

Sözcük kökleri ve ekler tamamen denetimsiz olarak kümelenmişlerdir. Benzer şekilde Türkçedeki harf uyumları ve yazımsal kurallar denetimsiz olarak öğrenilmeye çalışılmıştır.

Çalışma sırasında kök ve ek kümelenmesi için hiçbir sözdizimsel bilgi kullanılmamıştır. Sözdimsel bilginin kulanılması gelecekte bu çalışmaya eklenebilcek bir özelliktir. Örneğin *taş+ın* sözcüğü birden fazla anlama gelebilir. Sadece sözcüğün bağlam bilgisi (cümle içerisinde kullanıldığı yer, konu) bilindiği zaman aradaki ayrım yapılabilir:

- *taş* + *ın* : *taş* (isim) + *ın* (iyelik eki)
- *taş* + *ın* : *taş* (fiil) + *ın* (2. şahıs emir kipi)

Çalışmada kök ve ek kategorilerinin başarısı, sözcük türetme çalışmasını doğrudan etkilemektedir. Köklerin daha iyi kümelenebilmesi için ek kategorileri kullanılabilir. Ekler kendi içlerinde, alomorfik özelliklerine göre kümelendikten sonra elde edilen kategoriler sözcüklerin kümelenmesi aşamasında kullanılabilir. Sözcük kökleri aldığı eklere göre kümelenmeleri yerine ek kategorilerine göre kümelenebilirler. Bu da gelecekte yapılabilecek çalışmalar arasında yer almaktadır.

Hatalı üretilen sözcükler gözlemlendiğinde ek kategorilerindeki yanlışların önemli bir yer tuttuğu belirlenmiştir. Örneğin erek ve ecek eklerin hatalı bir şekilde aynı kategori içerisinde yer alması *giyemeyereksiniz* ve *aylıklaşılabilerek* gibi Türkçede yer almayan sözcüklerin türetilmesine neden olmuştur. Gelecekte bu bölümde de iyileştirmelerin yapılması söz konusudur.

KAYNAKLAR

[1] Ramchandra P Bhavsar ve BV Pawar. Rule based word morphology generation framework. *International Journal of Computer Science Issues*, **2011**.

[2] Mohammad Sadegh Rasooli, Thomas Lippincott, Nizar Habash, ve Owen Rambow. Unsupervised morphology-based vocabulary expansion. In *ACL (1)*, pages 1349–1359. **2014**.

[3] Lauri Karttunen ve Kenneth R Beesley. A short history of two-level morphology. *ESSLLI-2001 Special Event titled" Twenty Years of Finite-State Morphology*, **2001**.

[4] T. Bovermann, J. Rohrhuber, ve H. Ritter. Durcheinander: Understanding clustering via interactive sonification. In *Proceedings of the 14th International Conference on Auditory Display*. Paris, France, **2008**.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, ve Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, **2013**.

[6] Vishal Goyal ve Gurpreet Singh Lehal. Hindi morphological analyzer and generator. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 1156–1159. IEEE, **2008**.

[7] E Ross Stuckless. Real-time transliteration of speech into print for hearing-impaired students in regular classes. *American Annals of the Deaf*, 128(5):619–624, **1983**.

[8] Michele Banko ve Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting on association for computational linguistics*, pages 26–33. Association for Computational Linguistics, **2001**.

[9] John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198, **2001**.

[10] Hoifung Poon, Colin Cherry, ve Kristina Toutanova. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of*

the Association for Computational Linguistics, pages 209–217. Association for Computational Linguistics, **2009**.

[11] Benjamin Snyder ve Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *ACL*, pages 737–745. **2008**.

[12] Mathias Creutz ve Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 21–30. Association for Computational Linguistics, **2002**.

[13] Mathias Creutz ve Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):3, **2007**.

[14] August Schleicher. *Zur Morphologie der Sprache*, volume 1. K. Akademie der wissenschaften, **1859**.

[15] Brian Roark ve Richard William Sproat. *Computational approaches to morphology and syntax*. Oxford University Press Oxford, **2007**.

[16] Burcu Can ve Suresh Manandhar. Probabilistic hierarchical clustering of morphological paradigms. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 654–663. Association for Computational Linguistics, Stroudsburg, PA, USA, **2012**. ISBN 978-1-937284-19-0.

[17] Peter H. Matthews. *Morphology*. Cambridge University Press, **1991**.

[18] Mark Aronoff. *Morphology by itself: Stems and inflectional classes*. 22. MIT press, **1994**.

[19] Mark Aronoff. Word formation in generative grammar. *Linguistic Inquiry Monographs Cambridge, Mass.*, (1):1–134, **1976**.

[20] Celia Kerslake Aslı Göksel. *Turkish: A Comprehensive Grammar*. Rutledge, London, **2005**.

[21] Ahmet Afsin Akın ve Mehmet Dündar Akın. Zemberek, an open source nlp framework for turkic languages. structure. *Structure*, 10:1–5, **2007**.

[22] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, **1957**.

[23] John Lyons. *Natural Language and Universal Grammar: Volume 1: Essays in Linguistic Theory*, volume 1. Cambridge University Press, **1991**.

[24] James F Allen. Natural language processing. In *Encyclopedia of Computer Science*, pages 1218–1222. John Wiley and Sons Ltd., **2003**. ISBN 0-470-86412-5.

[25] Karen Sparck Jones. Natural language processing: a historical review. In *Current issues in computational linguistics: in honour of Don Walker*, pages 3–16. Springer, **1994**.

[26] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, **1950**.

[27] Christopher D Manning ve Hinrich Schütze. *Foundations of statistical natural language processing*, volume 999. MIT Press, **1999**.

[28] Andrew Carstairs-McCarthy. *An introduction to English morphology: words and their structure*. Edinburgh University Press, **2002**.

[29] Michael Fleming, Frank Hardman, David Stevens, ve John Williamson. *Meeting the Standards in secondary English: A guide to the ITT NC*. Routledge, **2003**.

[30] Muayad Abdul-Halim Ahmad Shamsan ve Abdul-majeed Attayib. Inflectional morphology in arabic and english: A contrastive study. *International Journal of English Linguistics*, 5(2):139, **2015**.

[31] Hatice ŞAHİN. Türkçe'de ön ek. *Uludağ Üniversitesi Fen-Edebiyat Fakültesi Sosyal Bilimler Dergisi*, pages 65–77, **2006**.

[32] Nizar Habash, Owen Rambow, ve George Kiraz. Morphological analysis and generation for arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 17–24. Association for Computational Linguistics, **2005**.

[33] MA Ford, MH Davis, ve WD Marslen-Wilson. Derivational morphology and base morpheme frequency. *Journal of Memory and Language*, 63(1):117–130, **2010**.

[34] Kemal Oflazer, Elvan Göçmen, Elvan Gocmen, ve Cem Bozsahin. An outline of turkish morphology. **1994**.

[35] Roman Jakobson. *Structure of language and its mathematical aspects*, volume 12. American Mathematical Soc., **1961**.

[36] Kemal Oflazer. Two-level description of turkish morphology. *Literary and linguistic computing*, 9(2):137–148, **1994**.

[37] Kimmo Koskenniemi. *Two-level morphology: a general computational model for word-form recognition and production*. Department of General linguistics, University of Helsinki, **1983**.

[38] Ronald M Kaplan ve Martin Kay. Phonological rules and finite-state transducers. In *Linguistic Society of America Meeting Handbook, Fifty-Sixth Annual Meeting*, pages 27–30. **1981**.

[39] Kimmo Koskenniemi. An application of the two-level model to finnish. *Computational morphosyntax: Report on research*, 1984:19–41, **1981**.

[40] Lauri Karttunen ve Kent Wittenburg. A two-level morphological analysis of english. In *Texas Linguistic Forum*, volume 22, pages 217–228. **1983**.

[41] Yukiko Sasaki Alam. A two-level morphological analysis of japanese. In *Texas Linguistic Forum*, volume 22, page 14Although. **1983**.

[42] Robert Khan. A two-level morphological analysis of rumanian. In *Texas Linguistic Forum Austin, Tex.*, 22, pages 253–270. **1983**.

[43] S Lun. A two-level morphological analysis of french. In *Texas Linguistic Forum*, volume 22, pages 271–278. **1983**.

[44] Jorge Hankamer. Finite state morphology and left to right phonology. In *Proceedings of the West Coast Conference on Formal Linguistics*, volume 5, pages 41–52. **1986**.

[45] Linfeng Song, Yue Zhang, Kai Song, ve Qun Liu. Joint morphological generation and syntactic linearization. In *AAAI*, pages 1522–1528. **2014**.

[46] Damir Boras, Davor Lauc, ve Nives Mikelić. Automatic morphological generation and analysis for the croatian language: Lexical inflectional database as personal name recognition module.

[47] Selçuk Köprü ve Jude Miller. A unification based approach to the morphological analysis and generation of arabic. In *3rd Workshop on Computational Approaches to Arabic Script-based Languages at MT Summit XII, A. Farghaly, K. Megerdoomian, and H. Sawaf, Eds. Ottowa, Canada: IAMT*. Citeseer, **2009**.

[48] Evan L Antworth. Pc-kimmo: a two-level processor for morphological analysis. *Occasional Publications in Academic Computing*, **1990**.

[49] Haşim Sak, Tunga Güngör, ve Murat Saraçlar. Resources for turkish morphological processing. *Language resources and evaluation*, 45(2):249–261, **2011**.

[50] Violetta Cavalli-Sforza, Abdelhadi Soudi, ve Teruko Mitamura. Arabic morphology generation using a concatenative strategy. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 86–93. Association for Computational Linguistics, **2000**.

[51] Kenneth R Beesley. Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 89–94. Association for Computational Linguistics, **1996**.

[52] George Anton Kiraz. Multi-tape two-level morphology: a case study in semitic non-linear morphology. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 180–186. Association for Computational Linguistics, **1994**.

[53] Abdelhadi Soudi, Violetta Cavalli-Sforza, ve Abderrahim Jamari. A computational lexeme-based treatment of arabic morphology. In *Proceedings of the Arabic Natural Language Processing Workshop, Conference of the Association for Computational Linguistics (ACL 2001)*, pages 50–57. **2001**.

[54] Nizar Habash. Large scale lexeme based arabic morphological generation. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*. **2004**.

[55] Tim Backwalter. Arabic morphological analyzer version 1.0. In *Linguistic Data Consortium*. University of Pennsylvania, **2002**.

[56] Khaled Shaalan, Azza Abdel Monem, ve Ahmed Rafea. Arabic morphological generation from interlingua. In *Intelligent Information Processing III*, pages 441–451. Springer, **2006**.

[57] Alexander Gode ve Hugh Edward Blair. *Interlingua: A Grammar of the International Language*. New York: Strom Publishers, **1951**.

[58] Alon Lavie, Florian Metze, ve Fabio Pianesi. Enhancing the usability and performance of nespole!: a real-world speech-to-speech translation system. In *Proceedings of the second international conference on Human Language Technology Research*, pages 269–274. Morgan Kaufmann Publishers Inc., **2002**.

[59] Einat Minkov, Kristina Toutanova, ve Hisami Suzuki. Generating complex morphology for machine translation. In *ACL*, volume 7, pages 128–135. **2007**.

[60] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. **2005**.

[61] Andrew McCallum, Dayne Freitag, ve Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, volume 17, pages 591–598. **2000**.

[62] Bernard Comrie. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press, **1989**.

[63] George W Stocking. *The ethnographer's magic and other essays in the history of anthropology*. Univ of Wisconsin Press, **1992**.

[64] Geoffrey Lewis. *Turkish Grammar*. Oxford University Press, Oxford, **2001**.

[65] Solomon Kullback ve Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, **1951**.

[66] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, ve Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. **2013**.

[67] word2vec - tool for computing continuous distributed representations of words. `https://code.google.com/archive/p/word2vec/`. Oluşturulma: Temmuz 30, 2013.

[68] Dae-Ho Baek, Ho Lee, ve Hae-Chang Rim. Conceptual clustering of korean concordances using similarity between morphemes. **1997**.

[69] Kemal Ofazer, Sergei Nirenburg, ve Marjorie McShane. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computational linguistics*, 27(1):59–85, **2001**.

[70] Mustafa Burak Öztürk ve Burcu Can. Clustering word roots syntactically. In *2016 24th Signal Processing and Communication Application Conference (SIU)*, pages 1461–1464. IEEE, **2016**.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : MUSTAFA BURAK ÖZTÜRK
Doğum Yeri : ESKİŞEHİR
Medeni Hali : Bekar
E-posta : burakozturk.cs@gmail.com
Adresi : Mutlukent Mah. Eserköy Sitesi 3-A/2 Blok Daire:2
Ümitköy,Çankaya,ANKARA

Eğitin

Lise : Karşıyaka Atakent Anadolu Lisesi, İzmir, TÜRKİYE, 2005
Lisans : Bilgisayar Mühendisliği, Hacettepe Üniversitesi, Ankara, TÜRKİYE, 2012

Yabancı Dil

İngilizce

İş Deneyimi

Bilgisayar Mühendisi, Bor Yazılım (Haziran,2012 - Halen)

Deneyim Alanları

Doğal Dil İşleme
Makine Öğrenmesi
Android Uygulama Geliştirme

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

Sözdizimsel Olarak Sözcük Köklerinin Kümelenmesi [70]. SIU 2016