T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİÇİMSEL ONTOLOJİ TABANLI BİLGİSAYARLI FRAMENET ÇALIŞMASI

Yelda FIRAT

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Yrd. Doç. Dr. Özlem UÇAR

EDİRNE-2015

T.Ü. Fen Bilimleri Enstitüsü onayı

Prof. Dr. Mustafa ÖZCAN
Fen Bilimleri Enstitüsü Müdürü

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

Prof. Dr. Yılmaz KILIÇASLAN
Anabilim Dalı Başkanı

Bu tez tarafımca okunmuş, kapsamı ve niteliği açısından bir Doktora tezi olarak kabul edilmiştir.

Yrd. Doç. Dr. Özlem UÇAR
Tez Danışmanı

Bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir Doktora tezi olarak oy birliği ile kabul edilmiştir.

Jüri Üyeleri	İmza
Prof. Dr. Ayşe Zeynep Hiçşaşmaz KATNAŞ	
Prof. Dr. Yılmaz KILIÇASLAN	
Prof. Dr. Nevcihan DURU	
Yrd. Doç. Dr. Özlem UÇAR	
Yrd. Doç. Dr. Erdinç UZUN	

Tarih: 13/03/2015

T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ DOKTORA PROGRAMI
DOĞRULUK BEYANI

İlgili tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin kaynak gösterilerek ilgili tezde yer aldığını beyan ederim.

13/03/2015

Yelda FIRAT

Doktora Tezi
Biçimsel Ontoloji Tabanlı Bilgisayarlı FrameNet Çalışması
T.Ü. Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Son yıllarda yapılan gerek doğal dil işleme gerekse hesaplamalı dilbilmi çalışmaları insan zihninin biçimsel yapısını ve içeriğini anlamaya yöneliktir. İnsan zihni tarafından yaratılan en genel model ontolojilerdir. Böylece ontolojilerle dil arasında bir ilişkinin kurulması gerekir. Ontolojiler doğaları gereği büyük olmaları nedeniyle, insan emeği ile oluşturulmalarında hata ve eksiklikler barındırma potansiyeli taşırlar. Bu da dilin insan zihnindeki anlam üretme süreçlerini matematiksel olarak gerçekleştirmeyi gerektirir. Bu anlamda dilin sözdizimsel ve anlamsal analizlerini sunan bir kaynağa ihtiyaç duyulur. Bunu en iyi sunan sözlüksel kaynak FrameNet'tir. Bu anlamda, bilgisayar ortamında FrameNet üzerinden tematik rol latisleriyle gerçeklik modelleri yaratma yoluna gidilir. Yaratılan bu gerçeklik modelleri ile dil ile FrameNet arasında bir ilişki sağlanır. Bu ilişki dil ile ontoloji arasındaki bağı da kurmuş olur. Ayrıca, bu gerçeklik modellerinin bilgisayarlı gerçekleştirimi, Türkçe'nin anlamsal çözümlenmesine de katkı sağlar.

Yıl : 2015
Sayfa Sayısı : 131
Anahtar Kelimeler : Biçimsel Ontoloji, Anlamsal Yapılar, FrameNet, Tematik Roller, Kavram Latisleri, Sözcük Anlam Belirginleştirme

Doctoral Thesis
Computerized Constructed Study of FrameNet Through Formal Ontology
Trakya University Institute of Natural Sciences
Department of Computer Engineering

ABSTRACT

Recent studies on both natural language processing and computational linguistics are directed towards understanding the formal structure of the human mind and its contents. The most general models created by the human mind are ontologies. However, a relationship between the language and ontology needs to be established. Ontologies large in nature, carry the risk of hosting potential for errors and lacks as they are being developed through human labor. Obtaining a relationship between the language and ontology requires processing of the meaning by generating language in the human mind mathematically. In this sense, a resource that provides the syntactic and semantic analyses of the language is needed. The lexical resource which offers this best is the FrameNet. For this purpose, reality models with thematic role lattices were created on the computer using the FrameNet. A relationship between language and FrameNet is provided by these reality models. This relationship establishes the link between ontology and language as well. Computerized implementation of this reality model contributes to semantic analysis of the Turkish language as.

Year : 2015
Number of Pages : 131
Keywords : Formal Ontology, Semantic Structures, FrameNet, Thematic Roles, Concept Lattices, Word Sense Disambiguation

ÖNSÖZ

Değerli danışmanım Yrd. Doç. Dr. Özlem UÇAR'a tez çalışmam boyunca gerek bilgisinden gerekse tecrübelerinden yararlanırken göstermiş olduğu hoşgörü ve sabrından dolayı teşekkürü bir borç bilir, saygı ve sevgilerimi sunarım.

Doktora çalışmalarım boyunca ilminden faydalandığım, yanında çalışmaktan onur duyduğum değerli hocam Prof. Dr. Yılmaz KILIÇASLAN'a müteşekkir olduğumu belirtir, çalışmalar boyunca sunduğu yardımları, göstermiş olduğu inceliği ve hoşgörüsü için gönülden teşekkür eder, saygılarımı sunarım.

Ayrıca çalışmalarım boyunca yardımlarını ve desteklerini esirgemeyen annem Özden FIRAT ve babam Emin FIRAT'a, yeri geldikçe beni motive etmeyi başaran oğlum Berkay'a tüm içtenliğimle teşekkür ederim.

Yelda FIRAT

İÇİNDEKİLER

ÖZET…….. i

ABSTRACT……. ii

ÖNSÖZ…… iii

SİMGELER DİZİNİ…… vi

TABLOLAR LİSTESİ……. viii

ŞEKİLLER LİSTESİ…….. ix

1. GİRİŞ……... 1

2. ALANYAZIN BİLGİLERİ……..……..... 3

2.1. Bir Anlam İlişkisi Olan Dil……. 3

2.2. Tematik Roller……..…….. 4

2.3. FrameNet……... 14

2.4. Biçimsel Ontoloji……... 18

3. DİL İLE FRAMENET ARASINDA BİR KÖPRÜ OLARAK TEMATİK ROL LATİSLERİ……...……..... 24

3.1. FrameNet'in Amorf Yapısı ….....……..... 24

3.1.1. Çerçeve Yapıları ……. 27

3.1.2. FrameNet Çerçeveleri Arasındaki İlişkiler……..…… 30

3.2. Tematik Rol Latisleri…… 37

3.2.1. Latislerde Tematik Alanlar…….. 43

3.2.2. Referans Parametreleriyle Zenginleştirilmiş Tematik Rol Yapıları…… 47

3.2.2.1. Konumsal Referanslar……... 47

3.2.2.2. Figüral Referanslar ……..... 49

3.2.2.3. Psikolojik Referanslar…… 50

3.2.3. Kavram Latislerinde Tematik Rol Yapıları……. 51

3.3. FrameNet İle Tematik Rol Yapıları Arasındaki İlişki……... 57

3.4. FrameNet - Tematik Rol Yapıları İlişkisine Yönelik Yapılan Çalışmalar…… 65

4. TÜRKÇE'NİN ANLAMSAL ÇÖZÜMLENMESİNİN BİLGİSAYARLI GERÇEKLEŞTİRİMİ……..... 72

5. SONUÇLAR VE TARTIŞMA…… 130

KAYNAKLAR……

ÖZGEÇMİŞ……
TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER…….....

SİMGELER DİZİNİ

ASP	Aspect
bry_referans	bireysel referans
CaseP	Case Phrase
CNI	Constructional Null Instantionation
Comp	Complement
DNI	Definite Null Instantionation
dnym	deneyimleyen
DP	Determiner Phrase
Ext	External
fig	figüral
hdf	hedef
INI	Indefinite Null Instantionation
knm	konum
kynk	kaynak
NP	Noun Phrase
P	Preposition
PP	Preposition Phrase
psk	psikolojik
S	Sentence
sit_referans	situasyonel referans
T	Tense
TP	Tense Phrase
uzm	uzamsal
V	Verb
VP	Verb Phrase

Kısaltmalar

AspEM	Aspect Event Measurer
AspOR	Aspect Originator
BNC	British National Corpus
CCG	Combinatorial Categorial Grammar

DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
DRS	Discourse Representation Structure
DRT	Discourse Representation Theory
FE	Frame Elements
LU	Lexical Units
MYSQL	My Sequential Query Language
PHP	Hypertext Preprocessor
SUMO	Suggested Upper Merged Ontology
UTAH	Uniformity of Theta Assignment Hypothesis

TABLOLAR LİSTESİ

Tablo 2.1 Panini'nin Sanskrit için Karaka rolleri
Tablo 2.2 *'give'* eyleminin sözcüksel girişi
Tablo 3.1 REVENGE çerçevesinin anlamsal rolleri
Tablo 3.2 CUTTING çerçevesinin anlamsal rolleri
Tablo 3.3 COMMERCE_MONEY_TRANSFER çerçevesinin çerçeve elemanları
Tablo 3.4 COMMERCE_GOODS_TRANSFER çerçevesinin çerçeve elemanları
Tablo 3.5 Transfer çerçevesinin çerçeve elemanları
Tablo 3.6 QUARRELLING çerçevesinin örnek tümcelerle tematik rol analizi
Tablo 3.7 REVENGE çerçevesinin örnek tümcelerle tematik rol analizi
Tablo 3.8 Örnek Türkçe tümcenin tematik rol analizi
Tablo 3.9 *'argue'* eyleminin sözcüksel girişi
Tablo 3.10 Anlamsal olarak işaretlenmiş derlem tümceleri
Tablo 3.11 Almanca *'streiten (tartışmak)'* eyleminin sözcüksel girişi
Tablo 3.12 Anlamsal olarak işaretlenmiş Almanca'da karşılığı olan tümceler
Tablo 4.1 Örnek işaretleme

ŞEKİLLER LİSTESİ

Şekil 2.1 (11) nolu tümcenin kural ağacı
Şekil 2.2 Dilbilgisi teorilerinde öğe yapısının rolü
Şekil 2.3 FrameNet anlamsal tiplerin bir parçasının SUMO'ya bağlantıları
Şekil 2.4 LF :: Motion ve LF :: Filling çerçeveleri için LF tip tanımlamaları
Şekil 2.5 Nesne ve özelliklerden oluşan kavram latisi
Şekil 2.6 Eklenen *'havuz'* nesnesiyle oluşan kavram latisi
Şekil 2.7 Yeni kavramların önerilebileceği situasyon
Şekil 2.8 İki kavramı birbirinden ayırmaya ihtiyaç duyulan durum
Şekil 2.9 Kavram latisinin bir parçası
Şekil 2.10 Tüm özellik ve nesnelerden sonra oluşan final ontolojisi
Şekil 3.1 (18) nolu tümceye ait kural ağacı
Şekil 3.2 (19) nolu tümceye ait kural ağacı
Şekil 3.3 CUTTING çerçevesine ait kalıtım ilişkilerini gösteren kısmın bir parçası
Şekil 3.4 Criminal - Process çerçevesi ve onun alt-çerçeveleri
Şekil 3.5 Çerçevelerin ilişki şekilleri
Şekil 3.6 TRANSFER çerçevesine ait ilişkiler şeması
Şekil 3.7 FrameNet'in amorf yapısı
Şekil 3.8 Üç boyutlu örnek kavram latisi
Şekil 3.9 Boxer tarafından DRS çıkışı
Şekil 3.10 Uzamsal konum latisi
Şekil 3.11 Figüral konum latisi
Şekil 3.12 Psikolojik alanın hiyerarşik organizasyon latisi
Şekil 3.13 Konumsal referanslar
Şekil 3.14 Figüral referanslar
Şekil 3.15 Psikolojik referanslar
Şekil 3.16 Tematik katmana ait örnek kavram latisi
Şekil 3.17 İki katman arasındaki bağlantı noktası
Şekil 3.18 Kavram latislerinde tematik ve eylem katmanlarına ait tematik rollerin birlikte gösterimi
Şekil 3.19 Quarreling çerçevesine ait ilişkiler

Şekil 3.20 Revenge çerçevesine ait ilişkiler
Şekil 3.21 Figüral referanslar ile Türkçe tümcenin anlamsal çözümlenmesi
Şekil 3.22 Placing çerçevesine ait ilişkiler
Şekil 3.23 FrameNet'in genelleştirilmiş hali
Şekil 3.24 Dizimsel özellikleri ve tematik rol bilgileri üzerinden anlamsal rollerin belirlendiği örnek tümce
Şekil 3.25 VerbNet sözlüğünden alınan örnek eylem sınıfı
Şekil 3.26 İngilizce ve Almanca sözcük girişlerin eşleştirilmesi
Şekil 4.1 yelda_tool.sql veritabanına ait tablolar
Şekil 4.2 Biçimbilimsel / dizimsel - tematik rol işaretleme ara yüzü
Şekil 4.3 İşaretlenmiş örnek tümcenin otomatik olarak çıkarılmış kural ağacı
Şekil 4.4 İşaretlenmiş örnek tümcenin otomatik olarak çıkarılmış latis diyagramı
Şekil 4.5 Tümce güncelleme ve silme arayüzü
Şekil 4.6 Tümce ekleme arayüzü
Şekil 4.7 Örnek bir tümcenin işaretleme arayüzü

BÖLÜM 1

GİRİŞ

Bu çalışmanın amacı, dil ile sözlüksel bir veri kaynağı olan FrameNet arasında tematik rol latisleriyle matematiksel bir köprü oluşturmaktır.

Bu matematiksel köprünün oluşturulmasında ontoloji kullanımı bir gerekliliktir. Ontolojiler doğaları gereği büyük olmaları nedeniyle, insan emeği ile oluşturulmalarında hata ve eksiklikler barındırma potansiyeli taşırlar. Ayrıca, bu şekilde oluşturulmuş olan bir ontolojinin uygulama gereksinimlerinden doğabilecek uyarlamaları da güç olacaktır. Dolayısıyla ontolojiler bilgisayar ortamında hazırlanmalıdır. Bu da ontolojilerin biçimsel olmasını gerektirir. Hazırlanan bu modeller, biçimsel bir dil kullanarak tanımlanmış gerçeklik modelleridir. Özellikle bilgisayarlı dilbilim uygulamalarında makine çevirisi sistemleri için bir model görevi görerek oluşabilecek zorlukları da ortadan kaldırırlar. Bu anlamda dilin sözdizimsel ve anlamsal analizlerini sunan bir kaynağa ihtiyaç duyulur. Bunu en iyi şekilde sunan sözlüksel kaynak FrameNet'tir. Leksikografik bir sözlük olan FrameNet, daha çok anlamsal roller üzerine kurulu bir model olmasına rağmen kavramların gerçek çıkarımlarını vermez. Dolayısıyla, bu modelin bazı biçimsel ontolojilerle ilişkilendirilmesi yapılarak dil ile bağlantısı sağlanır. SUMO (Suggested Upper Merged Ontology), LF, OMEGA, CYC, DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) WordNet, FrameNet gibi sözlüksel kaynakları kullanan belli başlı ontolojilerdendir. OMEGA'nın biçimsel olmayan kavram tanımları içermesi, CYC ve DOLCE ontolojilerinin de WordNet sözlüğünün küçük bir bölümünü eşleştirmesinden dolayı, doğal dil işleme uygulamalarını en iyi şekilde uygulayan SUMO ve LF ontolojileri FrameNet ile ilişkilendirilir. Daha önce yapılan çalışmalarda SUMO ile WordNet sözlüğünün tamamının eşleştirilmesi mevcuttur. Fakat WordNet, FrameNet gibi kavramların zengin anlamsal detaylarını ve biçimsel tanımlarını içermediğinden ve daha çok sözcüksel bilgiye sahip olmasından gereken çözümü

vermez. Dolayısıyla FrameNet sözlüksel kaynağındaki dilsel açıdan eksik olan anlamsal roller, biçimsel olarak tanımlanan SUMO sınıfları ile anlamlı hale gelir. Fakat SUMO'da da kavramlar, sözcüksel bilgi anlamında eksik kaldığından doğru eşleştirmelerin yapılmadığı görülür. LF ontolojisinde ise sözdizim ve anlam eşleştirmesi seçimsel sınırlılıklar (selectional restrictions) üzerinden yapılır. Sözdizimsel genellemelerin eksikliğinden ve anlam ile sözdizim arasındaki uyumu sağlamada kullanılan ETKİLEYEN, HEDEF, KAYNAK, TEMA, SEBEP gibi anlamsal rollerin FrameNet sözlüğündeki çerçeve elemanlarını yeterli ölçüde karşılamamasından bu ontoloji tipi de istenileni vermez. Bu nedenle kavramların gerçek çıkarımlarını sağlayan tematik rollerin kullanılması bir zorunluluk olur. Bu rollerin fraktal latislerde kullanılmasıyla da dil ile FrameNet arasındaki etkileşim sağlanır ve FrameNet sözlük kaynağı biçimlendirilir. Böylece mevcut FrameNet sözlüksel kaynağının tematik rol latisleriyle matematiksel bir yapı oluşturulur. Dolayısıyla, insana ait zihin yapısını ifade eden ontolojiler ile dil arasındaki ilişki de sağlanmış olur. Ayrıca bilgisayar ortamında oluşturulan bu yapı ile bir eyleme ait olan sözcüklerin ya da kavramların bilgi tabanı da yaratılır. Farklı diller için bir sözlük kaynağı gibi kullanılan bu bilgi tabanı, özellikle Türkçe'nin anlamsal çözümlenmesine katkı sağlar.

Bu bağlamda, çalışmanın ikinci bölümünde ontoloji ile kural arasındaki anlamsal ilişkiyi gösteren tematik rollerle ilgili arka plan bilgileriyle, FrameNet sözlüksel kaynağı ve bu kaynağın çeşitli tip ontolojilerle ilişkisi anlatılarak biçimsel ontolojiler hakkında bilgiler verilmiştir. Üçüncü bölümde mevcut FrameNet sözlüksel kaynağının eleştirisi yapılarak, biçimsel ontoloji tabanlı tematik rol latisleriyle oluşturulmuş gerçeklik modelleriyle ilişkisi anlatılmıştır. Dördüncü bölümde ise, derlem tabanlı işaretleme yöntemi ile tematik rol latisleriyle oluşturulmuş bir gerçeklik modelinin bilgisayarlı gerçekleştirimi gösterilmiştir. Bu bilgisayarlı gerçekleştirimde Türkçe'nin anlamsal çözümlenmesi hedeflenmiştir.

BÖLÜM 2

ALANYAZIN BİLGİLERİ

2.1. Bir Anlam İlişkisi Olan Dil

İnsan zihni, beynin üst düzey süreçlerini (düşünme, algılama, vs.) kapsayan bir yapıya sahiptir. Dil ise, insan zihninin bu karmaşık yapısı içinde birincil öneme sahip bir olgudur. Zihinsel işlevlerin şekillenmesinde ve yorumlanmasında temel bir rol oynayan dil, anlam ile biçim arasında bir köprüdür [70, 71].

1950'li yıllardan itibaren dil edinimi ile ilgili yaptığı çalışmalarla birçok dilbilmi kuramına özellikle bilgisayarlı dilbilmi çalışmalarına öncülük etmiş olan Chomsky'ye göre dil, insan zihninin bir aynasıdır. Ona göre dil, doğuştan gelen biyolojik bir yetenektir. Chomsky, dillerin arasındaki farklılıkların dillere yüzeysel temelde bakmaktan kaynaklandığını, bu farklılıkları daha temel benzerliklerin kuşattığını söyler. Bu benzerliklerin kaynağında da dile dair zihinsel işlemleri yaparken kullanılan bir Evrensel Gramer'den (Universal Grammar) bahseder. Chomsky'a göre bir insan ana dilindeki ifadeleri oluşturmayı ve yorumlamayı bilinçaltında yapar. Bu anlamda Chomsky'ın amacı teker teker içselleştirilmiş dillerden yola çıkarak bütün dillerle ilgili genellemeler yapabileceği bir evrensel dilbilgisini ortaya çıkarmaktır. Ona göre zihin grameri denilen şey, bir tümcenin dilbilgisi kurallarına uygun olup olmadığına dair farkında olmadan bir yargıya varma işlemini içerir. Bu süreç, bir tümcenin anlamından bağımsızdır. Bu anlamda Chomsky dili eksiksiz olarak açıklamak için çok daha az zihinsel işlem yapmayı gerektiren bir program sunar. Ona göre bu program ne kadar basitse özellikle çocukların dili öğrenme hızlarını açıklamak o kadar kolay olur [67].

Böylece Chomsky doğal dilleri matematiksel olarak formüle ederek üretici dilbilgisi teorisi sayesinde kurallardan anlamlara giden yolu belirler. Böylelikle, dili daha net bir şekilde, her biri sonlu uzunlukta ve sonlu bir üyeler kümesinde oluşturulan (sonlu ya da sonsuz) tümceler kümesi olarak inceler [66].

Chomsky'nin çalışmaları ışığında, son yıllarda yapılan gerek doğal dil işleme gerekse hesaplamalı dilbilmi çalışmaları, insan zihninin biçimsel yapısını ve içeriğini anlamaya yöneliktir. İnsan zihni tarafından yaratılan en genel model ontolojilerdir. Böylece ontolojilerle dil arasında bir ilişkinin kurulması gerekir. Bu da dilin insan zihnindeki anlam üretme süreçlerini matematiksel olarak gerçekleştirmeyi gerektirir. Bilgisayar ortamında oluşturulan gerçeklik modelleri ile ifadelerin tam anlamıyla modellenmesi sağlanır. Bu modeller bilgisayar bilimlerinin yapay zeka (artificial intelligence), anlambilimsel ağ (the semantic web), bilgi mimarisi ve bilgi gösterimi (information architecture, knowledge representation) gibi alanlarında kullanılır.

2.2. Tematik Roller

Birçok dilbilim kuramında, tematik rol terimi bir ad öbeğinin tümcenin eylemi ile girdiği anlamsal ilişkiyi ifade eder. Bir başka ifadeyle dilin sözdizimi ile anlamı arasında bir arayüzdür.

Aşağıda bu çalışmada kullanılan belli başlı geleneksel tematik rollerin bir listesi yer almaktadır. Türkçe ve İngilizce tümcelerdeki altı çizili ad öbekleri tanımlanan tematik rolü taşımaktadır [1]:

TEMA (THEME) – Hareket halinde bulunan veya bir yerde olan

(1) a. Kedi odada.

b. The cat is in the room.

ETKİLEYEN (AGENT) – Bir eylemi başlatan, yapan

(2) a. Can camı kırdı.

b. John broke the glass.

ETKİLENEN (PATIENT) – Bir eylemden etkilenen

(3) a. Can camı kırdı.

b. John broke the glass.

KAYNAK (SOURCE) – Hareketin çıkış noktası

(4) a. Can İstanbul'dan Ankara'ya gitti.

b. John went to Ankara from İstanbul.

HEDEF (GOAL) – Hareketin yöneldiği şey

(5) a. Can İstanbul'dan Ankara'ya gitti.

b. John went to Ankara from İstanbul.

KONUM (LOCATION) – Bir şeyin bulunduğu yer

(6) a. Kedi odada.

b. The cat is in the room.

YOL (PATH) – Bir varlığın hareket ettiği yol

(7) a. Can evde koridordan geçerek oturma odasından mutfağa doğru yürüdü.

b. John walked from a living room to a kitchen through a hallway in a house.

DENEYİMLEYEN (EXPERIENCER) – Bir olayı hisseden, algılayan veya fark eden

(8) a. Kedi topu gördü.

b. The cat saw the ball.

ARAÇ (INSTRUMENT) – Bir eylemi gerçekleştirmek için kullanılan

(9) a. Can ekmeği bıçakla kesti.

b. John cut the bread with a knife.

Birçok dilbilimci, tematik rolleri ve bu rollerin biçimbilimsel / dizimsel özelliklerini gramer teorisinin önemli bir parçası olarak tanımlar [2, 3, 4, 5].[1]

Doğal dil işleme uygulamalarında, verilen bir tümcenin anlamını çözümlemede tümcenin öğelerinin tematik rollerini belirlemek oldukça önemlidir [6].

Bu anlamda tümce öğelerinin gerçek anlamlarına ulaşmada bir başka deyişle kavramlaştırmada tematik rollerin kullanımı bir zorunluluktur.

Özellikle öğelerin sınıflandırılmasında, tematik rollerin önemi çok eski yıllara, Panini'nin Karaka Teorisine kadar uzanmaktadır. Bu teori, Karaka roller olarak ifade edilen tematik roller ile tümce öğelerinin sınıflandırılmasında önemli bir yer tutar. Bu bağlamda Panini ana dilin (Sanskrit) biçimsel özellikleri ile düşüncenin anlamsal kategorileri arasındaki ilişkiyi kuran ilk kişidir. Dilbilgisinde yer alan kurallar, anlamsal ifadeler ile onların biçimbilimsel / dizimsel kodları arasındaki ilişkiyi açıklamaya çalışır [7].

Panini'ye göre her eylem bir eylem sınıfının üyesidir. Bu eylem sınıfları tümce öğelerinin Karaka rollerini belirlemek için ortak özelliklere sahiptir. Panini'nin Karaka rolleri Tablo 2.1'de gösterilmiştir [8].

[1] Okur, tematik rollere ilişkin literatürün kapsamlı bir tartışmasını Kasper (2008)'de bulabilir.

Tablo 2.1 Panini'nin Sanskrit için Karaka rolleri

Karaka rol	**Panini'nin tanımlaması**	**Tematik rol**
apadana	Sabit bir noktadan çıkma	Çıkma durumu
sampradana	Ulaşılmak istenen hedef	Hedef
karana	Bir eylemi gerçekleştirmek için gerekli en etkili araç	Araç
adhikarana	Yer, konum	Kalma durumu
karman	Bir eylemi başlatan tarafından arzu edilen şey	Etkilenen
kardr	Bir eylemi başlatan, bağımsız kişi	Etkileyen

Karaka teorisi çok detaylı bir şekilde tartışılmamasına rağmen tematik rollerin modern teorilerinde bir öncü teori olarak ele alınır.

Tematik roller konusunda önemli çalışmalardan biri olarak gösterilen ve 1930 yılında Frank R. Blake tümcenin eylemi ile öğeleri arasındaki biçimsel ilişkiden bahseder. Blake'e göre anlamsal ilişkiler için biçimsel özellikler durum formu (case form) olarak isimlendirilir. Blake'in dilbilgisinde biçimsel özellikler isimler, zamirler, sıfatlar, zarflar ve fiiller için belirtilerek, biçimsel özelliklere göre tematik roller ise ETKİLEYEN, ETKİLENEN, ARAÇ, TEMA, UYARI, HEDEF, ALICI, YARARLANICI olarak gösterilmiştir. Blake'in çalışmalarından elde edilen genellemeler öğelerin dizimsel özelliklerinden ziyade biçimsel özelliklerine göre yapılmıştır [9].

1950'li yıllarda Noam Chomsky dilbilimini bilişsel psikolojinin bir parçası olarak tanımıştır [10].

Bugüne kadar yapılan birçok çalışmanın amaçları ve hedefleri onun çalışmalarıyla yakından ilişkilidir. Chomsky Teorisinin en önemli özelliği dilin dizimsel bileşeninin anlamsal bileşeninden ayrı olarak ele alınmasıdır. Chomsky'nin örnek verdiği (10) nolu tümce bu teoride dizimin anlamdan bağımsız olarak ele alındığını gösteren en ünlü tümcesidir.

(10) a. Renksiz yeşil düşünceler kızgın şekilde uyurlar.

b. Colorless green ideas furiously sleep.

Tümcelerin anlamsal gösterimleri ile birlikte biçimsel kategorilerden veya sözcük öbek tiplerinden oluşan dizimsel yapısının ilişkilendirilmesi karşımıza bağlama

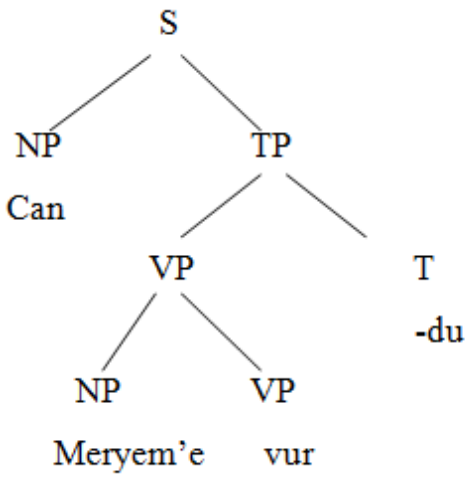
problemini (linking problem) ortaya koyar. Bir başka deyişle, şöyle bir soruyla karşılaşılır: Tümcelerin dilbilgisel yapıları anlamsal içeriklere veya yorumlara nasıl sahip olabilir? Bu sorunun cevabı (11) nolu örnek Türkçe tümce üzerinde şu şekilde gösterilir:

(11) a. Can Meryem'e vurdu.

b. *'Vuran' 'vurulan'* ı vurdu.

c. vur (x, y)

Dilin dizimsel yapısı Chomsky tarafından ağaçlarla gösterilmiştir. (11) nolu tümceye ait kural ağacı ise Şekil 2.1'de görülebilir.



Şekil 2.1 (11) nolu tümcenin kural ağacı

(11) nolu örnek tümcede de açıkça görüldüğü gibi vurmak eylemi anlamsal olarak her zaman iki öğeye sahiptir. Biri *'vuran'* diğeri ise *'vurulan'*dır. Dilbilgisel olarak ele alındığında ise '*vuran*' özne, '*vurulan*' ise nesne olarak ifade edilir. Böylece kural ile anlam gösterimleri arasında bazı genellemelerin yapılabileceği aşikardır.

Bir başka deyişle,'*vurmak*' eylemi ve onun öğelerinin sıralanışı *'öldürmek', 'okumak', 'görmek', 'yemek',* vs. gibi eylemlerin öğeleri ile aynıdır. Buradan da görüldüğü gibi *'öldüren', 'okuyan', 'gören', 'yiyen'* her zaman yapısal pozisyon içerisinde özne konumundadır. Bu anlamda, her dilbilimcinin bazı genellemeleri ortaya koyması için dizim ve anlam arasındaki ilişkiyi belirtmesi gerekir. Eğer bir tümcede eylemi gerçekleştiren var ise bu öğe mutlaka özne pozisyonu içinde yer almadır. Bunlar

bağlama teorisinde kendilerine bağımsız bir statü kazanarak meşrulaşırlar. ETKİLEYEN olarak isimlendirilerek eylemin tüm öğeleri arasında bir eylemi gerçekleştiren olurlar. Sonuç olarak, bağlama problemi ile birlikte tematik rollerin kullanılması da kaçınılmaz hale gelmektedir. Böylece bu roller eylem ve onun öğeleri arasındaki kural ve anlam arasındaki ilişkiyi göstererek gerçek anlamlara ulaşmada yol gösterici olurlar. 1980'li yıllarda Dowty bu rollerin gerekliliğini açık ifadelerle dile getirerek teorik konumlarını net bir şekilde ortaya koymuştur. Bu bağlamda karşımıza özel teorilerde ifade edilen tematik rollerin konumu için ne çeşit bir bilgiye ihtiyaç duyulur sorusu çıkmaktadır. Bir başka deyişle, ne çeşit bir bilgi anlamdan kurala giden yolu bulur? Bu soruların cevapları için birinci olasılık anlamın dünyanın evrensel bilgisi ile eşitlenebileceğidir. İkinci olasılık ise Chomsky'nın varsaydığı aynı yapıyı (aynı dilbilgisel kategorileri) paylaşan dizim ve (dilbilgisel olarak ilişkili) anlam ilişkisinden gelen bilgidir. Chomsky teorisine göre sadece dilbilgisel olarak ilişkili bu anlam parçaları, kural-anlam ara yüzünün bir kısmını oluşturur. Bir başka deyişle kurallardan öngörülen anlamsal gösterimlerdir [7].

Yukarıda bahsi geçen birinci olasılık 1983 yılından bu yana Jackendoff tarafından anlama yönelik olarak kavramsal yapı ismi üzerinden çalışılmaktadır. Ona göre dilbilimsel anlam algılanan dünya anlayışı ile koordine olur. Dolayısıyla insan algısı üzerinden ortaya çıkan tüm bilgiler dilsel olarak ilişkilidir [11].

Chomsky'ın söylediği gibi dil ile ilgili doğuştan gelen bilgilerin yanında duyu organları ile algılanan bilgiler de vardır.

Dikkate alınması gereken yaklaşımlarından biri Dowty'nin tematik rolleri belirlemede kullandığı sözcüksel ayrıştırma (lexical decomposition) yaklaşımıdır. Burada sözcüklerin gerçek anlamları bilişsel gerçeklik olarak tematik roller tarafından ele alınmalıdır. Ona göre sözcüklerin ya da öğelerin anlamsal belirleyicileri birer sözcüksel gereklilik gibi karakterize edilmelidir. Situasyonda yer alan eylem ona bağlı öğelerin anlamları üzerinden tanımlanır. Bu düşünceye bağlı olarak Dowty, kendi sözcüksel gereklilikleriyle birlikte ele alınan iki tane proto-rol önermiştir [65].

Bir başka deyişle; ona göre öğe seçimini tanımlamak için iki makro rol vardır. Bunlar: ETKİLEYEN ve ETKİLENEN dir (PROTO-AGENT, PROTO-PATIENT). İradeli, bilinçli olarak eylemi başlatan ve tümce içinde özne görevini gören öğe ETKİLEYEN tematik rolünü üstlenir. Üzerinde durum değişikliği meydana gelen, artan

TEMA (incremental THEME) özelliğine sahip olan ve bir başka öğe katılımcısının varlığından etkilenen öğe ise ETKİLENEN rolünü alır ve tümcenin nesnesi durumundadır [4].

Bu anlamda Croft'un rolleri ilk örnek olarak verilebilir. Bu roller bilişsel model üzerine oturtulmuş, olayların nedensel organizasyonundan türetilmiş rollerdir [12].

Croft ve 1972 yılında benzer rol tanımı yapan Jackendoff tematik rol kavramını anlamsal gösterimin bir parçası olarak ele almışlardır. Jackendoff'un daha sonraki yıllarda yaptığı çalışmalarda tematik roller kavramsal sistem olarak okunmuştur.

Wunderlich (2000) ve Kiparsky (1997)'nin yaptığı çalışmalarda ise Jackendoff'un çalışmalarına benzer bir dilbilgisel organizasyonun olduğu görülür. Bu çalışmalarda tematik rolleri içeren kavramsal sistem ve dilsel olarak ilişkili bilgiyi gösteren anlamsal form (semantic form) vardır [13].

Tematik rol durumları ile karşımıza çıkan bir başka soru ise bu rollerin kurallardan mı yoksa anlamdan mı kaynaklandığı sorusudur. Chomsky teorisine göre tamamıyla sözcüklerin dizimlerinden türetilmiştir. Bunun sebeplerine göz atıldığında üretici dilbilgisine (generative grammar) bakmak gerekir.

Üretici dilbilgisi, kuramsal dilbiliminde söz dizimiyle ilgili belirli bir yaklaşımı ifade eder. Bir dilin üretici dilbilgisi, o dildeki dilbilgisel tümce oluşturacak sözcüklerin kombinasyonlarını veren dizileri tahmin etmeye çalışır. Üretici dilbilgisi yaklaşımlarının çoğu, bir tümcenin biçim bilgisel yapısını tahmin etmeye odaklanır. Noam Chomsky, 1950'lerin sonunda gerçekleştirdiği çalışmalarla üretici dilbilgisinin ilk temellerini atmıştır. Kuram, ilk dönemlerinde dönüşümsel dilbilgisi olarak da adlandırılır. Üretici dilbilgisi, şu an bile işlerliğini korumaktadır. Chomsky, sonraki yıllarda bu kuramı olgunlaştırmıştır. 1995'te Minimalist (Yetinmeci) Programı yayınlamıştır. Noam Chomsky, üretici dilbilgisinin pek çok özelliğinin arkasında doğuştancı evrensel bir dilbilgisinin yattığını savunur. Üretici dilbilimciler, dilbilgisinin iletişimsel işlevin sonucundan ortaya çıkmadığını, sadece çevrenin etkisiyle dilbilgisinin öğrenilemeyeceğini iddia ederler. Bu bakımdan, bilişsel dilbilgisi işlevsel ve davranışçı kuramlardan ayrılır. Bu tip dilbilgisinde kural ya da dizim, derin yapı (deep structure) olarak öngörülmüş anlamsal yorumdan tamamıyla bağımsız olarak ele alınır. Böylece tematik roller tümcelerin yapısal gösterimlerinden elde edilirler. Bir başka deyişle, tamamıyla yapısal çevre içine gömüldüklerinden dizimsel kavram olarak da

görülebilirler. Ayrıca bu roller, eyleme ait öğelerin anlamlarına ulaşmada gerekli öbek tiplerin tanımlanması için indislenmiş öğeler olarak ta hizmet ederler. Örneğin *'give (vermek)'* eyleminin biçimsel mekanizmasında üç tane rol mevcuttur. (12) nolu tümceye göre *'give'* eyleminin sözcüksel girişi (lexical entry) Tablo 2.2'deki gibidir [8].

(12) a.$[_S[_{NP}$ Susan$]_i$ gave $[_{NP}$ the food$]_j$ $[_{PP}$to Reggie$]_k]$

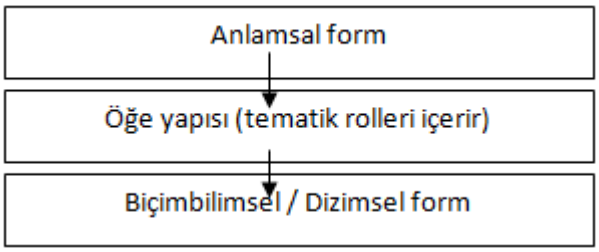
b. Suzan Reyhan'a yiyecek verdi.

Tablo 2.2 *'give'*eyleminin sözcüksel girişi

ETKİLEYEN KAYNAK NP	TEMA NP	HEDEF PP
i	j	k

Tablo 2.2'deki indis işaretlemelerine bakıldığında *'Susan (Suzan)'* ETKİLEYEN / KAYNAK'ın harici rolü olarak atanırken, *'the food (yiyecek)'* TEMA rolü olarak işaretlenir. *'Reggie (Reyhan)'* ise HEDEF rol olarak belirlenir [25].

1970'li yıllardan beri tümce öğe yapılarındaki değişimlerden dolayı tematik roller üzerindeki görüşlerde de değişiklikler olmaktadır. Tümcelerin öğelerini birbirinden ayırmaya yarayan dizimsel kategoriler (isim öbek tipi, eylem öbek tipi, sıfat öbek tipi gibi) ile gelişen öğe yapıları, dilbilgisinde tematik rolleri de barındıran bir konumu ifade eder. Şekil 2.2'de tematik rollerin konumları açık bir şekilde verilmiştir.



Şekil 2.2 Dilbilgisi teorilerinde öğe yapısının rolü

Şekil 2.2'de de görüldüğü gibi, öğe yapıları birçok dilbilimci tarafından da sözcüksel anlam ve dizim ara yüzü olarak ele alınır [74].

Karşımıza çıkabilecek bir diğer soru ise tematik rollerin nereye bağlanacağı ile ilgilidir. Bu sorunun cevabı için iki olasılık vardır. Birincisi, bu rollerin ağaç yapılarının içinde yapısal pozisyonlara bağlanmalarıdır. İkinci olasılık ise dilbilgisel fonksiyonlara bağlanmasıdır. Marantz, Dowty ve diğer birçok araştırmacı rolleri bu fonksiyonlara bağlamışlardır.

Tematik rol teorilerinin tarihi boyunca bütün rollerin aynı çeşit rol olmadığı görülür. Böylece bu roller çok boyutlu (multi-dimension) roller olarak isimlendirilirler. Örneğin, Croft iki farklı rol tipi tanımlamıştır. Bunlar doğrudan ve dolaylı tematik rollerdir. Bu roller bilişsel önem ve olayların nedensel ayrıştırılmasında önemli rol oynarlar. Bu anlamda doğrudan roller normal olarak özne, nesne veya dolaylı nesnelerden oluşur [12].

Jackendoff'un kavramsal yapı düşüncesi ise, HAREKET ve KONUMLA ilgili rolleri içeren tematik katman (thematic tier) ile ETKİLEYEN-ETKİLENEN ilişkisini gösteren rolleri içeren eylem katmanını (action tier) kapsar [11].

Bu anlamda Jackendoff, çok boyutlu yaklaşımı öneren ilk araştırmacılar olan Culiover / Wilkins'in (1986) çalışmalarını benimsemiştir.

Culicover / Wilkins rolleri uzanımsal roller (extensional roles) ve amaçlı roller (intentional roles) diye ikiye ayırmıştır. Uzanımsal roller Jackendoff'un önerdiği tematik katmanına ait rollere, amaçlı roller ise eylem katmanına ait rollere karşılık gelir [28].

Geriye kalan bir başka soru ise tematik rollerin birincil kavramlar olup olmadıklarıdır. Teorilerin dayandığı temel varsayım şudur ki; sözcükler olarak ifade edilen tematik rol kavramları küçük özellikler veya mantıksal yapılar olarak analiz edilirler.

Bu tip yaklaşımların ilkini Jackendoff çalışmalarında göstermektedir. Jackendoff tematik rolleri, DEĞİŞİM ve SEBEP gibi temel notasyonlardan oluşan ayrıştırılmış sözcük parçalarının kavramsal yapılarındaki pozisyonlarından türetir [1].

DeLancey, Goldberg, Van Valin ve Pinker gibi araştırmacılar konumsal ayrıştırma (localistic decomposition) kapsamında anlamın görünüşlerini (aspects of meaning) dilbilgisel olarak ilişkilendirerek anlamsal olarak ayrıştırılmış yapılar ortaya koyarlar [14, 18, 22, 23].

Son yıllarda birçok araştırmacı eyleme bağlı öğeleri anlamsal olarak ayrıştırarak dizimle ilişkili tematik özellikleri yansıtan indis yapısı üzerinde durmaktadırlar. Bu bağlamda, Kiparsky 1997 yılında yaptığı çalışmasında özellikler üzerinden tematik rollerin ayrıştırılmasını gerçekleştirmiştir [53].

Bu anlamda Chomsky Teorisi ile ilişkili tüm bu teorilerde bu roller birincil olarak ele alınırlar. Bu varsayımda kavramlar ya da sözcüklerin gerçek anlamları, eylem-öğe ilişkisinin olası kavramsallaştırmaların evrensel kümesi olan tematik roller aracılığı ile ifade edilir.

1990'lı yıllarda bağlama problemine yönelik kısıtlamalarından dolayı tematik roller ve buna bağlı olarak görünüşsel kavramlar inkar edilmiştir. Dowty, Tenny ve Arad'ın çalışmaları ile görünüşsel özellikler yeniden tanımlanarak bu kısıtlamalar ortadan kaldırılmaya çalışılmıştır. Bu konuda Dowty'nin '*artan TEMA*' kavramı ılımlı bir yaklaşım olarak görülür [4, 54, 55].

Dowty'nin bu görüşüne göre, amaçlı (telic) bir eylem söz konusudur. Nesne konumunda TEMA artan şekilde durumunu değiştirmektedir (change of state). Görünüşsellikle tanımlanan bu kavram Dowty'nin Öğe Seçim Prensibi'nin (Argument Selection Principle) formulasyonu ile yakından ilişkilidir. Özne ve nesne seçim ilkelerinin açıkça ifade edildiği bu prensipte; tümcelerde ETKİLEYEN'e (PROTO-AGENT) ait en fazla özellikleri barındıran öğe eylemin öznesi olarak görülürken, ETKİLENEN'e (PROTO-PATIENT) ait özellikleri taşıyan öğe nesne olarak ifade edilir. Dowty'nin özne / nesne seçimi örnek bir tümce ile açıklanabilir.

(13) a. Chris built a house.

b. Cengiz bir ev inşa etti.

tümcesinde *'Chris (Cengiz)'* özne durumundadır. Sahip olduğu özellikler ise şunlardır: istem, bilinçlilik, nedensellik, hareket, bağımsız var olma. *'a house (ev)'* ise nesne konumundadır. Sahip olduğu özellikler: durum değişikliği, artan TEMA, bağımlı var olma, nedensellikten etkilenen şeklinde açıklanabilir [65].

Dowty'nin bu çalışmalarını temel alan Tenny, Görünüşsel Arayüz Kuramını (Aspectual Interface Hypothesis) geliştirmiştir. Bu kurama göre; tematik yapı ve dizimsel öğe yapısı arasındaki eşleşme görünüşsel özellikler (aspectual properties) tarafından yönetilmektedir. Bu evrensel görünüşsel yapı; dizimsel yapı içerisindeki özne, doğrudan ve dolaylı nesne öğeleri ile ilişkilidir. Tenny'e göre tümcenin öğeleri

meydana gelen olayı ölçer. Bu ölçümü sağlayan ise, eylemin direk nesne pozisyonunda yer alan öğesidir. Bu ölçücü öğe (measurer) olayda meydana gelen durum değişikliğini ve hareketi ölçer. Dolayısıyla, görünüşsel kavram olarak ele alınır. Buradaki amaçlı olay (telic event) belli bir zaman sonunda durum değişikliği ile birlikte son bulur. Durum değişikliğini gösteren ölçücü öğe bir tümce ile örneklendirilebilir.

(14) a. John mowed the lawn.

b. Can çimleri biçti.

tümcesinde durum değişikliğini gösteren ölçücü öğe *'çim (lawn)'* dir. Çimlerin yarısının biçilmesi olayın devam ettiğini, tamamının biçilmesi ise olayın son bulduğunu gösterir.

(15) a. John killed Bill.

b. Can Bekir'i öldürdü.

örneğinde de amaçlı bir olay söz konusudur. Burada da durum değişikliği vardır. Fakat öldürme olayının yarısının gerçekleşmesi söz konusu değildir. Her iki örnekte de görüleceği üzere Tenny'e göre *'the lawn'* ve *'Bill'* ölçücü öğelerdir ve dizim yapısı içinde direk nesneleri ifade ederler. Bununla beraber amaçlı olmayan, durum değişikliği söz konusu olmayan olaylar da (atelic event) vardır. Bu tip tümce yapılarında durum değişikliğini ölçen ölçücü öğe yer almaz.

(16) a. John worked.

b. Can çalıştı.

tümcesinde olduğu gibi [54].

Açıkça görülmektedir ki, Tenny'nin sunduğu bu yapı tematik rollerin dizim içindeki eşleştirmesine kısıtlama getirir. Ama görünüşün bağlama ile yakından ilişkili olduğu da açıktır.

Tenny'nin sonrasında Arad'ın sunduğu radikal yaklaşımda (minimalistik teori) ise tematik roller reddedilir. Arad sadece görünüşsel özelliklerle kural-anlam arayüzünü açıklamıştır. Onun bu çalışmasında görünüş tematik kavramlardan bağımsız, anlam ve kural bağında önemli bir rol oynar [55].

Arad'ın da ifade ettiği gibi Baker'ın sunduğu tek boyutlu tematik rol atama hiyerarşisi-UTAH (Uniformity of Theta Assignment Hypothesis) sözcükler arasındaki tematik rol ilişkilerini gösterir. Bu hiyerarşide merkezi rol yapısal pozisyonda nesne görevini üstlenen TEMA'dır. Yine UTAH hiyerarşisinde iki ayrı rolün aynı pozisyonda

yer aldığı görülür (Örneğin ETKİLEYEN, DENEYİMLEYEN ve SEBEP OLUCU özne pozisyonunda görülebilirler) [56].

Bununla beraber eğer sipesifik bir rol farklı dizi yapılarında gözükürse UTAH hiyerarşisi bozulur. Bu bozulmadan dolayı Arad, öğeleri görünüşsel izdüşümlerin belirtecinde üretir. Ona göre bu öğelerden ilki olay ölçücü (event measurer) olarak ifade edilen ve ismin –i hali ile (accusative case) işaretlenen öğesidir-AspEM (aspect event measurer). Amaçlı eylemlerde yer alan bu öğe eylem tarafından tanımlanan olayın ölçücüsüdür. Bu bağlamda ikinci düğüm olayın başlatıcısı (originator) konumundaki öğesidir AspOR (aspect originator). Bu model içinde AspOR ETKİLEYEN tematik rolü ile eşleşir. TEMA rolü ise AspEM ile eşleşir. Eylemlerin üstlendiği görünüş bilgisi olayın amaçlı mı amaçsız mı olduğunu tanımlar. Amaçlı yani bitmişlik görünüşüne sahip [+telic] eylem [+EM] düğümüne sahiptir. Kılıcılık kavramı ise AspOR olmadan ifade edilemez. Buradan da anlaşılacağı üzere Arad'a göre eylemlere ait öğelerin doğru anlamlarının verilebilmesi için görünüşsel bilginin, eylemin oluştuğu dizimsel yapıda AspOR veya AspEM gibi öğelerle sınırlandırılması gerekir [55].

Sonuç olarak, eyleme bağlı öğelerin biçimbilimsel / dizimsel özellikleriyle birlikte tematik roller üzerinden anlamsal olarak ayrıştırılması bizi kavramsallaştırmaya götürür. Öğelerin bu kavramsallaştırma süreçlerinde, kapsamlı bir sözlüksel kaynağının kullanılması ise zorunludur. Bu anlamda kullanılan en kapsamlı sözlüksel kaynak FrameNet'tir.

2.3. FrameNet

FrameNet anlamsal çerçeveler (semantic frames) üzerine oluşturulmuş leksikografik bir sözlüktür. Sözcüklerin tümce içerisindeki anlamsal ve dizimsel bilgilerini etkileşimli hale getiren geleneksel sözlükten daha kapsamlı bir sözlük olan FrameNet, ilk olarak Berkeley Üniversitesi tarafından geliştirilmiştir [40].

FrameNet bir ontoloji olmadığı için bazı tip ontolojilerle ilişkilendirilmesi yapılarak dil ile bağlantısı sağlanır. SUMO, LF, OMEGA, CYC, DOLCE WordNet, FrameNet gibi sözlüksel kaynakları kullanan belli başlı ontolojilerdendir. OMEGA'nın biçimsel olmayan kavram tanımları içermesi, CYC ve DOLCE ontolojilerinin de WordNet sözlüğünün küçük bir bölümünü eşleştirmesinden dolayı, doğal dil işleme uygulamalarını en iyi şekilde uygulayan LF ve SUMO ontolojileri FrameNet ile ilişkilendirilir. Daha önce yapılan çalışmalarda SUMO ile WordNet sözlüğünün

tamamına yakın bir kısmının eşleştirilmesi mevcuttur. Fakat WordNet, FrameNet gibi kavramların zengin anlamsal detaylarını ve biçimsel tanımlarını içermediğinden daha çok sözcüksel bilgiye sahip olmasından gereken çözümü vermez. Dolayısıyla dilsel açıdan eksik olan FrameNet anlamsal rolleri, biçimsel olarak tanımlanan SUMO sınıfları ile anlamlı hale gelir. Ayrıca bir başka ontoloji tipi olan LF ontolojisi ile de FrameNet çerçeve yapısı arasında bir ilişki kurularak bir konuşma sistemi geliştirilmeye çalışılır.

Bu anlamda SUMO'ya bakılacak olunursa, 4000 tanımlı ifadenin ve 1000 tane sözcüğün biçimsel ontolojisini içeren bir açık kaynak olduğu görülür [48].

Ayrıca daha önceki çalışmalarda, SUMO ile WordNet sözlüğüne ait 100.000 isim eşleştirilmiştir. Bu çalışma tam olarak tamamlanmasa da doğal dil işleme uygulamalarına temel bilgiler sağlaması açısından dikkate değerdir [42].

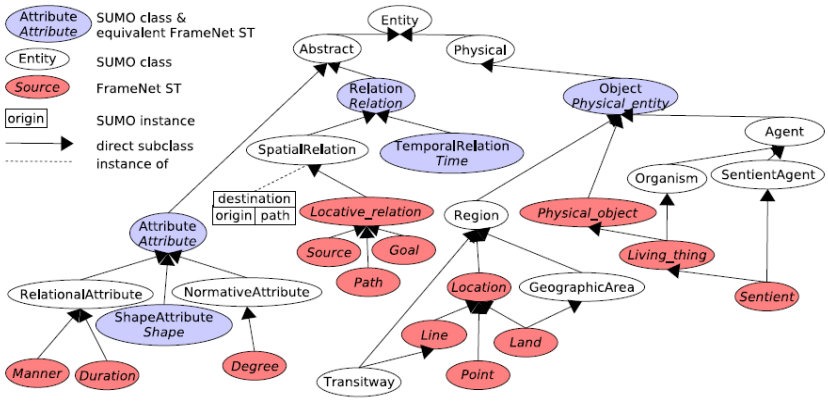
SUMO'nun en önemli özelliği bir takım kuralların kullanılmasıdır. Bu biçimsel tanımlamalar veya kurallar, ontoloji içinde sözcüklerin anlamlarını açık olarak verir. Bu ontoloji tipi açık kaynak ontoloji yönetim sistemine sahiptir. Bu sistem Sigma olarak isimlendirilir [38].

FrameNet ve SUMO birbiriyle uyumlu kaynaklardır. Bu iki kaynağın güçlü yönleri yüksek potansiyelli doğal dil işleme uygulamaları için ilişkilendirilir. FrameNet'i kullanan doğal dil işleme uygulamaları, çerçeve elemanları hakkında bilgiye ihtiyaç duyar. Örneğin, bu uygulamalarda kullanılan anlamsal çerçeve ayrıştırıcısı (semantic frame parser), çerçeve elemanları hakkında tam bir bilgi sağlayan bir metin parçası ister. Bu metin parçası sayesinde çerçeve elemanlarının tam tiplerinin ne olduğu kontrol edilir. Dolayısıyla çerçeve elemanlarının anlamsal tiplerine (Semantic Types-ST) veya anlamsal rollerine ihtiyaç duyulur. FrameNet kırk tane anlamsal tip tanımlar ve bunlar tip hiyerarşisi şeklinde sıralanır. Örneğin ATTACK (SALDIRI) çerçevesi içindeki *'Assailant (saldırgan)'* çerçeve elemanı *'SENTIENT (DUYARLI)'* anlamsal tipine sahiptir. SUMO sınıfları ile karşılaştırıldığında bu anlamsal tipler oldukça sığ kalır. Böylece, FrameNet'teki sözcüklerin anlamsal tipleri SUMO sınıfları ile eşleştirilir. Bu ilişkilendirmenin getirdiği avantajlar ise şunlardır:

- Yapay zeka uygulamaları SUMO'dan gelen bilgileri kullanabilir.
- SUMO alan ontolojileri ile yapılan ilişki ile özel uygulama alanlarında kullanılmak üzere anlamsal tipler elde edilebilir.

- FrameNet çerçeve tabanlı sözlük ve işaretlenmiş tümceler ile SUMO'nun ontolojik bilgileri elde edilir.

SUMO ve FrameNet arasındaki bütün bağlantılar SUO-KIF içinde gerçekleşir. SUO-KIF, SUMO içindeki sınıfları tanımlar. Aynı zamanda bu yapıda yukarıda tanımlanan FrameNet anlamsal tipleri veya rolleri kullanılır. Dolayısıyla, SUMO-FrameNet eşleştirmesi SUMO-WordNet eşleştirmesine göre daha esnek ve nettir. Şekil 2.3, FrameNet anlamsal tip (rol) hiyerarşisinin bir parçasının SUMO ile uyumunu gösterir.[2]



Şekil 2.3 FrameNet anlamsal tiplerin bir parçasının SUMO'ya bağlantıları

SUMO sınıf hiyerarşisi kısmi oranda FrameNet'in anlamsal tip (rol) hiyerarşisinden farklıdır. Şekil 2.3'de de görüldüğü üzere SUMO, soyut (abstract) ve somut (physical) varlıkları birbirinden ayırır. Dolayısıyla bu detayın, SUMO sınıfları ile anlamsal tipleri (rolleri) birbirinden ayırdığı farkedilir. FrameNet'teki bazı anlamsal roller SUMO sınıfları ile eşlenirken, bazı zamanlarda eşleşme olmaz. Bu gibi durumlarda anlamlar SUO-KIF'de yeniden tanımlanır [51].

Yapılan tüm geliştirme çalışmalarına rağmen FrameNet'in kullanıldığı SUMO ontolojisinde kavramlar, sözcüksel bilgi anlamında eksik kalır. Bu nedenle sınıf ve çerçeve elemanları arasında doğru eşleştirmelerin yapılmadığı görülür.

Anlamsal çıkarımlar elde etmede kullanılan bir başka ontoloji tipi ise LF ontolojisidir. Bu ontoloji ile FrameNet verilerine dayanarak konuşmaya dayalı bir

[2] Şekil 2.3 Scheffczyk, J., Pease, A., ve Ellsworth, M., Linking FrameNet to the SUMO Ontology isimli makalesinden alınmıştır.

diyalog sistemi geliştirilmeye çalışılır. Bu sistemde kullanılan TRIPS ayrıştırıcısı anlamsal çıkarımlar elde etmede kullanılan yardımcı bir araçtır. Bu çalışma, sözlük olarak çerçeve yapılarından oluşan FrameNet'i kullanır. Fakat FrameNet sözlüksel kaynağında da sözdizim ve anlam arasındaki ilişkiyi sağlayan bir bağ mevcut değildir. Yine bu sözlükte seçimsel sınırlılıklar da belirtilmemiştir. Mantıksal form gösterimleri için, LF ontolojisinde kullanılan TRIPS her bir sözcüksel giriş için sözdizim-anlam uyuşmasını sağlar. Örneğin *'load (yükleme)'* eylemi ETKİLEYEN-TEMA-HEDEF olarak tanımlanırken;

(17) a. Jim loads the oranges into the truck.

b. Jim portakalları traktörün içine yükler.

tümcesinde özne ETKİLEYEN rolünü, direk nesne TEMA rolünü, '*into (içine)'* önsözcüğü ile belirtilen tamlayıcı HEDEF rolünü alır. Böylece sözdizim-anlam eşleştirmesi sağlanır. Fakat bu eşleştirmede, verilen çerçevedeki anlamsal öğeler gerçek dünya bilgisini yaklaşık olarak verir. Bu anlamda olası belirsizlikler, sözdizimsel yapılara ve seçimsel sınırlılıklara bağlıdır. Dolayısı ile bu ontolojide sözdizim anlam uyuşması seçimsel sınırlılıklar ile olur. Seçimsel sınırlılıklar FrameNet veri tabanının bir parçası olmadığı için burada tanımlanan sisteme seçimsel sınırlılıklar eklenmeye çalışılır. Şekil 2.4'de LF :: Motion (Hareket) ve LF :: Filling (Doldurma) çerçeveleri için LF tip tanımlamaları görülür.[3] LF öğeleri ve çerçeveler arasındaki uyum seçimsel sınırlılıklar aracılığıyla olur [36].

[3] Şekil 2.4'e http://homepages.inf.ed.ac.uk/mdzikovs/papers/lf-framenet.pdf adresinden ulaşılabilir.

```
(define-type LF::Motion
 :sem (Situation (Aspect Dynamic))
 :arguments
   (Theme (Phys-obj (Mobility Movable)))
   (Source (Phys-obj))
   (Goal (Phys-obj))

(define-type LF::Filling
 :parent LF::Motion
 :sem (Situation (Cause Agentive))
 :arguments
   (Agent (Phys-obj (Intentional +)))
   (Goal (Phys-obj (Container +))))
```

Şekil 2.4 LF :: Motion ve LF :: Filling çerçeveleri için LF tip tanımlamaları

Özetle, LF ontolojisi ile FrameNet ilişkisinde detaylı sözdizim altarnetifleri kullanılmaz. Seçimsel sınırlılıklarla birlikte sözdizim yapılarının kullanıldığı sözlük ise VerbNet'tir. LF ontolojisinin daha kullanılabilir olması için TRIPS ile VerbNet arasında bağlantının kurulması gerekir. Yine LF ontolojisinde anlam ile sözdizim arasındaki uyumda kullanılan ETKİLEYEN, HEDEF, KAYNAK, TEMA, SEBEP gibi anlamsal roller FrameNet sözlüğündeki çerçeve elemanlarını yeterli ölçüde karşılamaz. Daha sonraki çalışmalarda LF ontolojisine ait bu eksiklikler giderilmeye çalışılsa da gerekli yeterliliği sağlamadığı görülür.

Verilen bilgilerden de anlaşıldığı üzere FrameNet bir ontoloji değildir. Bir sözlüksel kaynaktır. Çünkü yapısında barındırdığı bütün anlamsal öğeleri dildeki sözcüklerin anlamsal karşılıklarını belirlemek suretiyle oluşturur. Yukarıda bahsedildiği gibi değişik tip ontolojilerle dil ile bağıntısı sağlanmaya çalışılsa da gerekli performans elde edilemez. Daha önce de ifade edildiği gibi ontolojiler, gerçeğin insan zihni tarafından projekte edilmiş halleri olduğundan eğer dilden görece bağımsız anlamsal yapılar aranıyorsa dönüp ontolojilere bakılması gerekir.

2.4. Biçimsel Ontoloji

Genel bir bakış açısıyla, biçimsel bir dil kullanarak tanımlanmış gerçeklik modellerine biçimsel ontoloji denir. Dilin özellikle anlamsal çözümlemesi, gerçekliğin bir modelini kullanmayı gerektirir. Bu anlamda ilk olarak, Özel Uygulama Alanı Ontolojilerin (Domain-Specific Ontologies) tanımı yapılmalıdır [75].

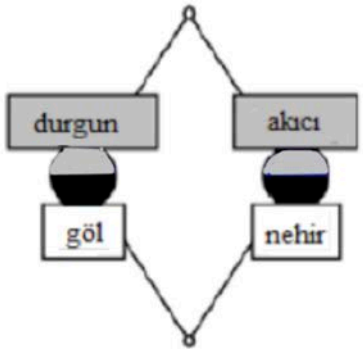
Ontolojiler, kavramların tanımlarını ve bir domain içinde bunların birbirleriyle ilişkilerini ifade eder. Doğaları gereği büyüktürler ve insan emeği ile oluşturulmaları halinde hata ve eksiklikler barındırırlar. Bu nedenle bilgisayar ortamında hazırlanmalıdırlar. Bu da onların biçimsel olmasını gerektirir. Böylece Biçimsel Kavram Analizi Kuramı ile bir gerçeklik modeli yaratma yoluna gidilir [59].

Biçimsel Kavram Analizi, veri kümeleri arasında kavramsal yapıları (conceptual structures) tanımlayan bir veri analizi kuramıdır. Kavramların matematiksel teorisidir. Bir başka deyişle bir kavram hiyerarşisidir. Biçimsel bağlamda verilen bir kavram özelliklerden ve bu özellikleri taşıyan nesnelerden oluşur. Bu nesneler ve özellikler bütün ontoloji tiplerinde ortak olarak paylaşılan özel koşullardır [32, 37].

Bu hiyerarşide bütün kavramlar alt kavram-üst kavram ilişkisi ile sıralanır. Bu yolla sıralanan küme, kavram latisi (concept lattice) olarak isimlendirilir. Bu kavram latisi bir diyagram şeklinde gösterilir. Özellikle dili oluşturan öğeler veya kavramlar bu diyagramda bir düğümü temsil edecek şekilde üst kavramdan alt kavrama çizgiler şeklinde birleştirilmiş olarak yerlerini alır.

Biçimsel Kavram Analizi'nin kullanıldığı bir ontoloji dizaynı aşağıda Hasse Diyagramı olarak bilinen şekiller ve bu diyagramlara ait tablolar şeklinde örneklendirilebilir [31].

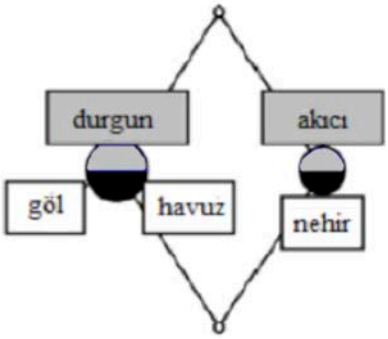
	akıcı	durgun
Göl		X
Nehir	X	



Şekil 2.5 Nesne ve özelliklerden oluşan kavram latisi

Şekil 2.5'deki latisde iki tane kavramın oluştuğu görülür. Bu diyagramda *'akıcı'* ve *'durgun'* özellikleri, *'göl'* ve *'nehir'* nesneleri ifade eder. Aynı örnekte nesnelerin sayısı arttırılırsa Şekil 2.6'daki diyagram elde edilir.

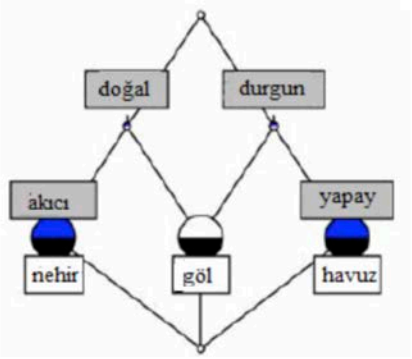
	akıcı	durgun
Göl		X
Nehir	X	
Havuz		X



Şekil 2.6 Eklenen *'havuz'* nesnesiyle oluşan kavram latisi

Şekil 2.6'da görüldüğü gibi *'göl'* ve *'havuz'* nesneleri tek bir kavramı göstermektedir. Bir kavrama iki isimle bir anlam yüklenemeyeceği için bu nesneleri birbirinden ayırmada yeni bir özelliğe ihtiyaç duyulur. Bu özellikler: *'doğal'* ve *'yapay'*dır. Şekil 2.7'de bu ayırım görülebilir.

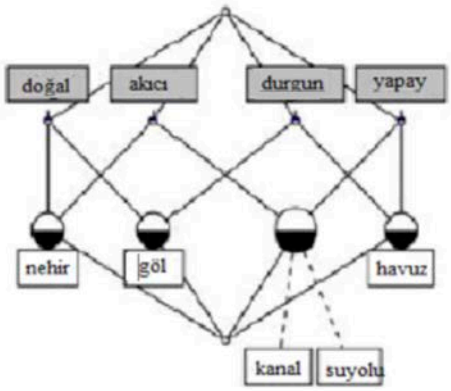
	akıcı	durgun	doğal	yapay
Göl		X	X	
Nehir	X		X	
Havuz		X		X



Şekil 2.7 Yeni kavramların önerilebileceği situasyon

Şekil 2.7'deki durumda da yeni kavramlar tanımlanabilir. Bu şekilde,*'doğal'* - *'yapay'* ile *'durgun'* - *'akıcı'* özelliklerini birlikte taşıyan bir nesne mevcut değildir. Fakat Şekil 2.8'e *'yapay'* ve *'akıcı'* özelliklerini taşıyan *'kanal'* ve *'suyolu'* nesneleri eklenerek bu durum şematize edilebilir.

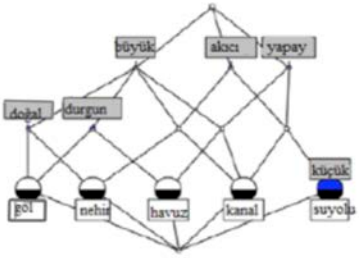
	akıcı	durgun	doğal	yapay
Göl		X	X	
Nehir	X		X	
Havuz		X		X
Kanal	X			X
Suyolu	X			X



Şekil 2.8 İki kavramı birbirinden ayırmaya ihtiyaç duyulan durum

Eklenen *'büyük'* ve *'küçük'* özellikleri ile *'kanal'* ve *'suyolu'* arasındaki ayırım Şekil 2.9'da olduğu gibi grafikselleştirilir.

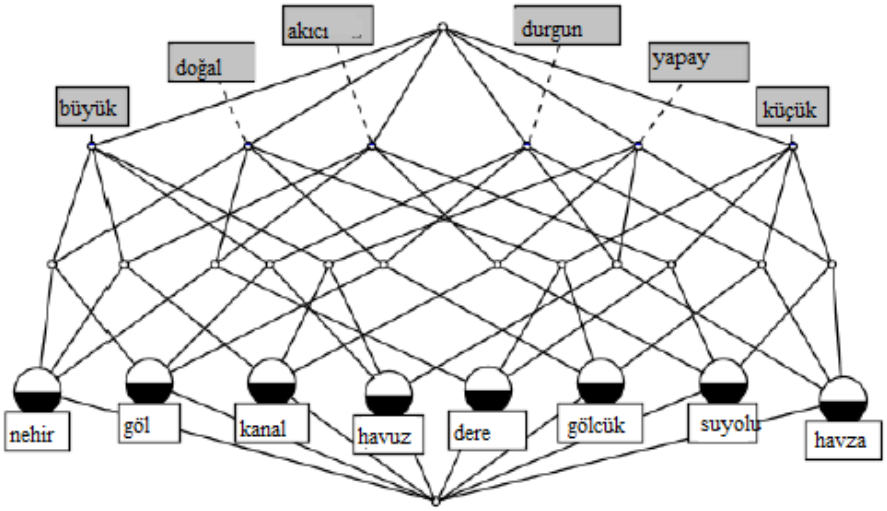
	akıcı	durgun	doğal	yapay	büyük	küçük
Göl		X	X		X	
Nehir	X		X		X	
Havuz		X		X	X	
Kanal	X			X	X	
Suyolu	X			X		X



Şekil 2.9 Kavram latisinin bir parçası

Şekil 2.10'da ise eklenen tüm nesne ve özelliklerden sonra oluşan final ontolojisi görülür.

	akıcı	durgun	doğal	yapay	büyük	küçük
Göl		X	X		X	
Nehir	X		X		X	
Havuz		X		X	X	
Kanal	X			X	X	
Suyolu	X			X		X
Dere	X		X			X
Gölcük		X	X			X
Havza		X		X		X



Şekil 2.10 Tüm özellik ve nesnelerden sonra oluşan final ontolojisi

Dilbilimcilerin de ifade ettikleri gibi ontolojiler ve dil arasındaki ilişki son yıllarda doğal dil işlemenin önemli konularından biridir. Böylece söylemi oluşturan öğeler tarafından kullanılan ontolojiler ifadelerin kavramsal içeriklerinin önemli kaynaklarıdır. Özellikle tematik rollerle birlikte biçimlendirilmiş bu yapılar anlamsal teknolojilerde sözlükle beraber bir model gibi kullanılırlar. Bu bağlamda karşımıza tematik ilişkiler kavramı çıkar. Daha sonra da ayrıntılı bir şekilde anlatılacağı üzere tematik ilişkiler, dilin öğelerinin biçimbilimsel ve dizimsel pozisyonları tarafından kodlanarak elde edilen dilbilgisel ilişkilerinin anlamsal bağıdır. Anlamsal yapılar bağlamında bu ilişkinin içerik kısmı Jackendoff tarafından geliştirilmeye başlanmıştır (1990).

Jackendoff'un teorisinde anlam, birbirinden bağımsız fakat birbiri ile etkileşim halinde olan katmanlar içinde organize edilir. Her bir katman farklı özellikleri ile anlama veya kavramsallaşmaya katkı sağlar [5].

Bugüne kadar yapılan çalışmalarda tematik roller, doğrusal bir hiyerarşiyle gösterilmiştir. Bizim çalışmamızda ise bu roller çok boyutlu katmanlarda yani fraktal latislerde gösterilerek üç boyutlu olarak çalışılmıştır. Böylece bu katmanlarda tanımlanan tematik rollerin çeşitli tip analizleri ile (farklı tematik rollerin aynı pozisyonda oluşmalarına çözüm getirilmesi gibi) sözcüklerin gerçek anlamlarına ulaşılmıştır.

Özellikle FrameNet sözlüğü üzerinden, latislerde gösterilen tematik rollerin, Türkçe'nin anlamsal çözümlenmesine de katkı sağlayabileceği görülmüştür.

BÖLÜM 3

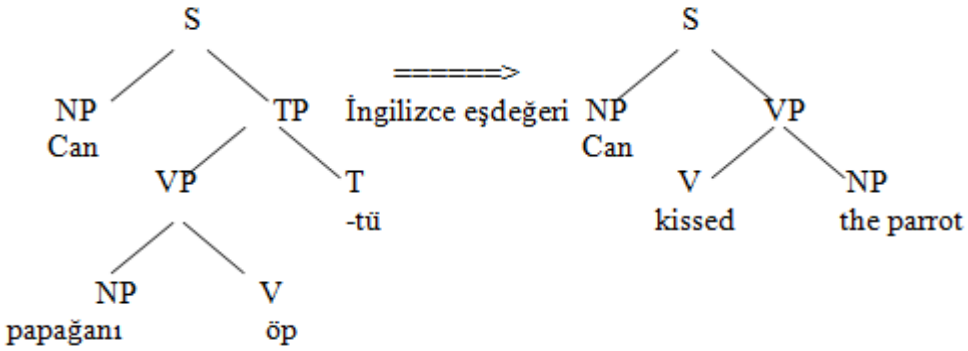
DİL İLE FRAMENET ARASINDA BİR KÖPRÜ OLARAK TEMATİK ROL LATİSLERİ

3.1. FrameNet'in Amorf Yapısı

İngilizce FrameNet modeli anlamsal roller üzerine kurulmuş bir modeldir. Anlamsal roller üzerine kurulu olan bu model amorf bir yapıyı gösterdiğinden buradaki roller doğrudan dil ile bağlantılanamaz. Aslında dilin biçimbilimsel / dizimsel yapısı oldukça katı bir görünüm sunar. Bu görünümde dilin dizimsel yapısını gösteren örüntüler mevcuttur. Dilin dizimsel biçiminin matematiğini Chomsky ağaçlarla göstermiştir. Sözdizimin son derece katı olduğu bu ağaç yapısında X-BAR örüntüsü yayılım gösterir. (18) ve (19) nolu örnek tümcelere ait ağaç yapıları Şekil 3.1 ve Şekil 3.2'de görülebilir.

(18) a. Can papağanı öptü.

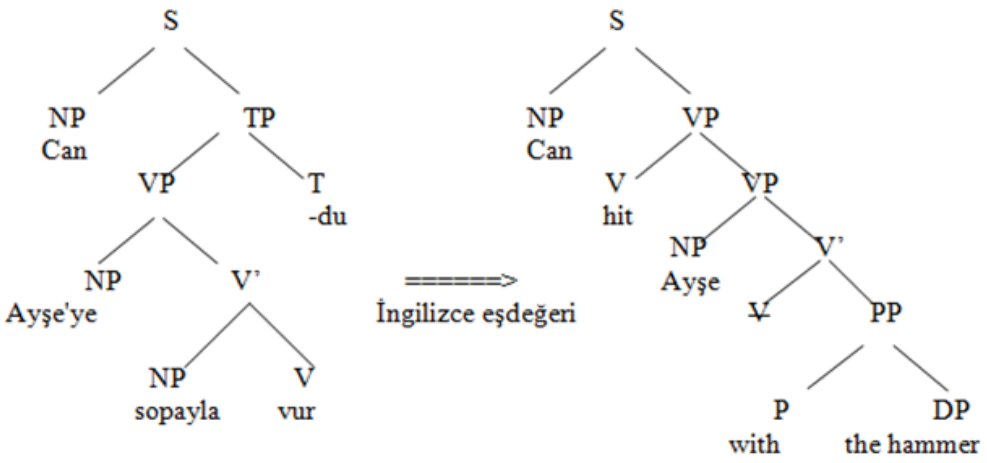
b. Can kissed the parrot.



Şekil 3.1 (18) nolu tümceye ait kural ağacı

(19) a. Can Ayşe'ye sopa ile vurdu.

b. Can hit Ayşe with the hammer.



Şekil 3.2 (19) nolu tümceye ait kural ağacı

Şekil 3.1 ve Şekil 3.2'de gösterilen bu örüntülerde tümcelerin her bir öğesi, bir sözcüksel öbek tipine karşılık gelir. Dolayısıyla tümceyi oluşturan her bir öğe bir tematik rol alırken, her bir rol sadece bir öğeye atanmak zorundadır. Bu da bize dilin katı yapısını gösterir.

Bu anlamda FrameNet'in dil ile ilişkisini gösteren amorf yapısına bakıldığında, iki temel bileşenin varlığından söz edilir. Bunlar: Çerçeveler (frames) ve bu çerçeveleri çağrıştıran sözcük birimleridir (lexical units-LU). Sözcük birimleri bir tümce içindeki yüklem, isim, sıfat ve zarflardan oluşur. Bir tümce içindeki sözcük birimlerine bağlı elemanlara çerçeve elemanları (frame elements-FE) denir. Sözcük birimlerine bağlı çerçeve elemanlarının anlamsal ve biçimbilimsel / dizimsel olarak işaretlenerek aynı tip sözcük birimlerini meydana getirmesiyle sözcük çerçeveleri oluşur [6].

"Çerçeveler, kavramsal yapıların şematik gösterimleridir" [30].

Bir çerçevenin anlamı; birbiriyle ilişkili anlamsal çerçevelerin, anlamsal ve biçimbilimsel / dizimsel özellikleriyle karakterize edilir. Bu anlamsal çerçeveler kavramsal rolleriyle birlikte birer çerçeve elemanıdırlar ve anlamsal öğeler olarak ta isimlendirilirler.

FrameNet; verilen sözcük birimine ait kavramsal yapıların veya çerçevelerin tanımlarını yapmayı, sözcük birimlerin yer aldığı tümceleri derlemden çıkarmayı, bu sözcüklerin yer aldığı çerçevelerin belirlenmesinde tümce işaretlemelerinin yapılmasını ve işaretlemelerin sonunda sözcük birimlerinin kullanım olasılıklarının hesaplanmasını amaçlar [57].

FrameNet modelinin temel bileşenleri olan; çerçeve, sözcük birimi ve çerçeve elemanlarını daha iyi anlamak ve tanımak için mevcut FrameNet sözlüksel kaynağından REVENGE (İNTİKAM ALMA) çerçevesi gözden geçirilebilir [30].

- Çerçevenin ismi REVENGE
- Sözcük birimine eş değer sözcükler: avenge (intikam almak), avenger (intikam alan), revenge (isim), revenge (eylem), get back at (intikam almak), get even with (intikam almak). Bu sözcük birimlerinin oluşturduğu küme bir çerçeveyi yani REVENGE çerçevesini çağrıştırır.
- Bu çerçevedeki en belirgin öğeler AVENGER (İNTİKAM ALAN), PUNISHMENT (CEZA), OFFENDER (SUÇLU), INJURY (ZARAR) dir. Bu öğelerin her biri birer çerçeve elemanıdır.

Görüldüğü üzere, bir derleme bağlı sözlük kaynağı olan FrameNet, çerçeve olarak adlandırılan ve bunların prototip durumlarını içeren bir kavramsal modeli kapsar ki, bu çerçeveler sözcüklerin çağrıştırdığı sınıflar olarak tanımlanabilir [43].

Verilen tümcelerin parçalara ayrılarak her bir parçanın biçimbilimsel / dizimsel ve anlamsal özelliklerinin işaretlenmesi, FrameNet veri tabanının oluşturulması, tümce parçacıklarının anlamsal rollerinin belirtilmesi en genel anlamda FrameNet'in kapsama alanını oluşturur.

FrameNet'te işaretleme, verilen sözcük birimine ait tümcelerin parçalara ayrılarak, her bir parçanın anlamsal, biçimbilimsel / dizimsel özelliklerinin örneklenmesidir. İngilizce FrameNet'ten alınan (20) nolu tümcede örnek bir işaretleme görülür.[4]

(20) a. She never spoke about her feelings.

b. O asla kendi hisleri hakkında konuşmadı.

Çerçeve : STATEMENT (İFADE)

Sözcük birimi : spoke.eylem (konuşmak)

[4] İngilizce FrameNet'e ait örnekler https://framenet.icsi.berkeley.edu/fndrupal/ adresinden ulaşılabilinir.

Çerçeve elemanı : SPEAKER (KONUŞMACI) ==> she (o).isim öbeği

Çerçeve elemanı : TOPIC (KONU) ==> about her feelings (onun hisleri hakkında).isim öbeği

FrameNet veri tabanı, tümcelere ait sözcük birimlerinin tanımını içerir. Geleneksel sözlüklerden farklı olarak çok miktarda bilgi sağlar. Bu veri tabanı, sözcüklerin anlamları altında tümcedeki her bir çerçeve elemanının biçimbilimsel / dizimsel ve anlamsal gösterimlerini tanımlayan karakteristik özellikleri kapsar [44].

Bu veri tabanı üç temel bileşenden oluşur. Birincisi, geleneksel sözlük tipi verisini, sözcüklerin anlamsal ve biçimbilimsel / dizimsel olarak formüle edilen ilişkilerini içeren bileşendir. İkincisi, çerçevelerin basit kavramsal tanımlamalarını ve isimlerini içerir. Üçüncü bileşen ise sözcük birimlerinin anlamsal ve biçimbilimsel / dizimsel özelliklerinin işaretlenmesiyle oluşturulan örnek tümceleri içeren bileşendir [45].

Kısaca bu veri tabanı; çerçeve ve çerçeve elemanları hakkındaki bilgiler ile çerçeveler arasındaki ilişkileri kapsayarak, her bir sözcüğün derlem içindeki kullanım olasılıklarının ve sıklıklarının hesaplanmasına katkı sağlar. Böylece çerçeve elemanlarına bağlı sözcük birimlerinin gruplanması gerçekleşmiş olur [73].

3.1.1. Çerçeve Yapıları

Her bir çerçeve için*;* tümceyi meydana getiren ve çerçeveleri çağrıştıran çerçeve elemanlarının kümesi, çerçeve yapıları olarak ifade edilir [58].

Ayrıca bir çerçeve, ismi ve FrameNet veri tabanındaki biçimsel olmayan tanımlaması ile de ilişkilidir.

Çerçeve elemanları çerçeveler için iki şekilde sınıflandırılırlar. Merkezi çerçeve elemanları (core) ve merkezi olmayan çerçeve elemanlarıdır (non-core). Merkezi çerçeve elemanları, her bir çerçeve için kavramsal gereklilikten dolayı işaretlenmesi gereken elemanlardır. Sözcük birimlerinin bir çerçeveyi çağrıştırabilmesi için bu elemanlarının işaretlenmesi zorunludur. Örneğin REVENGE çerçevesindeki AVENGER rolü işaretlenmesi zorunlu bir elemandır. Merkezi olmayan çerçeve elemanları ise, gerekliliği olmayan elemanlardır. Bir çerçeveyi karakterize eden elemanlar değillerdir. Dolayısı ile işaretlenme zorunluluğu yoktur. REVENGE çerçevesindeki TIME (ZAMAN) elemanı gibi [64].

REVENGE çerçevesine ait çerçeve elemanları Tablo 3.1'deki gibidir.

Tablo 3.1 REVENGE çerçevesinin anlamsal rolleri

REVENGE ÇERÇEVESİ: ÇERÇEVE ELEMANLARI	
MERKEZİ	**MERKEZİ OLMAYAN**
-AVENGER (İNTİKAM ALAN)	-PLACE (YER)
-INJURY (ZARAR)	-TIME (ZAMAN)
-OFFENDER (SUÇLU)	-DEGREE (DERECE)
-PUNISHMENT (CEZA)	-MANAGER (YÖNETİCİ)
-INJURED_PARTY (ZARAR GÖREN)	-PURPOSE (AMAÇ)

Tablo 3.1'deki çerçeve elemanlarına göre işaretlenmiş ve Mevcut FrameNet sözlüğünde yer alan örnek tümceler aşağıda görülmektedir.

(21) a. [They AVENGER] took **revenge** [for the deaths of two loyalist prisoners INJURY]

b. [Onlar İNTİKAM ALAN [iki sadık mahkumun ölümü nedeniyle ZARAR] intikam aldılar.

(22) a. [The next day TIME], [the Roman forces AVENGER] took **revenge** [on their enemies OFFENDER]

b. [Bir sonraki gün ZAMAN], [Roma güçleri İNTİKAM ALAN] [düşmanlarından SUÇLU] intikam aldı.

(23) a. [The ban PUNISHMENT] is [Prince Charles's AVENGER] **revenge** [for her refusal to spend Christmas with the rest of the royals INJURY].

b. [Prens Charles'ın İNTİKAM ALAN] koyduğu [yasak CEZA] [kraliyet ailesinin Christmas tatilini ret edenler ZARAR] içindir.

Aynı şekilde CUTTING (KESME) çerçevesi daha detaylı bir şekilde ele alındığında biçimsel olmayan tanımlaması ise şöyledir:

a. An AGENT cuts an ITEM into PIECES using an INSTRUMENT

b. ETKİLEYEN keser MADDE - yi PARÇALAR halinde ARAÇ - ı kullanarak

Tablo 3.2 CUTTING çerçevesinin anlamsal rolleri

CUTTING ÇERÇEVESİ: ÇERÇEVE ELEMANLARI	
MERKEZİ	**MERKEZİ OLMAYAN**
- AGENT (ETKİLEYEN)	- INSTRUMENT (ALET)
- ITEM (MADDE)	- MANNER (TUTUM)
- PIECES (PARÇALAR)	- MEANS (ARAÇ)
	- PLACE (YER)
	- PURPOSE (AMAÇ)
	-RESULT (SONUÇ)
	- TIME (ZAMAN)

Tablo 3.2'de gösterilen CUTTING çerçevesinin merkezi çerçeve elemanları:

AGENT Bir şeyi dilimleyen kişi

ITEM Dilimlenen şey

PIECES Orijinal şeyin dilimleri

CUTTING çerçevesinin merkezi olmayan çerçeve elemanları:

INSTRUMENT Bir şeyi dilimlere ayıran araç

MANNER Bir şeyin dilimlendiği durum

MEANS Dilimlemeyi başaran kişinin hareketi

PLACE Dilimlemenin gerçekleştiği yer

PURPOSE Bir şeyin dilimlenme amacı

RESULT Bir şeyin dilimlenmesinin sonucu

TIME Dilimlenmenin gerçekleştiği zaman

CUTTING çerçevesine ait sözcük birimleri:

Carve (kesmek), chop (doğramak), cube (küp küp kesmek), cut (kesmek), dice (ufak ufak doğramak), filet (fileto), mince (eti kıymak), pare (kabuğunu doğramak), slice (dilimlemek).

Çerçeve elemanlarının sınıflandırılmasının yanında tümce içinde bazı öğelerin eksikliği de önemli bir konudur. Üç tip eksik kullanım vardır:

- Öznesi olmayan tümceler-CNI (Constructional Null Instantiation)
- Nesnesi olmayan tümceler-INI (Indefinite Null Instantiation)

- Tamlayıcısı olmayan tümceler-DNI (Definite Null Instantiation)

Tümcelerdeki çerçeve elemanlarının eksikliği kavram latislerin analizinde ve tasarımında önemli bir konuyu işaret eder.

3.1.2. FrameNet Çerçeveleri Arasındaki İlişkiler

Eğer çerçevelerin kapsama alanı genişletilmek ve tümcelere ait öğelerin doğru çerçeveleri belirlenmek istenirse, çerçeveler arasında ilişki kurularak bu amaca ulaşılabilir. İngilizce FrameNet'te oluşturulan çerçeveler arasındaki en önemli ilişki kalıtım şeklinde olanıdır. Bununla beraber çerçeve elemanları yani sözcük birimine bağlı öğeler arasında da çerçeveler arasındaki hiyerarşiden dolayı bir ilişki ağı söz konusudur. Çerçeveler arasındaki en önemli ilişki tipleri şunlardır: Kalıtım (inheritence), alt-çerçeve (sub-frame), kullanma (using).

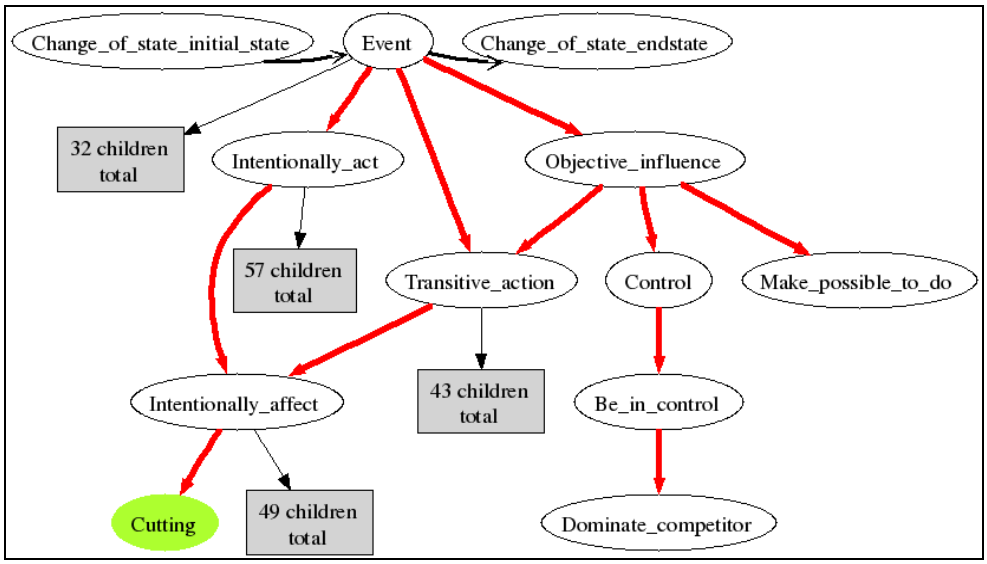
Kalıtım ilişkisinde, bir çerçeve diğer çerçevenin tüm özelliklerine sahip olabilir. Bu ilişki bir IS-A ilişkisidir. Bu ilişki tipi İngilizce FrameNet'e ait örnek bir tümce üzerinden gösterilebilir:

(24) a. Simpson boldly accused him to his face of treason.

b. Simpson küstahça onu ihanetinden dolayı suçladı.

(24) nolu tümce JUDGEMENT-COMMUNICATION (YARGI-İLETİŞİM) çerçevesine bağlıdır. Aynı zamanda bu çerçeve JUDGMENT (YARGI) ve COMMUNICATION (İLETİŞİM) çerçevelerinden miras alır. *'Simpson'* JUDGEMENT çerçevesinden *'COGNIZER (YARGILAYAN)'* anlamsal (kavramsal) rolünü alırken, COMMUNICATION çerçevesinden *'COMMUNICATOR (KONUŞMACI)'* anlamsal rolünü üstlenir. *'of treason (ihanet)'* sözcüğü ise hüküm (karar) için bir sebep olduğundan *'MANNER (TUTUM)'* rolünü alır [44].

FrameNet veri tabanından CUTTING çerçevesinin kalıtım ilişkisine bakılırsa; bu çerçeve hiyerarşik olarak INTENTIONALLY_AFFECT (İSTEMLİ_ETKİ), INTENTIONALLY_ACT (İSTEMLİ_HAREKET) ve TRANSTIVE_ACTION (GEÇİŞLİ_HAREKET) çerçevelerinden miras alır. Bu grafik Şekil 3.3'de gösterilmiştir.



Şekil 3.3 CUTTING çerçevesine ait kalıtım ilişkilerini gösteren kısmın bir parçası

INTENTIONALLY_AFFECT çerçevesinin tanımı aşağıdaki gibidir:

Biçimsel olmayan tanımı:

a. An AGENT causes a PATIENT to be affected, sometimes by a particular MEANS or by use of an INSTRUMENT
b. ETKİLEYEN sebep olur ETKİLENEN - nin etkilenmesine, özel bir TUTUM - la veya ARAÇ kullanarak

INTENTIONALLY_AFFECT çerçevesinin merkezi çerçeve elemanları:

AGENT İsteyerek bir kişinin bir başka şey üzerine etki etmesi (ETKİLEYEN)

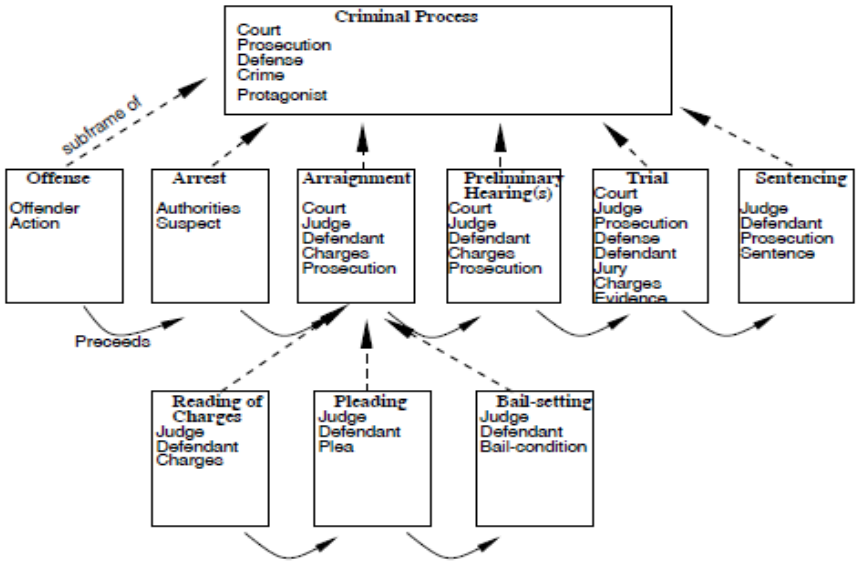
PATIENT Uygulanan güçten veya etkiden etkilenerek üzerinde durum değişikliği yaşanan giriş (ETKİLENEN)

Kalıtım ilişkisi hakkında önemli nokta şudur ki; eğer bir çerçeve başka bir çerçeveden miras alırsa, bu iki çerçevenin çerçeve elemanları arasında bir eşleşme vardır. CUTTING çerçevesi ve INTENTIONALLY_AFFECT çerçevelerinin çerçeve elemanları arasındaki eşleşme çok aşikar bir şekilde görülür. (AGENTS veya INSTRUMENTS arasındaki eşleşme gibi) Bununla birlikte CUTTING çerçevesinin ITEM'i ile INTENTIONALLY_AFFECT'in PATIENT'ı arasındaki eşleşme, PATIENT'in özel bir tipi olan ITEM ile olur. Bir diğer ilginç nokta da, CUTTING çerçevesine ait dilbilgisel tanımlamalardır. Örneğin dünyadaki hiçbir dil CUTTING çerçevesine ait ITEM ve AGENT arasındaki farkı göstermede dilbilgisel yapı

kurallarına sahip değildir. Bununla birlikte soyut bir derecelenmeyi gösteren INTENTIONALLY_AFFECT çerçevesinin tanımlandığı seviye, dilbilgisel yapıların ortaya çıkmasını sağlar. Dünyadaki bütün diller bu soyut seviyede AGENT -PATIENT ayırımını tanımlar. Daha da özelleştirilirse bu roller tematik rol olarak isimlendirilir.

Alt-çerçeve, bir çerçevenin başka bir çerçevenin parçası olduğu durumdur. Her bir parçanın kendini ayrı bir çerçeve olarak tarif ettiği alt-çerçeve ilişkisi, var olan durumları (olayları) ve geçişleri birbirine bağlamak için kullanılır [49].

Alt-çerçeve ilişkisi yine İngilizce FrameNet üzerinden bir örnek üzerinde gösterilebilir.



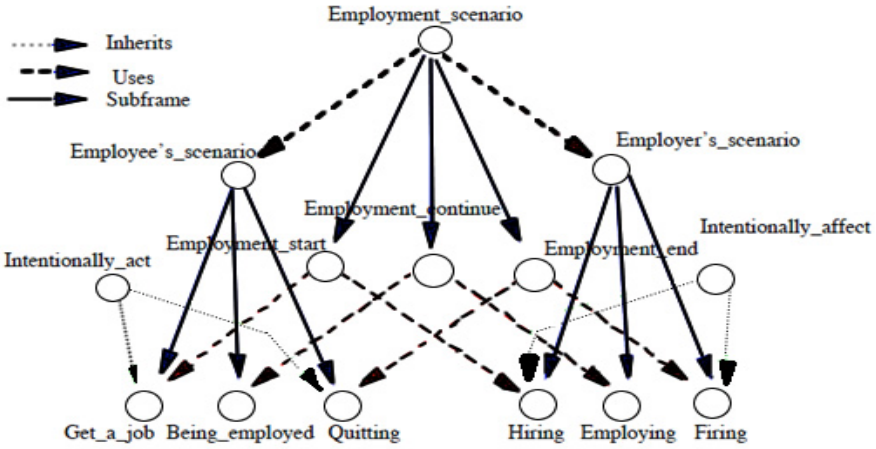
Şekil 3.4 Criminal_Process çerçevesi ve onun alt-çerçeveleri

Şekil 3.4'de görüldüğü gibi, CRIMINAL_PROCESS (CEZA_SÜRECİ) çerçevesi CRIME (SUÇ) domaini içerisinde bir takım alt çerçevelere ve çerçeve elemanlarına sahiptir.'S*USPECT (ŞÜPHELİ)' 'CHARGES (SUÇLAR)'* üzerinden *'AUTHORITY (UZMAN)'* tarafından tutuklanır. Bu *'SUSPECT'* bir *'DEFENDANT (SANIK)'* gibi suçlanır. Bu yüzden *'SUSPECT'* ARREST (TUTUKLAMA) çerçevesi içinde belirtilir. Yine aynı kişi TRIAL (MAHKEME) çerçevesi içinde *'DEFENDANT'* olarak tanımlanır. Görüldüğü üzere ARREST çerçevesi '*apprehend (yakalama)*' eylemi ile ilişkilidir. Böylece FrameNet çalışması *apprehend* eylemi içeren tümceleri toplar ve bu

tümceleri *'AUTHORITIES'*, *'SUSPECT'*, *'CHARGES'* çerçeve elemanları ile işaretler [44].

Kullanma ilişki tipi ise, bir çerçevenin takdir ettiği bir başka çerçevenin bazı sözcük birimlerini kullanması şeklinde tanımlanabilir. Örneğin, İngilizce FrameNet'te EMPTYING (BOŞALTMA) çerçevesi ve REMOVING (UZAKLAŞTIRMA) çerçevesi birbirlerini kullanır. Bu çerçevelerin merkezi ve merkezi olmayan çerçeve elemanları birbirlerine benzer. Örneğin, REMOVING çerçevesine ait sözcük birimlerin yarısı EMPTYING çerçevesi tarafından paylaşılır.

Bu tanımlanan üç ilişki tek bir şekil üzerinde gösterilebilir.

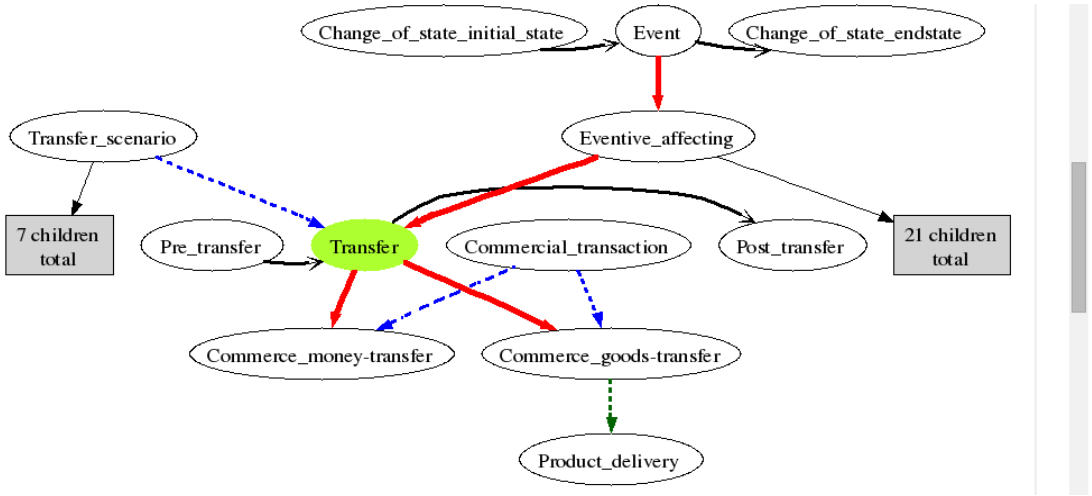


Şekil 3.5 Çerçevelerin ilişki şekilleri

Şekil 3.5'de yukarıda tanımlanan ilişki şekilleri EMPLOYMENT_SCENARIO (İŞ_DURUMU) çerçevesi üzerinden açık bir şekilde ifade edilir [52].[5]

Yukarıda verilen bilgilerden yola çıkarak FrameNet'in bu yapısı sözcüklerin veya kavramların gerçek çıkarımlarını vermez. Bu durum Şekil 3.6'da İngilizce FrameNet sözlüğündeki TRANSFER (AKTARMA) çerçevesi üzerinden ayrıntılı bir şekilde ele alınabilir.

[5] Şekil 3.5'de yer alan çerçeve isimleri https://framenet.icsi.berkeley.edu/fndrupal/ adresinden alınmıştır.



Şekil 3.6 TRANSFER çerçevesine ait ilişkiler şeması

Şekil 3.6'daki şematik gösterimde TRANSFER çerçevesi EVENTIVE_AFFECTING (OLAYSAL_ETKİ) çerçevesinden miras alırken, COMMERCE_MONEY_TRANSFER (TİCARET_PARA_AKTARIMI) ve COMMERCE_GOODS_TRANSFER (TİCARET_MAL_AKTARIMI) çerçeveleri de TRANSFER çerçevesinden miras alır. Tablo 3.3'de ve Tablo 3.4'de ise COMMERCE_MONEY_TRANSFER ve COMMERCE_GOODS_TRANSFER çerçevelerinin çerçeve elemanları verilmiştir.

Tablo 3.3 COMMERCE_MONEY_TRANSFER çerçevesinin çerçeve elemanları

COMMERCE_MONEY_TRANSFER: ÇERÇEVE ELEMANLARI	
MERKEZİ	**MERKEZİ OLMAYAN**
BUYER (MÜŞTERİ)	EXPLANATION (AÇIKLAMA)
SELLER (SATICI)	PLACE (YER)
MONEY (PARA)	MEANS (ARAÇ)
	PURPOSE (AMAÇ)
	TIME (ZAMAN)

Tablo 3.4 COMMERCE_GOODS_TRANSFER çerçevesinin çerçeve elemanları

COMMERCE_GOODS_TRANSFER: ÇERÇEVE ELEMANLARI	
MERKEZİ	**MERKEZİ OLMAYAN**
BUYER (MÜŞTERİ)	EXPLANATION (AÇIKLAMA)
SELLER (SATICI)	PLACE (YER)
MONEY (PARA)	MEANS (ARAÇ)
GOODS (MAL)	PURPOSE (AMAÇ)
	TIME (ZAMAN)

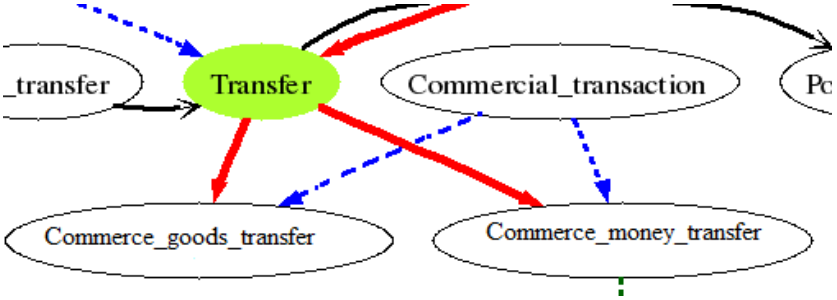
Tablo 3.3 ve Tablo 3.4'de her iki çerçevenin de benzer çerçeve elemanlarına sahip olduğu görülür. Bu çerçevelerin miras aldıkları TRANSFER çerçevesinin çerçeve elemanları ise Tablo 3.5'deki gibidir.

Tablo 3.5 Transfer çerçevesinin çerçeve elemanları

TRANSFER: ÇERÇEVE ELEMANLARI	
MERKEZİ	**MERKEZİ OLMAYAN**
DONOR (VEREN)	EXPLANATION (AÇIKLAMA)
RECIPIENT (ALICI)	PLACE (YER)
THEME (TEMA)	MEANS (ARAÇ)
	PURPOSE (AMAÇ)
	TIME (ZAMAN)

Tablo 3.5'e göz atıldığında ise TRANSFER çerçevesinin çerçeve elemanlarının genelleştiği aynı zamanda miras verdiği diğer iki çerçevenin çerçeve elemanları ile eşleştiği fark edilir.

Bununla beraber Şekil 3.6'ya yeniden dönüldüğünde aynı düzeyde yer alan COMMERCE_MONEY_TRANSFER ve COMMERCE_GOODS_TRANSFER çerçevelerinin yerlerinin FrameNet'in amorf yapısından dolayı rahatlıkla değiştirilebileceği görülür. Şekil 3.7'de bu durum şematize edilmiştir.



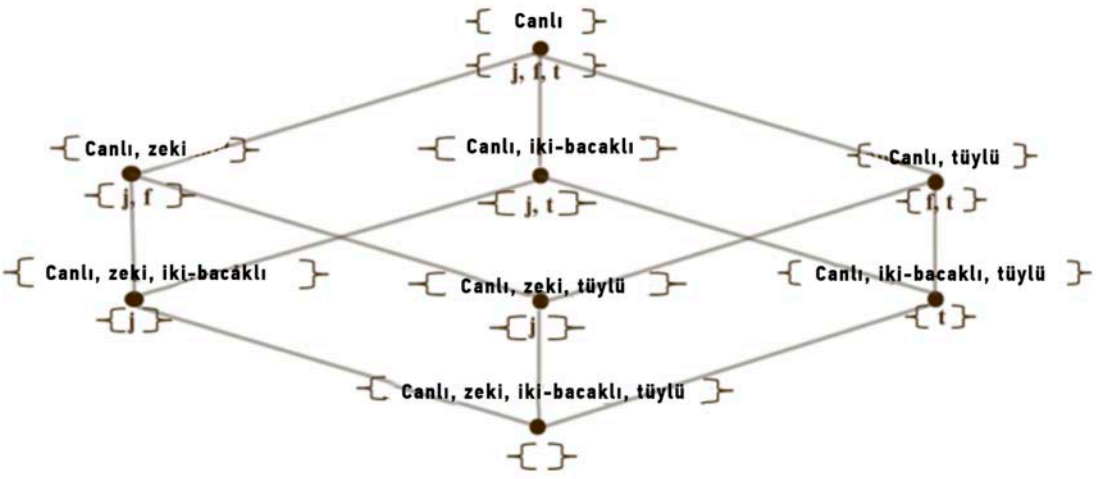
Şekil 3.7 FrameNet'in amorf yapısı

Chomsky tarafından ifade edilen dilin katı yapısı içinde, Şekil 3.7 bize bu hiyerarşide çerçeve elemanları dolayısı ile çerçeveler arasında anlam belirsizliklerinin oluşabileceğini gösterir.

Bu anlamda gereken çözüm tematik rollerle birlikte fraktal latisleri kullanmaktır. Böylece dilin katı yapısı ile FrameNet'in amorf yapısı arasında bir ilişki kurularak dil ile bir gerçeklik modeli olan ontolojiler arasında bağ kurulmuş olur.

3.2. Tematik Rol Latisleri

Tematik rol latisleri, bilgisayarlı dilbilim ya da doğal dil işleme uygulamaları için geliştirilen biçimsel bir dil kullanarak tanımlanmış gerçeklik modelleridir. Bizim tarafımızdan gerçekleştirilen bu gerçeklik modelinin kavram latisi, Şekil 3.8'deki gibidir.



Şekil 3.8 Üç boyutlu örnek kavram latisi

Şekil 3.8'deki örnek kavram latisinde nesneler *Jale (j)*, *Fino (f)*, ve *Tivity (t)*, özellikler ise *canlı*, *zeki*, *iki-bacaklı* ve *tüylü* olarak ifade edilirler.

Oluşturduğumuz bu kavram latisleri ile biçimlendirilen ontolojilerin, bilgisayarlı dilbilim uygulamalarında özellikle makine çevirisi sistemleri için üstlenecekleri görev çok açık bir şekilde görülmektedir. Bu anlamda hedeflenen çeviri sistemi İngilizce ve Türkçe tümcelerin ayrıştırılması için doğal dil işleme modullerini gerektirir. Ayrıştırma için gerekli istatiksel ve kural tabanlı yaklaşımlar ise doğal dil işlemede temel stratejik yaklaşımlardır [60].

Bu anlamda oluşturulan C & C arayüzü, istatiksel öğrenme algoritmalarıyla birlikte temel dizimsel ve anlamsal formüller kullanarak eldeki derlemden o dile ait dilbilgisel kuralların çıkarılmasını sağlayan doğal dil işleme sistemini içerir. Özellikle dilin istatiksel modelinin, doğal dil işleme sistemlerinde yüksek güvenirlilik ve etkili seviyelere öncülük etme gerçeği de unutulmamalıdır. Böylece yüksek ölçekli dil işleme olasılığı da artmaktadır. Bununla beraber Türkçe için yeterli büyüklükte ve işaretlenmiş derlem, istatiksel model için uygun değildir. Böylece Türkçe'den elde edilen veriler, kural-tabanlı mekanizma tarafından kullanılarak işlenmelidir [72].

Böylece C & C arayüzü ile birlikte gelen doğal dil işleme sistemi, Birleşimsel Kategorili Dilbilgisi (Combinatorial Categorial Grammar-CCG) ayrıştırıcısıyla şekillenir [61].

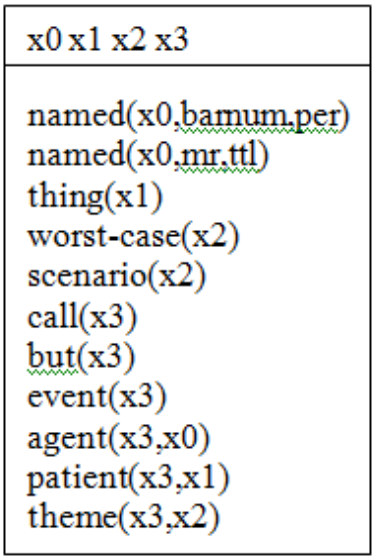
C & C arayüzünün bir diğer bileşeni Boxer ismini alır. Bu bileşen, C & C ayrıştırıcısı ile CCG'den türetilen çıkışları alarak Söylem Gösterim Teorisinin (Discourse Representation Theory-DRT) kutu yapıları gibi anlamsal gösterimler üretir. DRT, Söylem Gösterim Yapısı (Discourse Representation Structure-DRS) olarak bilinir. Ve iki temel bileşenden oluşur:

- Söylem girişleri kümesi
- Söylem girişleri hakkında bilgi veren DRS durumlar kümesi

Boxer İngilizce'nin anlamsal çözümlenmesi için gerekli olan bir bileşen olduğundan (25) nolu İngilizce tümce üzerinden örneklendirilmesi aşağıda görüldüğü gibidir [62].

(25) a. But Mr. Barnum called that a worst-case scenario.

b. Fakat Mr. Barnum bu durumu en kötü durum senaryosu olarak tanımladı.



Şekil 3.9 Boxer tarafından DRS çıkışı

Şekil 3.9'daki DRS çıkışında, tümceyi oluşturan indislenmiş öğeler ve bunların anlamsal gösterimleri görülür. Örneğin x0 tümcenin öznesini, x3 en kötü durum senaryosunu, agent(x3,x0) ise bu iki girişe göre oluşan anlamı verir.

DRS, söylemin konuşmacılar ve dinleyiciler bağlamında yorumların kısmi gösterimleri olarak düşünülebilir. Bilişsel dilbilimcilerin de vurguladıkları gibi söylemin yorumlanması dil formu tarafından tam anlamıyla gerçeklenemediğinden gösterimler kısmi olmaktadır.

Dilin yorumu için gerekli anlam yapılandırmasında bilişsel yaklaşım; birleştirici, ayrıntılı ve kavramsal olmalıdır. Bu görüşe göre anlamın tanımı dilin öğelerinin bir araya getirilmesi kadar basit değildir. Aksine ifadeler, sözcüksel bileşenler, dizimsel konfigurasyonlar anlamın yapılandırılması için gerekli olan temel girişlerdir.

Söylemi oluşturan öğeler tarafından kullanılan ontolojiler ise biçimlendirilmiş yapılardır. Bu yapılar sözlükle beraber bir model gibi kullanılarak anlamsal çözümlemeye katkı sağlarlar.

Bu anlamda Kılıçaslan ve Güner (2010) WordNet'i kullanarak İngilizce dilinin sözlüğü üzerinden bir gerçeklik modeli geliştirdiler. Onlar, tematik rollerle birlikte geliştirdikleri kavram latis üreticisi ile otomatik olarak WordNet'ten ontolojiye dönüşüm gerçekleştirerek Türkçe'nin anlamsal çözümlenmesine yardımcı oldular. Bir başka deyişle Biçimsel Kavram Analizinin kullanılmasıyla bir gerçeklik modeli yaratma

yoluna gittiler. Günümüzde ise bu çalışmalarını, anlamsal roller üzerine kurulu olan FrameNet sözlüğü üzerinden devam ettirmektedirler [63].

Biçimsel ontoloji tabanlı bu gerçeklik modellerinde tematik alanların kullanımına bakmadan önce tematik ilişkilerin neyi ifade ettiği konusuna değinmek gerekir.

Bu anlamda, tematik ilişkilerin yaratıcısı olan Jeffrey Gruber ilk olarak Konumsal Teoriyi (Localistic Theory) önermiştir. Hareket eylemleri üzerinden tematik rollerin analizinin yapılmasından dolayı teori bu ismi almıştır [2].

Gruber tarafından bu anlamda teori tam olarak açıklanamamış fakat Jackendoff, Foley / Van Valin gibi kişiler tarafından geliştirilmiştir. Onların teorilerine göre İngilizce'de tematik roller eylemlerin ön-sözcüksel yapı olarak bilinen ilgeçlerden türetilmiştir. Buradaki ana düşünce kullanılan her tümcenin hareket veya soyut hareket içeren öğe içermesidir. Bu da TEMA rolünü ortaya çıkarmaktadır. Bu durum *'from (-den, -dan)'* ve *'to (-e, -a)'* ilgeçleri tarafından belirtilmektedir. Bir başka deyişle, hareket bir kaynağa (KAYNAK rolüne) ve bir de hedefe (HEDEF rolüne) sahiptir [1, 29].

Gruber bunu basitleştirerek şu şekilde ifade etmiştir:

(26) a. Peter SOURCE sends the parrot THEME to the pope GOAL.

b. Peter KAYNAK gönderir papağanı TEMA papaya HEDEF

Tümce şu şekilde de ifade edilebilir:

(27) a. The parrot THEME goes from Peter SOURCE to the pope GOAL.

b. Papağan TEMA gider Peterden KAYNAK papaya HEDEF

Örnek tümcelerde görüleceği üzere Gruber Teorisinde YAPAN *'from'* ön-sözcüğünden türetilmiştir [2].

Tematik ilişkiler resmi olarak 1972 yılında tanımlanmıştır ve dilbilgisinin kavramsal yapısı içinde daha sonraki çalışmalarda kavramsal bilgi olarak yerini almıştır.

Tematik ilişkiler bağlamında ilk olarak açık bir şekilde tematik hiyerarşi, Gruber'in tematik rol listesine bağlı olarak Jackendoff tarafından bulunmuştur. Bu tematik hiyerarşi aşağıda gösterilen rollerden oluşur.

- ETKİLEYEN
- KONUM, KAYNAK, HEDEF
- TEMA

Bu hiyerarşiye bağlı olarak Jackendoff, Tematik Hiyerarşi Durumu'nu (ETKİLEYEN > KONUM, KAYNAK, HEDEF > TEMA) şeklinde formüle etmiştir [1].

Jackendoff Teorisindeki tematik rollerin kullanımı Tematik Hiyerarşi Durumun'daki gibi genellemelerin yapılması açısından oldukça uygun bir yoldur. Jackendoff teorisinde tematik roller birincil kavram olarak alınmamasına rağmen bu roller konumsal ve nedensel yapıların ayrıştırılmasında önemli bir yer tutar. Bu kavram 1983, 1987 ve 1990'lı yıllarda önemli olabilecek atılımlarla çalışılmıştır. Jackendoff teorisinde konumsal ayrıştırma, biçimlendirme kuralları tarafından kavramsal kategorilerin terimlerinde yeniden tanımlanmıştır. Buna göre TEMA rolü GO (GİTMEK), BE (OLMAK) gibi fonksiyonlarının birinci öğesi olurken, KAYNAK *'from'* 'un öğesi, HEDEF *'to'* 'nun öğesi olur. Teşvik edici ise SEBEP OLMANIN birinci öğesi, etki ise ikinci öğesi olur.

Jackendoff'un tematik hiyerarşiye dayalı olarak geliştirdiği kavramsal yapı basitçe şu şekilde ifade edilir:

(28) a. [ETKİ(X,..)] (AKTÖR)
b. [ETKİ(..,Y)] (ETKİLENEN, YARARLANAN)
c. [OLAY / DURUM] F(X,.. (TEMA))
d. [YOL / YER F(X)] (KONUM, KAYNAK, HEDEF)

(29) nolu tümce Jackendoff'un çalışmalarından bir örnektir.

(29) a. Peter sent the pope a parrot.
b. Peter papaya bir papağan gönderdi.
c. [CAUSE ([PETER]$_i$, [GO([PARROT]$_j$, FROM[PETER] TO [POPE]$_k$)])]

tümcesindeki *'send (göndermek)'* eyleminin kavramsal yapıdaki basitleştirilmiş halini (29c)'de göstermek gerekirse; *'Peter'* KAYNAK ve ETKİLEYEN, *'a parrot (papağan)'* TEMA ve *'the pope (papa)'* ise YARARLANAN ve HEDEF rollerini üstlenir. Bunun tam tersi tümce şu şekilde olursa;

(30) a. Peter sent a parrot to the pope.
b. Peter bir papağanı papaya gönderdi.

Eylem katmanındaki değişken YARARLANAN olmayacağından *'the pope'* tematik katmanından sadece HEDEF rolünü alır. Buradan da anlaşılacağı üzere; her iki tümce versiyonunda, YARARLANAN'ın kavramsallaştırılmasındaki farklılıklardan dolayı anlamsal belirsizlikler görülür. Bu bilgilere ek olarak,*'to'* ön-sözcüğünden dolayı (30)

nolu tümcede *'send'* eyleminin başarısının NEDENSELLİKTEN olmadığı görülür. Bununla birlikte *'send'* eyleminin NEDENSELLİKTEN gerçekleşmesi (29a) nolu örnekteki çift nesne kullanımı ile gerçekleştiği dikkat çekilir. Bu da zenginleştirilmiş kavramsal yapıdan dolayı Jackendoff'un öğe ardalanmasının yeterli düzeyde olduğunu gösterir [5].

Jackendoff'un bu kavramsal yapısı rakipsiz görünmesine karşın, bünyesinde birçok problemi de barındırır. Bu yapıda, hem geçişsiz hem de öznesi yüklem öbeğinde olan eylem öğelerinin yer değiştirmesi ile ilgili bir bilgi bulunmazken bir öğenin birden fazla rol almasına yönelik çıkan anlamsal belirsizliklere yer verilmez. Jackendoff Teori'sindeki bir diğer problem ise buradaki yaklaşımların sadece İngilizce için geçerli olmasıdır. Türkçe'nin durum eklerine göre çözüm getirmesi de mümkün değildir.

1970 ve 80'li yıllar boyunca anlamsal roller üzerine çalışan Charles Fillmore ise çalışmalarını İngilizce FrameNet modeli altında, çerçeve anlamları (frame semantics) üzerinden götürmektedir. Bu çerçeve anlamları, dilsel analiz sistemi olan durum dilbilgisi (case grammar) üzerinden gelişir. Bu dilbilgisi eylemin değerliği (özne ve nesnelerin sayısı) ile dilbilgisel bağlam arasındaki ilişkiye odaklanır. Bu teori, tümcelerin derin durumları (deep cases) ile yüzeysel olarak dizimsel yapısını analiz eder. Fillmore'a göre ETKİLEYEN, OBJE, FAYDALANAN, KONUM veya ARAÇ eylem tarafından istenenlerdir. Örneğin İngilizce'deki *'give'* eylemi ETKİLEYEN, OBJE ve FAYDALANAN rollerini ister. Fillmore bu sistemde eyleme ait durum çerçevesini (case frame) şekillendiren derin durumların (anlamsal rollerin) sayısının belirlenmesi gerektiğini söyler. Böylece durum çerçevesi; anlamsal değerliğin, eylemlerin, sıfatların ve isimlerin önemli görünüşlerini tanımlar. İngilizce tümcelerde durumların bazıları zorunlu olurken, bazıları seçimliktir. Zorunlu işaretlenmesi gereken durumlar silinmemelidir. Aksi takdirde dilbilgisine uymayan tümceler meydana gelir.

(31) a. Mary gave the apples.

b. Meryem elmaları verdi.

İngilice tümcesinde zorunlu işaretlenmesi gereken durum olan (ETKİLEYEN) *'Mary'* silinirse, dilbilgisine uymayan bir tümceyle karşılaşılır. Ayrıca bir eylemin durum çerçevesi ETKİLEYEN içerirse bu rol aktif tümcenin öznesi olarak algılanır [15].

Georgetown Üniversitesinde dilbilim profesörü olan Walter A. Cook, S. J. Fillmore'un orijinal çalışmasını takip eden durum dilbilgisi teorisyenlerin en başında

yer alan kişi olmuştur. Cook, 1970'lerin başından 1990'lara kadar yaptığı akademik çalışmalarda durum dilbilgisini dilsel analizde, dil öğretme metodunda ve diğer uygulamalarda bir araç gibi geliştirmeye çalışmıştır. Ona göre durum dilbilgisi kuramı Anlamsal Değerlik Teorisidir (Semantic Valence Theory). Bu teori eyleme ait öğeler üzerinden tümcenin mantıksal formunu tanımlar. Ve bu öğeler ETKİLEYEN, OBJE, KONUM, KAYNAK, HEDEF olarak etiketlenmelidir [68].

Bu anlamda Fillmore'un durum dilbilgisi üzerinden geliştirdiği yapı dilbilgisi ile Jackendoff'un tematik analizleri kıyaslandığında; Fillmore'un yaklaşımı, bir eylemin aldığı perspektife dayalı olarak kural içinde nasıl kavramlaştırılarak ifade edileceğine dayanır. Fillmore'a göre *'sell (satmak)'* ve *'buy (satın almak)'* eylemlerinin perspektifleri z'nin x'den y'ye transferiyle birlikte w'nin y'den x'e transferini ifade eder. Ele alınan bu perspektif durum çerçevesi içindeki bir takım durum ve haller ile bağımlıdır. (32) nolu örnekte bu durum açık bir şekilde görülür [34].

(32) a. John sold the book to Mary.

b. Can kitabı Meryem'e sattı.

c. Mary bought the book from John.

d. Meryem kitabı Can'dan satın aldı.

Jackendoff'un tematik analizleri ise bir eylemin tüm görünümlerini kapsadığı için Fillmore'un bu düşüncesi Jackendoff'un tematik analizleriyle örtüşmez.

Türkçe'nin anlamsal çözümlenmesine katkısı açısından bizim geliştirdiğimiz çalışma ise; Fillmore ve arkadaşlarının geliştirdiği durum çerçevesi içinde öğelerin görünüşlerini tutan Framenet sözlüksel kaynağını kullanarak, kavramsallaştırmada da Jackendoff'un tematik analizlerini dikkate alarak fraktal latislerle bir gerçeklik modeli yaratmayı hedefler. Böylece Jackendoff'un tematik çalışmalarında ortaya çıkan belirsizlik ortadan kalkarak, dil ile FrameNet sözlüğü arasındaki matematiksel köprü inşa edilmiş olur.

3.2.1. Latislerde Tematik Alanlar

Tematik rol teorisinin en önemli görevi dizimsel olarak birbirinden farklı aynı iki tümcenin tematik rol atamalarının doğru bir şekilde nasıl yapılacağını göstermektir. Biçimsel Kavram Analizini kullanmak suretiyle yapısal latislerde yapılan ayrıştırma ile birden fazla rolün aynı düğümde yer alması sağlanarak bu problem çözüme kavuşur.

Biçimsel Kavram Analizi ilk defa Ganter ve Wille (1996) tarafından geliştirilmiştir. Biçimsel bağlamda verilen veri, daha öncede belirtildiği gibi iki kümeden ve bu iki kümenin ilişkisini gösteren ifadeden oluşur. Kümelerden birisi özellikleri barındırırken, diğeri bu özellikleri taşıyan nesneleri içerir. Bu şekilde elde edilen latisler, verileri hiyerarşik bir organizasyon içinde modellenmesine imkan verir [39].

Bu hiyerarşik organizasyon, iki temel katman üzerinde oluşturulur (tematik katman ve eylem katmanı). Tematik roller ise bu katmanların temel taşlarını meydana getirir [6].

Gestalt psikolojisi de dikkate alındığında anlam, FİGÜR (FIGURE) ve YER (GROUND) ikilisi içinde oluşur. Buradaki FİGÜR terimi, çalışmamızda TEMA rolü olarak (yer değiştiren şeklinde) ele alınarak tüm bu roller üzerinden çalışan rol olarak görev yapar. Dolayısıyla merkezi rol olarak çalışır. Bu anlamda TEMA rolü eylem katmanında da gösterilebilir. (AKTÖR ve ETKİLENEN, TEMA rolünü üstlenebilir.) Bununla birlikte daha önce bahsedilmeyen DENEYİMLEYEN ve ALGI rolleri de aynı figüral ve konumsal roller gibi çalışmamızda ele alınır. Bu rollere psikolojik roller de denilir. Bu rollerin işaretlenmesi şu şekilde örneklendirilir:

(33) Hizmetçi DENEYİMLEYEN bir fare ALGILANAN gördü.

(34) Hizmetçi ETKİLEYEN doğradı.

(34) nolu tümcede ‘*Hizmetçi’* AKTÖR rolünü üstlenirken istemli olarak bir işin yapılması söz konusu olduğundan (DENEYİMLEYEN rolüde geçerli olacağından) AKTÖR ETKİLEYEN rolüne dönüşür.

(35) O DENEYİMLEYEN hizmetçiyi AKTÖR bıçakla ARAÇ soğanı ETKİLENEN doğrarken gördü.

tümcesinin öznesi DENEYİMLEYEN rolünü alırken, ‘*hizmetçi’* AKTÖR, ‘*soğan’* ETKİLENEN, ‘*bıçak’* ise ARAÇ rolünü alır.

Özetlemek gerekirse bizde tematik roller üç alt gruba ayrılır:

{

{KONUM, KAYNAK, HEDEF, YOL}, {TEMA, AKTÖR, ETKİLENEN, ARAÇ}, {DENEYİMLEYEN}

}

Buradan da anlaşılacağı üzere, gruplandırma figüral roller (figural roles), konumsal roller (locative roles) ve psikolojik roller (psychological roles) diye gerçekleşir.

Gruber'in Konumsal Ayrıştırma Teori'sinden türetilen prensiple konumsal tematik roller ayrıştırılarak daha da genelleştirilebilir. Böylece yapısal latislere aktarılabilir. Buna göre, bütün konumsal roller uzamsal konum (spatial location) ile ilişkilidir. Her bir konumsal rol uzamsallık içeren grup içinde ayrıştırılmalıdır.

KONUM ={UZAMSAL, KONUM}

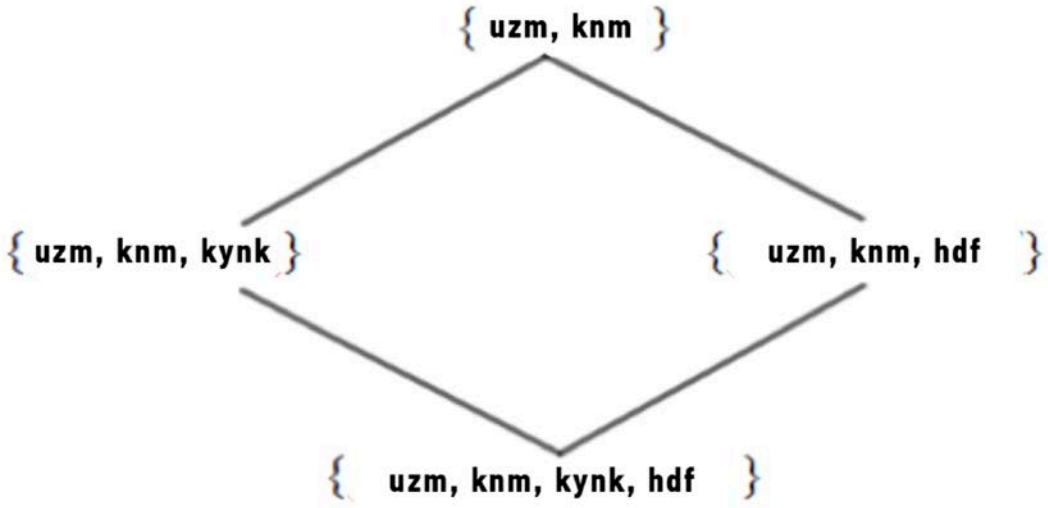
KAYNAK ={UZAMSAL, KONUM, KAYNAK}

HEDEF ={UZAMSAL, KONUM, HEDEF}

YOL ={UZAMSAL, KONUM, KAYNAK, HEDEF} olur.

Matematiksel gösterim için uzamsal konumlanma alanı (tematik katman) şöyle ifade edilir:

{{UZAMSAL, KONUM },
{UZAMSAL, KONUM, KAYNAK }
{UZAMSAL, KONUM, HEDEF },
{UZAMSAL, KONUM, KAYNAK, HEDEF}}



Şekil 3.10 Uzamsal konum latisi

Şekil 3.10'daki uzamsal konum latisinde roller; uzamsal (uzm), konum (knm), kaynak (kynk), hedef (hdf) dir.

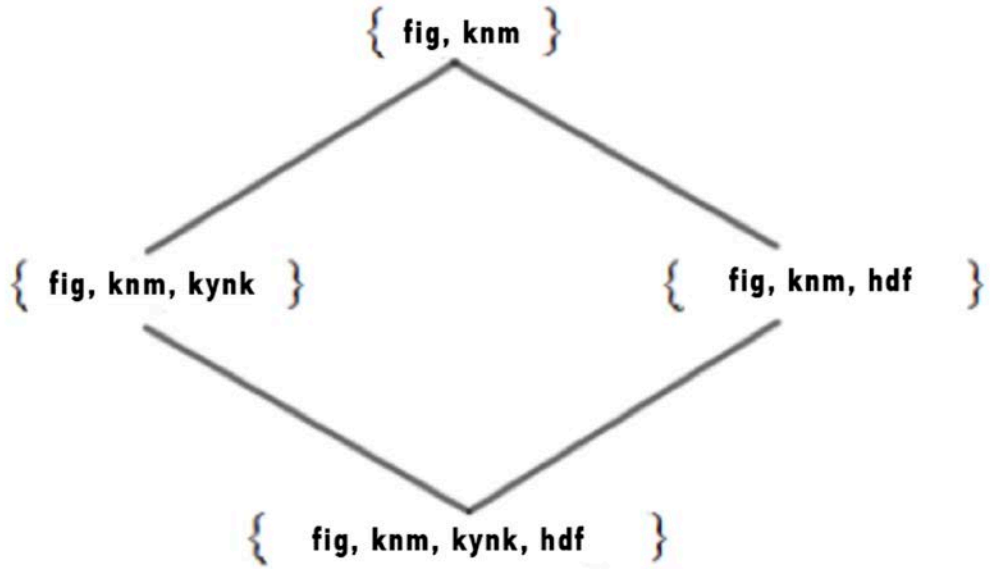
Eylem katmanına ait figüral roller ise herhangi bir yere konumlanan yani biçimsel (figüral) rollerdir. Dolayısıyla bu katmanı tanımlayabilmek için birincil giriş *figüral (fig)* olmalıdır. Bu alanın tüm rolleri figüral girişe göre ayrıştırılmalıdır.

TEMA ={FİGÜRAL, KONUM}

AKTÖR ={FİGÜRAL, KONUM, KAYNAK}

ETKİLENEN ={FİGÜRAL, KONUM, HEDEF}

ARAÇ ={FİGÜRAL, KONUM, KAYNAK, HEDEF}



Şekil 3.11 Figüral konum latisi

Şekil 3.11'de de görüldüğü gibi konumsal rollerle figüral roller arasında bir paralellik kurulmuştur. Bu roller içinde ARAÇ rolünün karakteristik özellikleri ise oldukça ilginçtir.

- AKTÖR bir eylemi ARAÇ ı kullanarak gerçekleştirir.
- AKTÖR ARAÇ üzerinde eylemi gerçekleştirir.
- ARAÇ ETKİLENEN gibi davranır.

Dolayısıyla burada AKTÖR-ARAÇ/ETKİLENEN ikili durumu olduğundan

ARAÇ ={FİGÜRAL, KONUM, KAYNAK, HEDEF} gibi ele alınır.

Son olarak psikolojik alana ait roller için, Gruber'in (1965) yaptığı *soyut geçişin bilgisel teması* tanımına benzer bir rol tanımlanması yapılır [2].

Aşağıdaki tümcede de bu açık olarak görülür.

(36) Can dünyanın düz olduğunu Bekir'den öğrendi.

(36) nolu tümcede *dünyanın düz oluşu* bilgisinin KAYNAK'tan (Bekir'den) HEDEF'e (Can) doğru geçişi vardır. Bir başka deyişle izah edilmek istenen, bilginin soyut bir yol üzerinden bir zihinden başka bir zihne geçişidir. Dolayısıyla;

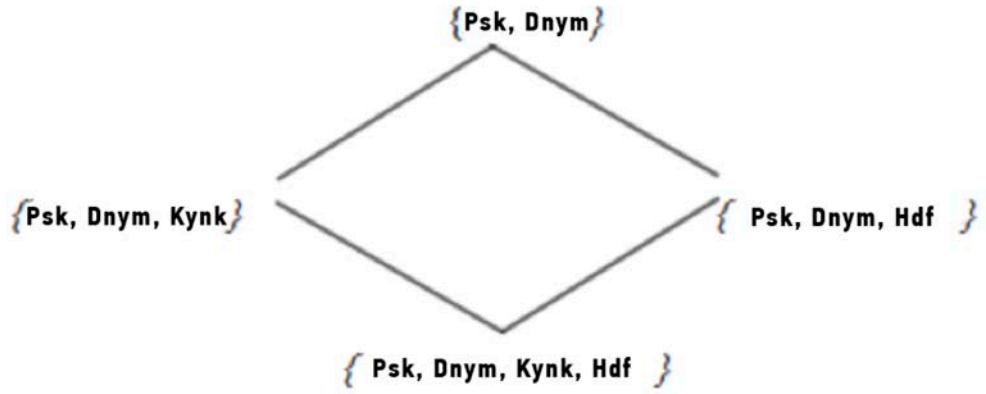
KONUM ={PSİKOLOJİK, DENEYİMLEYEN}

KAYNAK ={PSİKOLOJİK, DENEYİMLEYEN, KAYNAK}

HEDEF ={PSİKOLOJİK, DENEYİMLEYEN, HEDEF}

YOL ={PSİKOLOJİK, DENEYİMLEYEN, KAYNAK, HEDEF} olur.

Psikolojik alanın hiyerarşik organizasyonu Şekil 3.12'deki gibidir.



Şekil 3.12 Psikolojik alanın hiyerarşik organizasyon latisi

Şekil 3.12'deki psikolojik roller, psikolojik (psk), deneyimleyen (dnym) şeklinde ifade edilir. Ayrıca her konumsal rolün psikolojik rol içeren grup içinde ayrıştığı görülür.

3.2.2. Referans Parametreleriyle Zenginleştirilmiş Tematik Rol Yapıları

Tematik rol yapılarını yeni parametreler ekleyerek daha da zenginleştirmek mümkündür. Böylece bu roller yeniden organize edilirler.

3.2.2.1. Konumsal Referanslar

Aristo'nun da içinde bulunduğu birçok filozof konumların varlıklardan ontolojik olarak ayrılmaları gerektiğini işaret etmişlerdir [35].

Bizde ise, *konumlar* ve *varlıklar* arasında ontolojik farka karşılık gelen ve dilbilgisi üzerinde işlevliği olan *salt konumlar (mere locations)* ve *konumsal / lokatif referanslar (locative references)* arasındaki farkı çizen bir çalışma mevcuttur.

Konumlara referans vermede *'içte', 'sağda', 'burada', 'orada', 'yerde', 'alanda'* gibi isim grupları kullanılır. Buradaki konumsal referanslar; Jackendoff'un *'referanslarından (references)'*, Langacker'in *'işaretlerinden (landmarks)'*, Talmy'nin *'yerlerinden (grounds)'* farklı değildir [69, 24, 26].

Konumsal referanslar aşağıdaki örnek tümcelerle açıklanabilir.

(37) Fare $_{\text{TEMA}}$ kutunun içinde. $_{\text{KONUM}}$

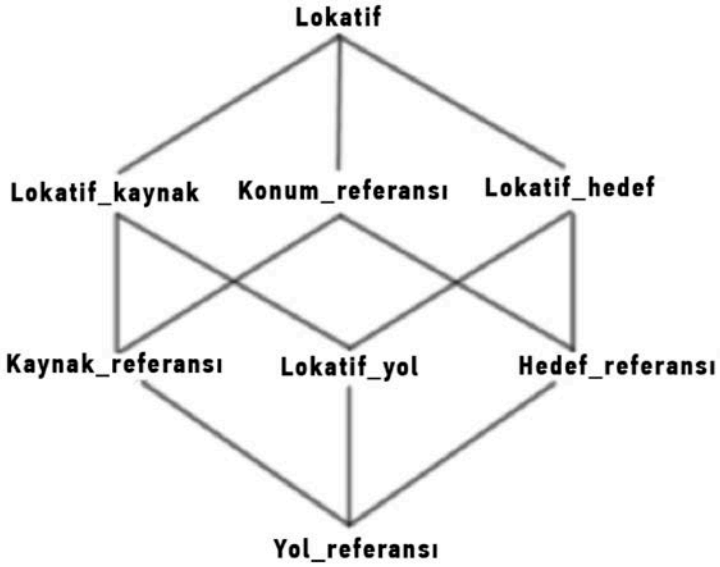
Bu ifade de *'kutunun içinde'* konumla ilgili olan konumsal referansı gösterirken, *'fare'* nin bulunduğu konumu sözcükleştirir.

(38) Fare hala aynı yerde fakat kutu her nasılsa oradan kayboldu.

(38) nolu tümcede ise, konum onu kuşatan kutudan ayrılmıştır. (37) nolu örnekte *'kutu'* konumun referansıdır. Benzer şekilde, *'hizmetçiyi geçme', 'masanın altından'* ve *'vazonun sağına doğru'* ifadeleri (39) nolu tümcede sırasıyla YOL'un referansı, KAYNAK'ın referansı, HEDEF'in referansı olur.

(39) Fare masanın altından vazonun sağına doğru hizmetçiyi geçti.

Üretilen bu model için referans roller şu şekilde ifade edilir: *Konum _referansı, Kaynak_ referansı, Hedef_referansı* ve *Yol_referansı.* Şekil 3.13'de bunlar açık bir şekilde görülür.



Şekil 3.13 Konumsal referanslar

Şekil 3.13'de gösterilen referanslarla oluşturulan yapı konumsal alanı ifade eder.

3.2.2.2 Figüral Referanslar

Situasyonların matematiksel kuramına göre; bir situasyon, bilişsel olarak bir kişinin algıladığı dünyadır [21].

Cooper'a göre, situasyonlar organizmalar tarafından algı kapasitesi ile doğrudan algılanırlar [20].

Bize göre varlıklar, bilişsel sürecin veya algının sonucunu kodlayan soyut nesnelerdir. Daha da biçimselleştirirsek; situasyonlar bu kuramın birinci derece nesneleridir. Algılayan için bir situasyon, bir varlığı tanımlayan bu kişi için bir referans gibi hizmet eder.

Bu situasyonel referansların dilbilgisinde nasıl geçerli olabildiği sorusuyla da karşı karşıya kalınabilir. Diller bu sorunun cevabı için destekleyici kanıtlar sunarlar. Bu kanıtlardan en ilginci Türkçe'deki ismin –i hali morfeminden gelir. Aşağıdaki örnekte görüldüğü gibi, Türkçe tümcenin direk nesnesi bu morfemi taşıyabilir veya taşıyamaz.

(40) a. Oya bahçe-de bir köpek gör-üyor.

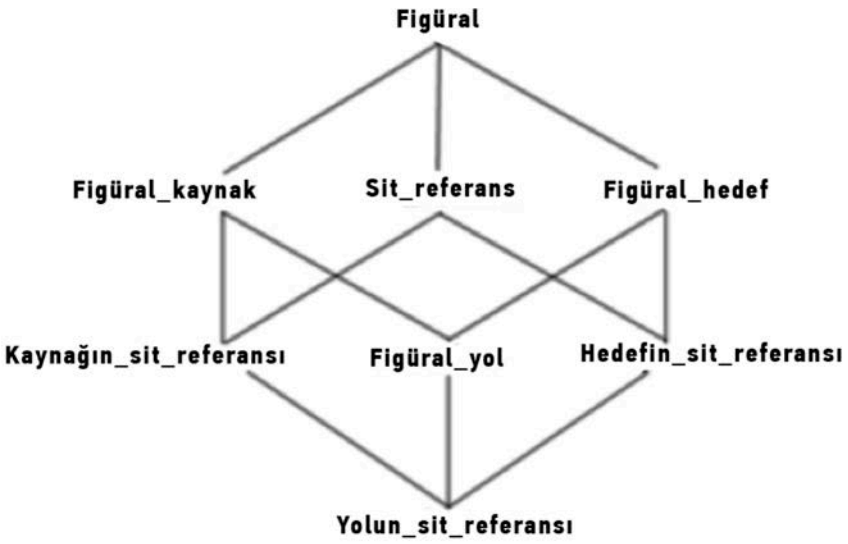
b. Oya bahçe-de bir köpeğ-i gör-üyor.

KILIÇASLAN (2006) Türkçe'nin durum imlemini dikkate alarak anlam üzerine çalışmalar yapmıştır. Onun yaptığı çalışmalar, tanımlanmış situasyonlar (described) ve kaynak (resource) situasyonları arasındaki fark üzerinedir. Yaptığı tanımlarda durum

imlemi olmayan nesnenin anlam içeriği, tanımlanmış situasyonun bir parçasıdır. Bu situasyon çeşidinde, olay nesnenin oluştuğu tümce tarafından işaret edilir. Durum imlemli nesnenin anlam içeriği ise kaynak situasyon tarafından gösterilir. Bir başka olayın tanımlanmasında bu situasyon çeşidine başvurulur. (40a) nolu tümcede bir referans ile *'bir köpeğin'* algılayan tarafından tanımlanması yapılırken, (40b) nolu tümcede ise başka bir situasyonun tanımlandığı görülür. Buradaki örneklerde görüldüğü üzere, Türkçe'de direk nesne üzerindeki durum ekinin eksikliği bağlamında nesneyi tanımlayan referans, tanımlanmış bir situasyonu işaret ederken ismin -i hali ekinin kullanıldığı nesnenin tanımlanması ise bir başka situasyonu işaret eder [16].

Anlaşılacağı üzere, situasyonel referanslar her zaman tanımlanan varlıklar için TEMA görevini üstlenir.

Yukarıda yapılan tanımlara göre, *situasyonel referans (sit_referans)* ile figüral alan tanımlanabilir. Bu alan için önerilen model Şekil 3.14 deki gibidir.



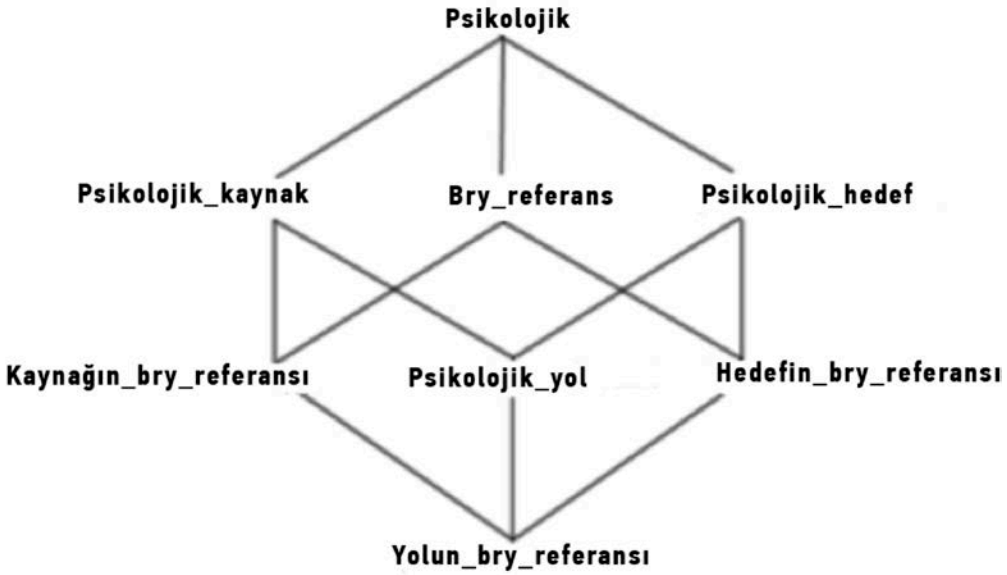
Şekil 3.14 Figüral referanslar

Şekil 3.14'de de görüldüğü üzere situasyonel referanslar varlığın konumunu tanımlayan girişlerdir.

3.2.2.3. Psikolojik Referanslar

Psikolojik alan da bir önceki alan tanımlamalarında olduğu gibi referans parametresi ile ifade edilebilir. Kişinin ruh haline göre birey bir referans olarak ele

alınır. Bu referans *bireysel referans (bry_referans)* dir. Şekil 3.15'de de psikolojik alan için tanımlanan yapı gözükmektedir.



Şekil 3.15 Psikolojik referanslar

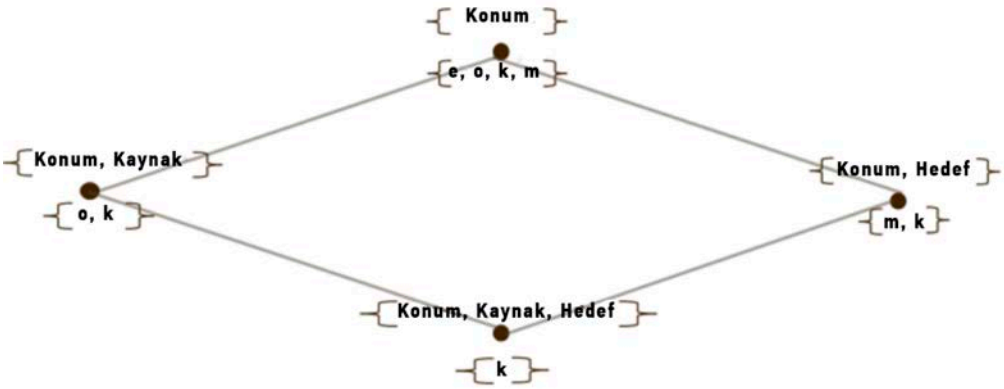
Şekil 3.15'de tanımlanan referanslar kişinin zihnindeki varlığın konumunu işaret eder.

3.2.3. Kavram Latislerinde Tematik Rol Yapıları

Yukarıda anlatılan bilgilerden yola çıkılarak tematik rollerin kullanıldığı kavram latisleri oluşturulabilir. Öncelikli olarak tematik katmanın matematiksel yapısının oluşumunu ele alalım.

(41) Çocuk evde koridordan geçerek oturma odasından mutfağa doğru yürüdü.

eyleminde *'ev'*, *'oturma odası'*, *'mutfak'*, *'koridor'* sırasıyla KONUM, KAYNAK, HEDEF ve YOL tematik rollerini alır. Bununla beraber *'oturma odası'*, *'koridor'* ve *'mutfak'* çocuğun eylemin farklı zamanlarında bulunduğu yer olarak KONUM tematik rolünü üstlenen öğelerdir. Bununla beraber *'koridor'*, eylemin başlangıç noktasında çocuğun hareket ettiği yer olarak düşünüldüğünde HEDEF tematik rolünü alır. Olayın sonuçlanmasında ise KAYNAK tematik rolünü üstlenir. Aynı zamanda bu tümcenin öznesi *'çocuk'* TEMA rolünü alır. Tematik katmana ait bu roller Şekil 3.16'daki kavram latisinde gösterilmiştir.



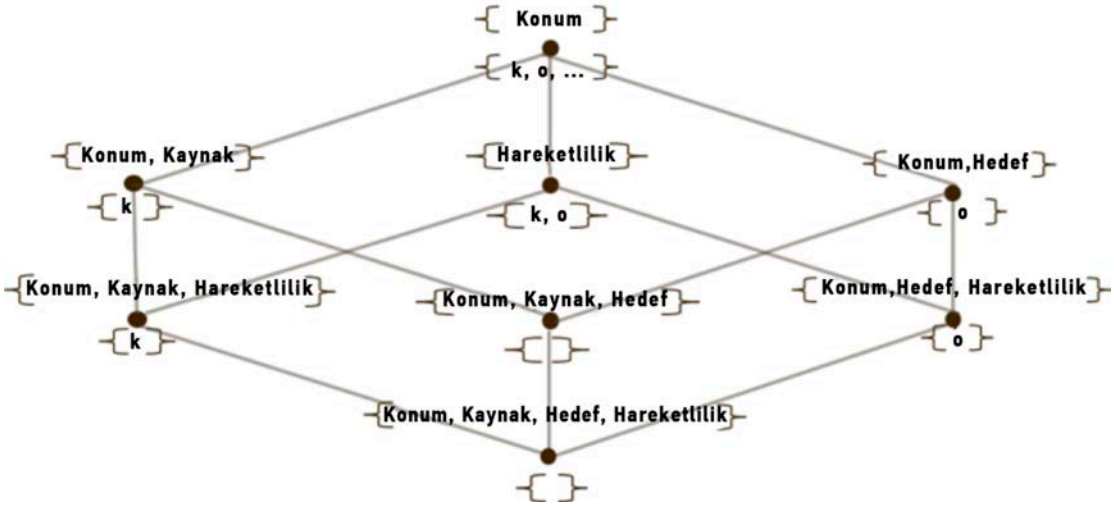
Şekil 3.16 Tematik katmana ait örnek kavram latisi

Şekil 3.16'daki bu diyagramda*'ev', 'oturma odası', 'mutfak'* ve *'koridor'* sırasıyla e, o, m, k harfleri ile sembolize edilmiştir.

Bazı durumlarda tümcenin öğeleri hem eylem katmanına ait rolleri alırken, hem de konumla ilgili rollere sahip olabilir. (42) nolu örnek tümcede bu durum rahatlıkla görülebilir.

(42) Kız oğlana top ile vurdu.

tümcesinde *'top'* TEMA tematik rolünü alır. '*kız'* ve '*oğlan'* sırasıyla KAYNAK ve HEDEF rollerini alırken aynı zamanda eylem katmanına ait ETKİLEYEN ve ETKİLENEN rollerini üstlenir. (42) nolu tümceye ait öğelerin tematik ve eylem katmanlarındaki konumları Şekil 3.17'deki gibi grafikselleştirilebilir.



Şekil 3.17 İki katman arasındaki bağlantı noktası

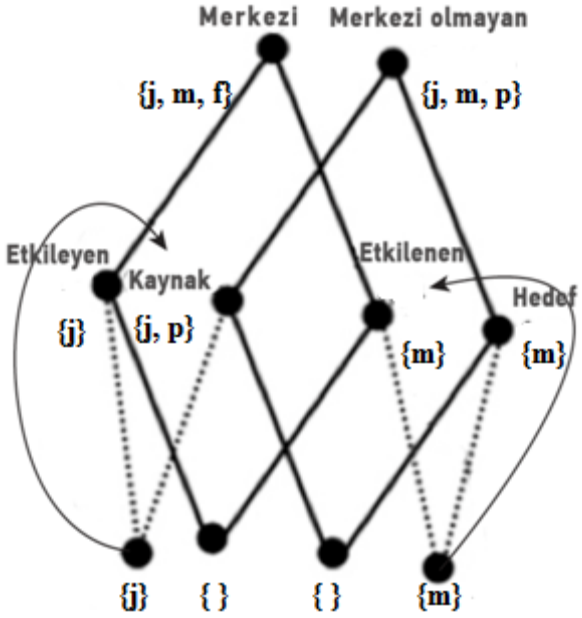
Şekil 3.17'de *k* ve *o* harfleri sırasıyla *'kız' ve 'oğlan'* sözcüklerini temsil eder [76].

Bununla birlikte tematik katmanla birlikte eylem katmanına ait roller, örnek bir tümce üzerinden Şekil 3.18'deki kavram latisleriyle şu şekilde gösterilir:

(43) a. John sent Mary Fido from Paris.

b. John Mary'e Paris'ten Fido'yu gönderdi.

tümcesinde *'John'* ETKİLEYEN rolünü, *'Mary'* hem ETKİLENEN hem de HEDEF rolünü *'Paris'* ise *'from'* ilgecinden dolayı KAYNAK rolünü üstlenir. Bu tümce için oluşturulan latis grafiği Şekil 3.18'deki gibidir.



Şekil 3.18 Kavram latislerinde tematik ve eylem katmanlarına ait tematik rollerin birlikte gösterimi

Şekil 3.18'deki latis diyagramında da görüldüğü gibi nesneler düğümlerin alt kısmında ifade edilirken, özellikler düğümün üst kısmına yerleşir. Nesneler kümesini oluşturan harfleri açık bir şekilde ifade etmek gerekirse; j (John), m (Mary), f (Fido), p (Paris) dir.

Bu bölüme ait diyagramlarda bugüne kadar doğrusal olarak ifade edilen tematik roller latislerle üç boyut haline getirilerek matematikselleştirilmiştir. Böylece yapısal latislerle sözcüklerdeki belirsizliğin çözüm yoluna gidilmiştir.

Üretilen kavram latislerinde de anlaşıldığı ve yapılan tematik rol tanımlamalarında da görüldüğü üzere yapısal pozisyonda merkezi rol TEMA dır. TEMA; hareket halinde olan veya konumlanan objeyi ifade eder.

(44) Hizmetçi koridor boyunca oturma odasından mutfağa doğru yürüdü.

tümcesinin öznesi olan *'hizmetçi'* TEMA olarak ifade edilir. Burada KAYNAK tan HEDEF e doğru bir YOL üzerinden öznenin hareketi söz konusudur.

Daha öncede ifade edildiği gibi, Jackendoff'un tematik katmanda tanımladığı konumsal roller (YOL, KAYNAK, HEDEF) tümcede sırasıyla *through*, *from*, *to* ön sözcükleri ile ifade edilir. Jackendoff'un tanımladığı eylem katmanı ise AKTÖR-

ETKİLENEN ilişkisi ile ilgilidir. Olayı gerçekleştiren AKTÖR, olaydan etkilenen ise ETKİLENEN dir. (45) nolu tümcede *'hizmetçi'* AKTÖR iken *'soğan'* ETKİLENEN olarak işaretlenir [11].

(45) Hizmetçi soğanı doğradı.

(46) ve (47) nolu örneklerde de tümce öğelerine çok boyutlu katmanla birden fazla rol atamasının yapıldığı görülür.

(46) Suzan Ferit'e vurdu.
TEMA HEDEF (Tematik katman)
AKTÖR ETKİLENEN (Eylem katmanı)

(47) Pelin topu fırlattı.
KAYNAK TEMA (Tematik katman)
AKTÖR ETKİLENEN (Eylem katmanı)

Bu anlatılanlar ışığında, kural ve ontolojinin birbiri ile ilişkilendirilerek anlama gidilmesi, tematik rollerin fraktal latislerde tanımlanması ile olmaktadır. Dowty tematik roller üzerinden belirlenen bu anlamlar için bireysel tematik roller (individual thematic roles) ifadesini kullanmıştır [4].

Buna göre *'vurmak'* eyleminin öğeleri *'vuran'* ve *'vurulan'* bireysel rollerini, *'koşmak'* eylemi ise *'koşan'* rolünü gerektirir. Eylem-öğe çiftlerinin kümesine göre öğelerin gereklilikleri başka eylem-öğe çiftleri tarafından paylaşılır. Örneğin, *'vurmak'* eylemine ait ETKİLEYEN rolü *'koşmak'* eyleminin ETKİLEYEN rolünde olduğu gibi aynı soyut gerekliliğe sahiptir [7].

Aşağıda konunun anlaşılmasına yönelik örnek tümceler detaylarıyla birlikte anlatılmıştır.

(48) Taş camı kırdı.

tümcesinde *'taş'* görülen sitüasyonda bir güç uygular ve DENEYİMLEYEN olmadığı için sadece AKTÖR tematik rolünü üstlenir. *'cam'* ise olaydan etkilenen olduğu için AKTİ rolünü oynar.

(49) Can Meryem'i öldürdü.

tümcesinde ise *'Can'* aslında NEDEN OLAN tematik rolünü oynamaktadır. Buna ek olarak DENEYİMLEYEN rolü de eklenebilir. Fakat bu yalnızca yorumda güçlü bir eğilimden kaynaklanır. NEDEN OLAN, AKTÖR gibi bir güç uygular ama AKTÖR bunu sitüasyonun içinden yaparken, NEDEN OLAN dışından yapar. Eğer ETKİLEYEN'in tanımı ((AKTÖR veya NEDEN OLAN) + DENEYİMLEYEN) olarak

alınırsa tümcedeki *'Can'* ETKİLEYEN olarak etiketlenir. *'Meryem'* ise olaydan etkilenen olduğu için ETKİLENEN olarak alınır.

(50) FBI Berkay'ı tutukladı.

tümcesinde*'FBI'* ETKİLEYEN *'Berkay'* ise hem olaydan etkilendiği hem de etkinin farkında olduğu için ETKİLENEN rolünü üstlenir.

(51) Başbakan Ankara'ya gitti.

tümcesinde ise *'başbakan'* bir eyleme neden olmadığı için ve olayın tüm evresinde yer aldığı için hem TEMA hem de DENEYİMLEYEN rolünü üstlenir. *'Ankara'* ise HEDEF olur.

(52) Can topu yokuş aşağı yuvarladı.

tümcesine bakılırsa *'Can'* (NEDEN OLAN + DENEYİMLEYEN) özelliğinden dolayı ETKİLEYEN rolünü,*'top'* ise olayın tüm evresinde yer aldığı için TEMA tematik rolünü, *'yokuş aşağı'* ise YOL tematik rolü üstlenir. Bu tümcede de görüldüğü üzere merkezdeki en temel rol TEMA'dır. Çünkü bu rol sitüasyon boyunca sürekli varlığını gösterir.

(53) Can Meryem'e bir kitap verdi.

tümcesinde *'Can'* yaptığı eylemin bilincinde ve farkında olduğu için ETKİLEYEN rolünü alırken, *'Meryem'* hem olaydan etkilenen hem de olayın farkındalığına sahip olduğu için ETKİLENEN tematik rolünü üstlenir. *'bir kitap'* ise olayın merkezinde yer aldığı için TEMA tematik rolünü alır.

Hiç şüphesiz ki yukarıdaki örneklerde de görüldüğü üzere, kavramsal latislerde konumlanan tematik roller içerisinde en çok tartışılan rol TEMA rolüdür. Bununla birlikte ETKİLEYEN rolü de önem arz eder.

1965 yılında ETKİLEYEN tematik rolünü ilk bulan kişi olan Gruber bile bu rol için tek bir tanım yapamamıştır. Gruber ETKİLEYEN tematik rolünü; bir eylemin gerçekleşmesini bile bile izin veren, isteyerek bir durumun değişmesini sağlayan neden olucu şeklinde tanımlamıştır [2].

Gruber'in takipçisi Jackendoff'un ETKİLEYEN'i bir etkiye neden olandır [1].

Daha sonraki yıllarda Jackendoff ETKİLEYEN rolü için irade, hareketlilik ve eyleme başlatıcı özelliklerini de ilave etmiştir.

Cruse'a göre kılgın olarak ifade edilen ETKİLEYEN, kendi enerjisini kullanan nesne tarafından eylemin gerçekleştirilmesini sağlayan bir öğedir. Cruse'un

ETKİLEYEN tematik rolünün analizinde belirlediği özellikler: iradeli, etkili, başlatan ve kılgın şeklindedir. Bu özellikler Halliday ve Fillmore gibi Gruber'in ETKİLEYEN kavramını içerir [33].

Tematik rol kavramında dikkate değer diğer önemli çalışmalarda geçişlilik, zaman, görünüş, ruh hali ve enerji transferi kavramları ele alınmıştır. Bu kavramların tematik ilişkiler ile yakından ilgili olduğu görülmüştür. Ve geçişliliğin derecesinin tematik hiyerarşi üzerinde kılgınlığın derecesi ile eş süreli olduğu fark edilmiştir.

DeLancey de yazılarında kılgınlık ve geçişlilik arasındaki ilişkiyi açık bir şekilde ele almıştır. Ona göre geçişlilik olayı, bir etkiye öncülük eden ve bir olaya neden olan kontrol zincirinden tek bir nedene doğru izlenebilir bir şeydir. Bu nihai neden ETKİLEYEN üzerinde iradeli bir hareket olabilir. Bu iradeli hareket doğrudan ETKİLEYEN üzerinde bir hareket doğurur ve direk olarak dış dünyayı etkiler [17].

Van Valin'e göre ise ETKİLEYEN amaçlı hareketler için tanımlanmıştır. Van Valin kılgınlığı yöneten üç faktörden bahseder: Eyleme bağlı ad öbeğinin anlam özellikleri, ad öbeğinin yapısal içeriği ve eylemle ad öbeğinin birlikte oluşturduğu dilbilgisel yapıdır. Ona göre ETKİLEYEN şu özelliklere sahip olmalıdır: konuşmacı, söylem katılımcısı, insan, akılcı, iradeli, canlı, somut varlık. Buna ek olarak Van Valin ETKİLEYEN, GÜÇ ve ARAÇ arasındaki farkları şöyle özetlemiştir: ETKİLEYEN: Canlı (tercihen insan), etkileyici, GÜÇ: Cansız, etkileyici, başlatıcı, ARAÇ: Cansız, etkileyici, başlatıcı olmayan [18].

Bu bağlamda, ETKİLEYEN için önemli kavramlar: nedensellik, irade, amaçlılık, kontrol, etkileyici ve harekettir [19].

3.3. FrameNet İle Tematik Rol Yapıları Arasındaki İlişki

Daha önce de ifade edildiği gibi, FrameNet bağlamında bir çerçeve, çeşitli katılımcıları içeren bir situasyonun şematik gösterimidir. Böylece bir çerçeve, tanımlanan varlıklara göre situasyonel referanslar gibi algılanabilir. Ayrıca çerçeveler, FrameNet veri tabanı ve tematik rol yapıları arasında ilişki noktaları gibi bir görevi de üstlenir. Bu bağlamda, bir çerçevede tanımlanan anlamsal roller situasyonel referans düğümü üzerinden kavram latislerinde ilişkili düğümlerle eşleşebilir.

FrameNet sözlüksel veri tabanı yaklaşık olarak 1200 çerçeve ve bunlara ait çerçeve anlamsal rolleri içerir. Bu rolleri tematik rol yapısında uygun düşen düğümlere

bağlama, soyut düzeyde ve bütünsel olarak aralarındaki hiyerarşik ilişkilerin görünmesine olanak tanır.

Bazı sözdizimsel kuramlarda tematik roller anlam-kural arayüzünde merkezi rol görevini görürken bizim oluşturduğumuz tematik yapılar, bu rollerin doğal dilin anlam ile kural arasında nasıl arayüz görevi üstlendiğini ortaya koymaktadır. Daha açık bir şekilde ifade etmek gerekirse, konumsal roller ilgeç öbeği gibi sözcükleştirilirken, figüral ve psikolojik alanlardaki roller özne ve direk nesne gibi kodlanırlar. Böylece, geliştirilen modelde tümcenin öğeleri tematik rol yapıları üzerinden FrameNet'in ilişkili parçaları ile bağlanır. Bağlantı aşamaları ise şu şekildedir:

- Çerçeveyi çağrıştıran sözcük birimi psikolojik olmayan eylemle eşleşirse, figüral roller yapısının *sit_referans* düğümü bu çerçeveden miras alarak, yapının ilgili düğümleri özne ve direk nesne olur.
- Eğer eylem psikolojik ise, psikolojik roller yapısının *bry_referans* düğümü bu çerçeveden miras alarak, psikolojik alanın konumsal ilgeç öbekleri konumsal rol yapısının ilgili düğümleri ile eşleşir.
- Son olarak çerçevenin ilgili çerçeve elemanlarıyla eşleşen düğümlere *Sit_referans* ve *bry_referans* bağlanır.

Her bir çerçeve elemanının bir kavram olarak düşünüldüğü bu ilişkide sözcük birimi ve onun çerçeve elemanlarının tematik roller üzerinden bulunan anlamsal rolleri ile doğru çerçeveler üretilerek anlam belirsizliği gibi problemler ortadan kalkar.

Aşağıdaki örneklerde İngilizce FrameNet'te yer alan örnek çerçeveler kullanarak o çerçeveleri çağrıştıran sözcük birimlerine ait öğelerin anlamsal rollerinin tematik rol analizleri ile nasıl belirlendiği verilmiştir. Böylece FrameNet sözlüğü ile dil arasındaki bağlantı da görülmüş olur.[6]

Örnek 1

QUARRELING (KAVGA) Çerçevesi

Biçimsel olmayan tanımı:

a. A group of ARGUERS express incompatible opinions about an ISSUE.
b. TARTIŞANLAR ifade ederler kendilerine uymayan düşücelerini BİR KONU hakkında

Tematik rol analizi:

[6] Örnek 1 ve Örnek 2 deki tümceler İngilizce FrameNet veri tabanından alınmıştır.

Tablo 3.6 QUARRELLING çerçevesinin örnek tümcelerle tematik rol analizi

(1)	Tematik roller	We AGENT bickered about semantics THEME for an hour. Biz ETKİLEYEN bir saat anlam hakkında TEMA tartıştık.
	Anlamsal roller	We ARGUERS bickered about semantics ISSUE for an hour. Biz TARTIŞANLAR bir saat anlam hakkında KONU tartıştık.
(2)	Tematik roller	Mr. and Mrs. Popple AGENT always argued angrily at least once a week. Bay ve Bayan Popple ETKİLEYEN her zaman haftada en az bir kez şiddetli olarak tartışırlar. (INI-nesnesi olmayan tümce)
	Anlamsal roller	Mr. and Mrs. Popple ARGUERS always argued angrily MANNER at least once a week. Bay ve Bayan Popple TARTIŞANLAR her zaman haftada en az bir kez şiddetli TARZ olarak tartışırlar.

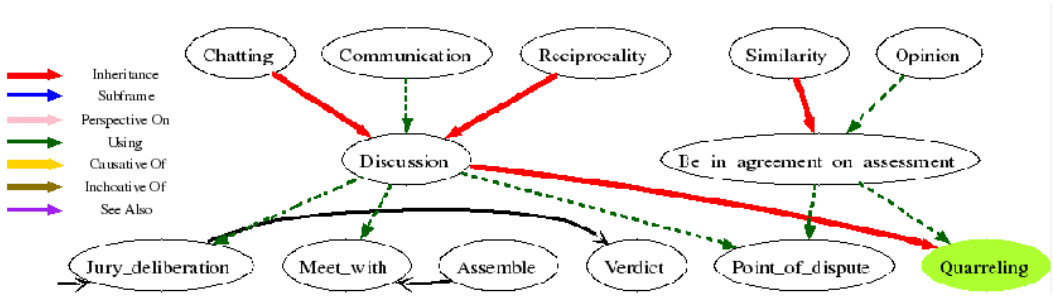
Tablo 3.6'daki (1) nolu örnekte, *'Biz (We)'* canlı bir varlık olduğu için ve farkında olarak (NEDEN OLAN + DENEYİMLEYEN) bir eylem gerçekleştirdikleri için bu sözcük öbeğin tematik rolü ETKİLEYEN ve bu rol üzerinden anlamsal rolü '*birbirleri ile tartışanlar (arguers)'*dır. Tartışma eylemi boyunca varlığını gösteren sözcük *'anlam hakkında (about semantics)'* olduğu için *'anlam (semantics)'* sözcüğü *'hakkında (about)'* ilgeci ile TEMA tematik rolü üzerinden *konu (issue)* anlamsal rolünü alır.

(2) nolu örnekte *'Bay ve Bayan Popple (Mr. and Mrs. Popple)'* farkında olarak bir eylem gerçekleştirdikleri için bu sözcük öbeğin tematik rolü ETKİLEYEN ve bu rol üzerinden anlamsal rolü '*birbirleri ile tartışanlar (arguers)'* dır. Bu rol işaretlenmesi zorunlu roldür. Nasıl tartıştıklarını gösteren gerekliliği olmayan sözcük ise *'şiddetli (angrily)'*dir. *'tarz (manner)'* anlamsal rolünü alır.

İlişki tipleri:

Çerçeve elemanlarının tematik rol analizleri ile elde edilen çıkarımlarıyla doğru çerçevelere ulaşıldığından çerçeveler arasında doğru ilişkiler kurularak doğru çerçeve sınıfları elde edilir. Böylece FrameNet'in amorf yapısı da dil ile ilişkili hale gelir.

QUARRELING çerçevesinin İngilizce FrameNet sözlüğünden alınan ilişki şeması Şekil 3.19'daki gibidir.



Şekil 3.19 Quarreling çerçevesine ait ilişkiler

Şekil 3.19'da da görüldüğü gibi QUARRELING çerçevesi DISCUSSION (TARTIŞMA) çerçevesinden, bu çerçevede RECIPROCALITY (KARŞILIKLI KONUŞMA) çerçevesinden miras alır.

Mevcut FrameNet'te yer alan REVENGE çerçevesine ait anlamsal rollerinin tematik rol analizleri ise aşağıdaki gibidir.

Örnek 2

REVENGE (İNTİKAM ALMA) Çerçevesi

Biçimsel olmayan tanımı:

a. An AVANGER performs a PUNISHMENT on a OFFENDER
b. İNTİKAM ALAN uygular CEZA bir SUÇLUYA

Tematik rol analizi:

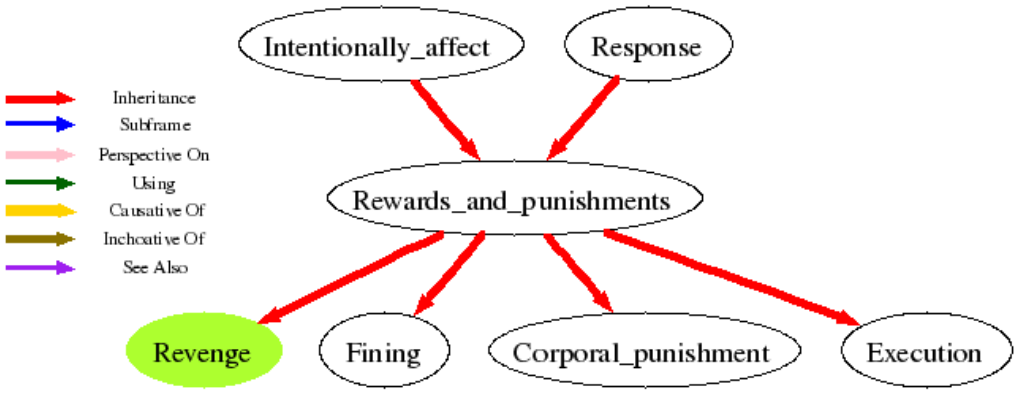
Tablo 3.7 REVENGE çerçevesinin örnek tümcelerle tematik rol analizi

Tematik roller	In "The Crossing Guard", Jack Nicholson AGENT is a policeman out to revenge his murdered son. Jack Nicholson ETKİLEYEN öldürülen oğlunun intikamını alan polis çıktı. (INI-Nesnesi olmayan tümce)
Anlamsal roller	In "The Crossing Guard", Jack Nicholson AVENGER is a policeman out to revenge his murdered son INJURED_PARTY. Jack Nicholson İNTİKAM ALANöldürülen oğlunun MAĞDUR intikamını alan polis çıktı.

Tablo 3.7'deki örnek tümcede *'Jack Nicholson'* bilinçli olarak ve isteyerek bir eylemi gerçekleştirdiğinden ETKİLEYEN tematik rolü üzerinden *'intikam alan (avenger)'* anlamsal rolünü üstlenir. İşaretlenmesi zorunlu olmayan rol olan *'öldürülen oğlan (murdered son)'* ise *'mağdur (injured_party)'* rolünü alır.

İlişki tipleri:

Şekil 3.20'de bu çerçeveye ait ilişki şekilleri görülür.



Şekil 3.20 Revenge çerçevesine ait ilişkiler

Şekil 3.20'de görülen REVENGE çerçevesi REWARDS_and_PUNISHMENTS (ÖDÜLLER_ve_CEZALAR) çerçevesinden, o da INTENTIONALLY_AFFECT (İSTEMLİ_ETKİ) ve RESPONSE (KARŞILIK) çerçevelerinden miras alır.

Örnek Türkçe bir tümcenin analizi ile Türkçe'nin FrameNet sözlüğüne bağı kurularak, çerçeve elemanlarının oluşturduğu sözcük biriminin yer aldığı çerçeveye ulaşılabilir.

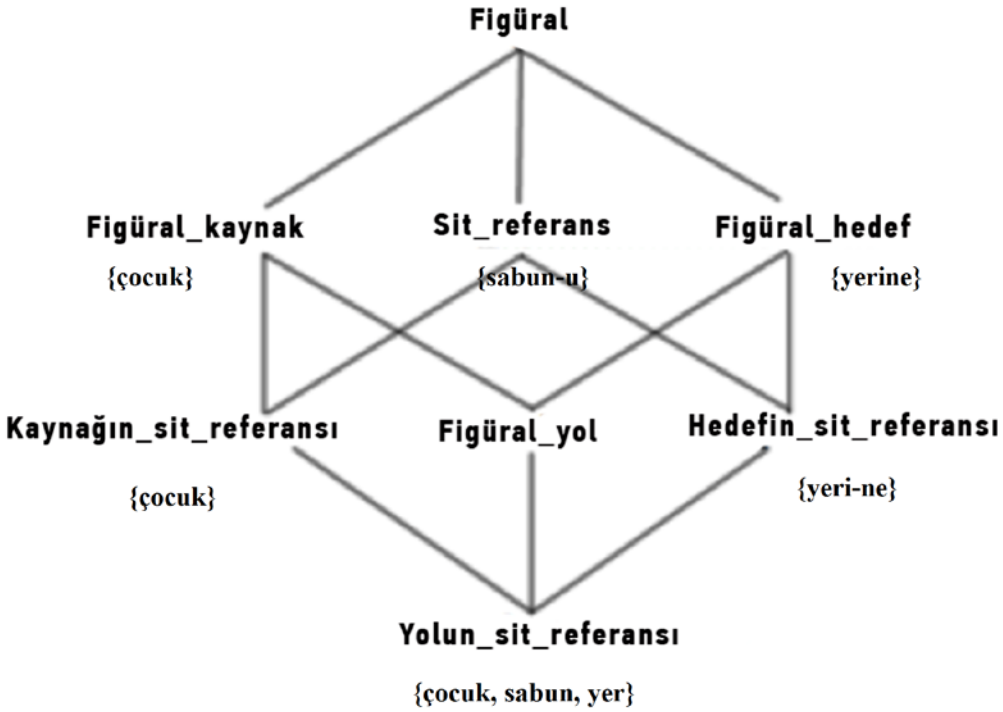
Örnek 3

Tematik rol analizi:

Tablo 3.8 Örnek Türkçe tümcenin tematik rol analizi

Tematik roller	Çocuk ETKİLEYEN/KAYNAK sabunu TEMA yerine HEDEF koyuyor.
Anlamsal roller	Çocuk KOYAN sabunu MADDE yerine YER koyuyor.

Tablo 3.8'deki *'koymak'* eylemi psikolojik bir eylem olmadığı için figüral roller yapısının *'sit_referans'* düğümü ile, mevcut FrameNet sözlüğündeki ilgili çerçeveden miras alarak yapının ilgili düğümleri özne ve direk nesne olur. *'Çocuk'* isteyerek ve bilinçli olarak olayı gerçekleştirdiğinden ETKİLEYEN tematik rolünü alır. Anlamsal rolü *'koyan'* olur. 'sabun-u' sözcüğü ise ismin –i halini tutan *situasyonel_referans* tarafından TEMA rolünü üstlenir. *'madde'* anlamsal rolünü alır. *'yeri-ne'* ise *'hedef_sit_referansı'* aracılığıyla *'sabunun'* konumunu belirlediğinden HEDEF tematik rolü üzerinden *'yer'* anlamsal rolünü alır. Şekil 3.21'de bu durum şematize edilmiştir

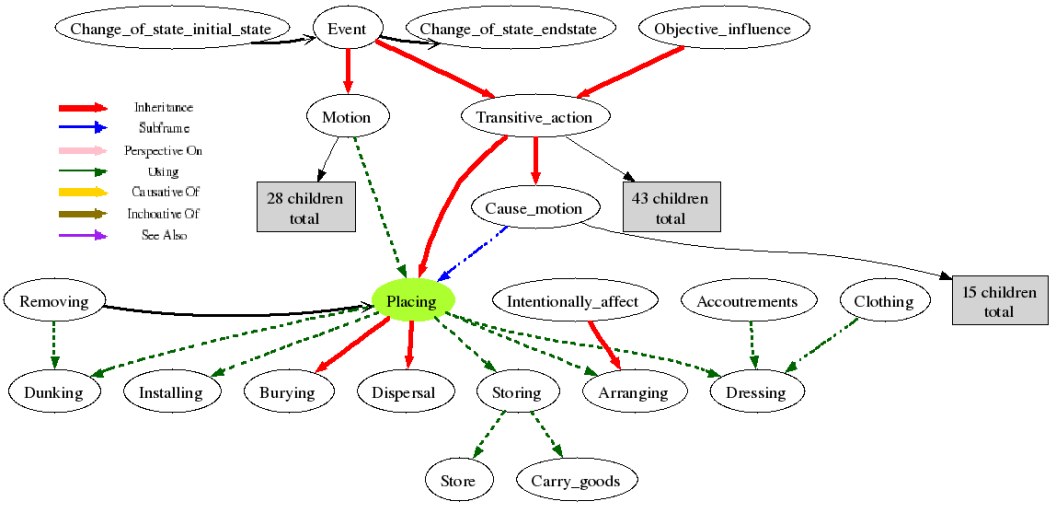


Şekil 3.21 Figüral referanslar ile Türkçe tümcenin anlamsal çözümlenmesi

Bu bilgilerden yola çıkarak 'koymak' eylemi Mevcut FrameNet sözlüğündeki PLACING (YERİNE KOYMA) çerçevesiyle eşleşir. Böylece PLACING çerçevesinin diğer çerçevelerle ilişkisine bakılarak doğru sözcük birimlerine ve bu sözcük birimlerini meydana getiren çerçeve elmanlarına erişilir.

İlişki tipleri:

Şekil 3.22'te bu ilişki ağı görülebilir.



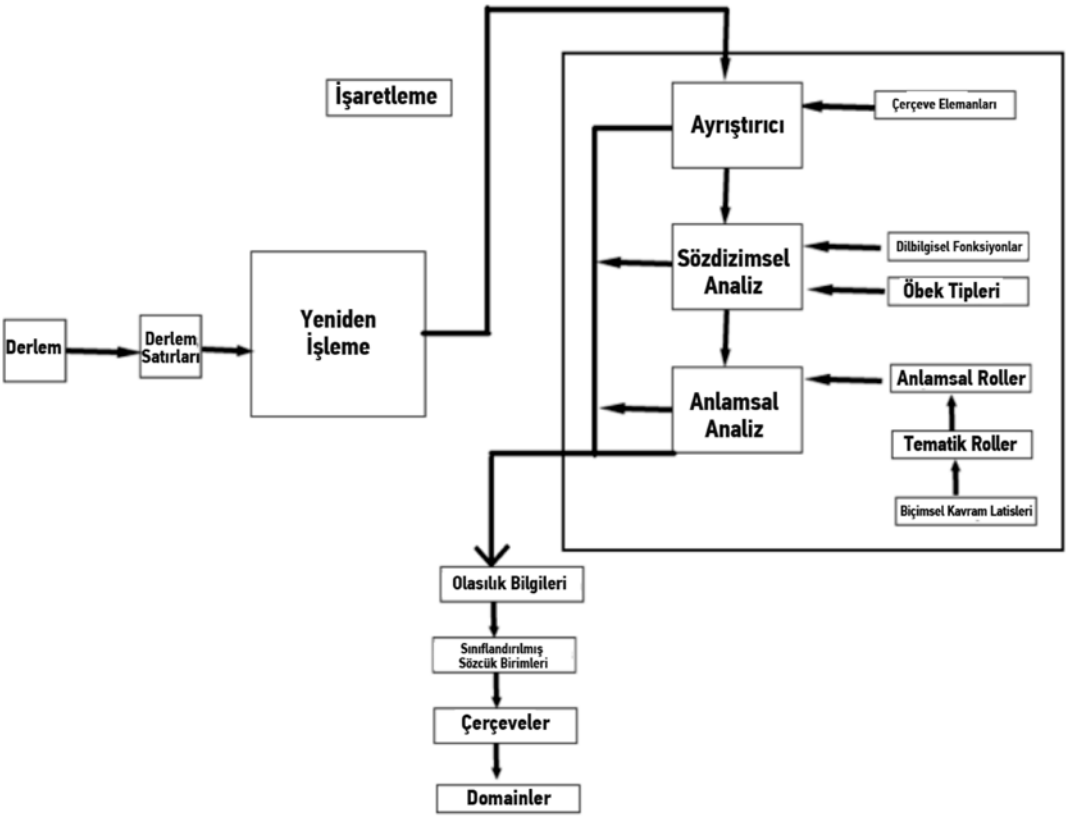
Şekil 3.22 Placing çerçevesine ait ilişkiler

Şekil 3.22'de PLACING çerçevesi TRANSITIVE_ACTION (GEÇİŞLİ_HAREKET), OBJECTIVE_INFLUENCE (NESNEL_ETKİ) ve EVENT (OLAY) çerçevelerinden miras alır.

Özetle, yapılan bu analizlerle FrameNet sözlüksel kaynağı yeniden biçimlendirilerek Türkçe'nin anlamsal çözümlenmesine katkısı da sağlanmış olur. Bu analizde, çerçeve elemanları (sözcükler) tematik roleriyle latisler üzerinde gösterilerek ilişkisel olarak sıralanırlar. Böylece özelliklerinin yanısıra kavram latisleriyle modellenerek anlamlı hale gelirler.

"Böylece FrameNet için de bir ontolojidir denilebilir" [41].

FrameNet'in kullanılan tematik rollerle birlikte genelleştirilmiş hali Şekil 3.23'de grafikselleştirilmiştir.



Şekil 3.23 FrameNet'in genelleştirilmiş hali

Şekil 3.23'de görüldüğü gibi, FrameNet sözlüksel kaynağı çerçeve ve çerçeve elemanlarının indexini içerir ki genel anlamda sözcüksel giriş olarak isimlendirilir. Aynı zamanda FrameNet, her sözcük birimi için bir işaretlemedir. Gelişmiş ontoloji tabanlı kavram latisleriyle, tümcelerin çerçeve elemanları tanımlanır. Bu latislerle, incelenen tümcenin sözcük öbeklerinin dizimsel ve biçimbilimsel özellikleri de dikkate alınarak tematik rollerine ve bu roller üzerinden anlamsal rollere ulaşılır.

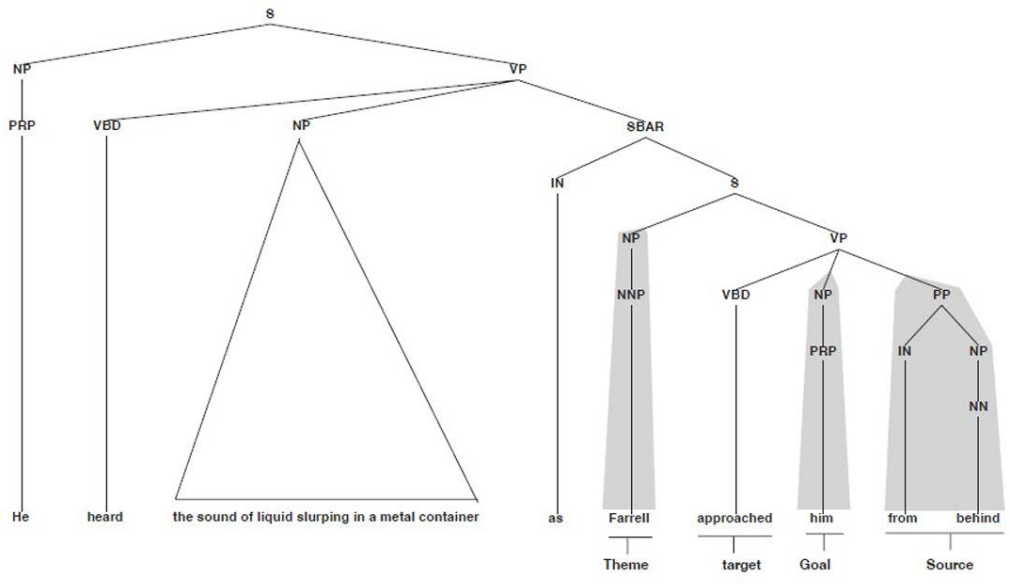
3.4. FrameNet - Tematik Rol Yapıları İlişkisine Yönelik Yapılan Çalışmalar

Bugüne kadar yapılan çalışmalarda, sözcük birimleri ve buna bağlı çerçeve elemanlarının anlamlarına ulaşma, genelde öğelerin biçimbilimsel ve dizimsel çözümlenmesi üzerinden olmuştur. Bu süreç yetersiz kaldığından son yapılan çalışmalarla öğelerin tümce içerisindeki dilbilgisel yapıya göre tematik rolleri üzerinden anlamlara ulaşılma yoluna gidilmiştir. Tematik rol atamalarında yapılan çalışmalar, öğelerin atanan tematik rollerin anlam özellikleri ile ilişkili olduğunu göstermiştir.

Özellikle; Van Valin ve LaPolla'nın, Folli ve Harley'in, Levin'in tümcelerin öğe yapılarıyla ilişkili tematik rol atamaları dikkate değerdir.

İlk FrameNet çalışması Berkeley Üniversitesi tarafından İngilizce için Charles Fillmore ve arkadaşları tarafından 1990'lı yıllarda oluşturulmuştur. Bu çalışmada 100.000.000 sözcük ve 50,000 tümceden oluşan BNC (British National Corpus) kullanılmıştır. Bu derlemden el ile işaretlenen 50,000 tümcenin sözcüklerinin anlamsal ve dizimsel özelliklerinden bir veri tabanı elde edilmiştir. Bu bilgilerden oluşturulan bir diğer veri tabanı ise 80 tane çerçeveyi içermektedir. İngilizce FrameNet çalışmasında öğelerin dizimsel özellikleri üzerinden belirlenen anlamsal rolleri ile sözcüklerin sınıflandırılmasına gidilmiştir. Bu projede roller anlamsal çerçeveler için tanımlanmaktadır. Tematik roller ise kısmi olarak kullanılmaktadır. Günümüzde ise bu sözlük içerdiği çerçeve sayısı bazında genişletilmeye çalışılmaktadır.

Bir başka çalışma, Daniel Gildea ve Daniel Jurafsky tarafından FrameNet projesi üzerine inşa edilmiştir. Araştırmacılar kullandıkları 50.000 tümcenin anlamsal rollerini el ile işaretleyerek istatiksel sınıflandırma yoluna gitmişlerdir. Anlamsal rollerin belirlendiği görsel, Şekil 3.24'de görülür [6].



Şekil 3.24 Dizimsel özellikleri ve tematik rol bilgileri üzerinden anlamsal rollerin belirlendiği örnek tümce

Şekil 3.24'de görüldüğü üzere tümce öğelerinin anlamsal rollerine dizimsel özellikleri ile ilişkili evrensel ve daha küçük rol kümesini ifade eden geleneksel soyut tematik roller üzerinden ulaşılmıştır.

Bu çalışmada;

- Her bir tümce kural ağacı ile parçalara ayrıştırılmıştır.
- Bu parçaların dizimsel özellikleri belirlenerek dilbilgisel fonksiyonlarına ulaşılmıştır.
- Elde edilen anlamsal rol bilgileri (tematik ve anlamsal roller) ve sözcük birimlerinin dağılım frekans bilgilerinin kombinasyonu ile sözcüklerin sınıflandırılması gerçekleştirilmiştir.

Elektronik ortamda FrameNet gibi çalışmalarla kullanılmak üzere İngilizce için yapılan en geniş kapsamlı fiil sözlüğü ise Levin'in geliştirdiği VerbNet'tir. VerbNet, eylemlerin hiyerarşik bir organizasyon içerisinde dizimsel ve anlamsal açıdan sınıflandırıldığı bir teknik sözlüktür. VerbNet tekniğinde eylemlerin sınıflandırılmaları yapılır. Sınıflar ve alt-sınıflar, eylemler arasındaki biçimbilimsel / dizimsel ve anlamsal ilişkilerin sağlanması ile gerçekleştirilir. Her bir eylem sınıfı, öğelerin dizimsel özellikleri ile birlikte tematik rolleri ve seçimsel sınırlılıkları tarafından belirlenir. Özellikle seçimsel sınırlılıklar öğelerin tematik rollerini belirlemede önemli bir yer tutar. Öğenin canlı olup olmaması (animate, inanimate), soyut ve somut olması (abstract, concrete) gibi durumlar seçimsel sınırlılıkları ifade eder [50].

VerbNet sözlüğünden alınan örnek eylem sınıfı ve üyeleri Şekil 3.25'de incelenebilir [27].[7]

[7] VerbNet sözlüğündeki diğer eylem sınıfları http://verbs.colorado.edu/~mpalmer/projects/verbnet.html adresinden ulaşılabilir.

SINIF: mine-10.9 [Üyeler:2, Çerçeveler:2]

ÜYE: mine (maden)

ÜYE: quarry (taş ocağı)

ANLAMSAL ROL: Agent [+animate | +organization]

ANLAMSAL ROL: Theme [+concrete]

ANLAMSAL ROL: Source [+location]

ÇERÇEVE: [Basic Transitive]

ÖRNEK: The men mined the gold. / Adamlar altın çıkardılar.

SÖZ DİZİM: Agent v Theme

ANLAMSAL YÜKLEM: cause (Agent,E)

ANLAMSAL YÜKLEM: location (start (E), Theme, ?Source)

ANLAMSAL YÜKLEM: not(location (end (E), Theme, ?Source))

ÇERÇEVE: [PP // source-PP]

ÖRNEK: The men mined the gold from the abandoned mine.
Adamlar terkedilmiş madenden altın çıkardılar.

SÖZ DİZİM: Agent v Theme {{+src}} Source

ANLAMSAL YÜKLEM: cause (Agent,E)

ANLAMSAL YÜKLEM: location (start (E), Theme, Source)

ANLAMSAL YÜKLEM: not (location (end (E), Theme, Source))

Şekil 3.25 VerbNet sözlüğünden alınan örnek eylem sınıfı

FrameNet ile yapılan bir diğer çalışma ise iki dilli eşleştirme yapısıdır. Bu yapı Hans Boas tarafından İngilizce ve Almanca sözcük birimleri üzerinden başarıyla gerçekleştirilmiştir. Hans Boas İngilizce sözcük birimlerinin İngilizce FrameNet veri tabanındaki dizimsel ve anlamsal özelliklerine bağlı olarak Almanca karşılıklarının bulunmasına yönelik çalışmalar yapmıştır. Özellikle İngilizce FrameNet'teki COMMUNICATION-CONVERSATION (İLETİŞİM-KONUŞMA) çerçevesi içindeki *'argue (tartışmak)'* eylemine yönelik dizimsel ve anlamsal analizi ile eşleştirme çalışması dikkate değerdir.[8] *'argue'* eylemine ait İngilizce derlemden seçilen tümcelere

[8] İngilizce FrameNet veri tabanı sürekli olarak güncellendiğinden COMMUNICATION_CONVERSATION çerçeve ismi QUARRELLING olarak değişmiştir.

ait sözcüksel girişler Tablo 3.9'da, anlamsal olarak gerekli işaretlemeler Tablo 3.10'da gösterilmiştir.

Tablo 3.9 *'argue'* eyleminin sözcüksel girişi

	Interlocutors (konuşmacılar)	TARGET (sözcük birimi)	Topic (konu)
1	NP.Ext (isim öbeği.harici)	argue.verb (tartışmak.eylem)	INI (nesnesi boş)
2	NP.Ext (isim öbeği.harici)	argue.verb (tartışmak.eylem)	PP_over.Comp (tamlayıcı üzerine)
3	NP.Ext (isim öbeği.harici)	argue.verb (tartışmak.eylem)	PP_about.Comp (tamlayıcı hakkında)

Tablo 3.10 Anlamsal olarak işaretlenmiş derlem tümceleri

1	[$_{\text{interlocutors}}$ Mr and Mrs Popple] argued once a week [$_{\text{topic}}$ INI]. [$_{\text{konuşmacılar}}$ Bay ve Bayan Popple] haftadabir kez [$_{\text{konu}}$ INI] tartıştı.
2	[$_{\text{interlocutors}}$ Auction houses and buyers] argue [$_{\text{topic}}$ over compensation]. [$_{\text{konuşmacılar}}$ Müzayedeci ve alıcılar] [$_{\text{konu}}$ bedel üzerine] tartışır.
3	[$_{\text{interlocutors}}$ They] argued [$_{\text{topic}}$ about it]. [$_{\text{konuşmacılar}}$ Onlar] [$_{\text{konu}}$ onun hakkında] tartıştı.

'argue' eyleminin sözcüksel girişi ve anlamsal analizinin Almanca karşılıkları ise Tablo 3.11 ve Tablo 3.12'da gösterilmiştir.

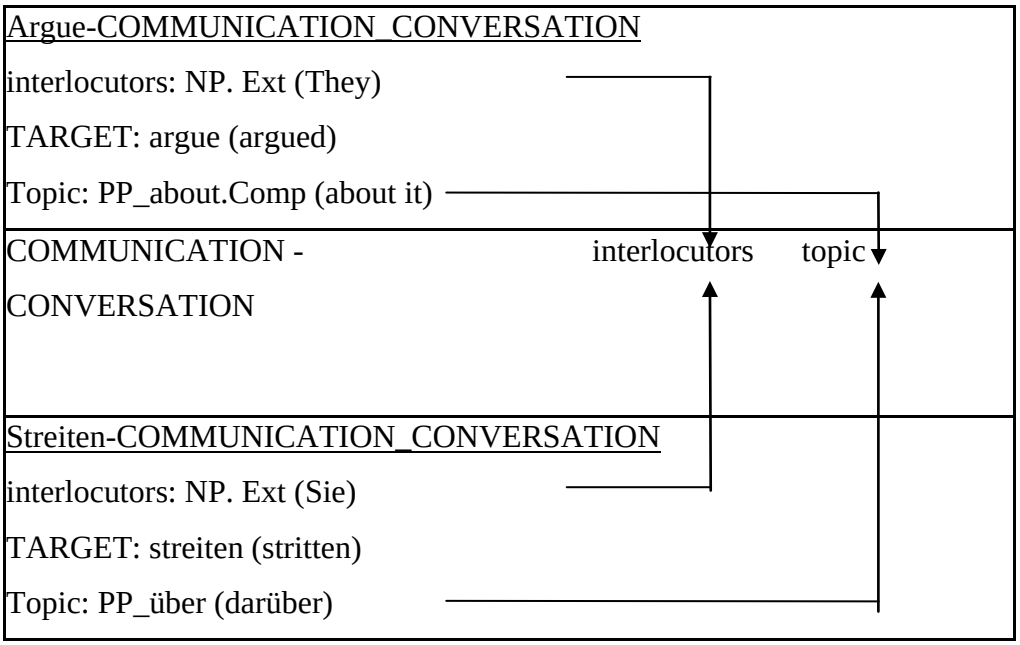
Tablo 3.11 Almanca *'streiten (tartışmak)'* eyleminin sözcüksel girişi

	Interlocutors	TARGET	topic
1	NP.Ext	streiten	INI
2	NP.Ext	streiten	PP_um.Comp (tamlayıcı üzerinde)
3	NP.Ext	streiten	PP_über.Comp (tamlayıcı hakkında)

Tablo 3.12 Anlamsal olarak işaretlenmiş Almanca'da karşılığı olan tümceler

1	[$_{\text{interlocutors}}$ Herr und Frau Popple] stritten ein mal pro Woche [$_{\text{topic}}$ INI]. [$_{\text{konuşmacılar}}$ Bay ve Bayan Popple] haftada bir kez [$_{\text{konu}}$ INI] tartıştı.
2	[$_{\text{interlocutors}}$ Auktionhouser und Kaufer] streiten [$_{\text{topic}}$ um die Entschadigung]. [$_{\text{konuşmacılar}}$ Müzayedeci ve alıcılar] [$_{\text{konu}}$ bedel üzerine] tartışır.
3	[$_{\text{interlocutors}}$ Sie] stritten [$_{\text{topic}}$ daruber]. [$_{\text{konuşmacılar}}$ Onlar] [$_{\text{konu}}$ onun hakkında] tartıştı.

İngilizce ve Almanca sözcük parçalarının ilişkisi ise Şekil 3.26'da gösterilmiştir [47].



Şekil 3.26 İngilizce ve Almanca sözcük girişlerin eşleştirilmesi

Dilin anlamsal çözümlenmesine yönelik olarak tematik rollerle ilgili olarak yapılan bu çalışmalar oldukça yetersiz kalmaktadır. Bu nedenle matematikleştirilmiş bir yapının doğal dile entegre edilmesi gerekir. Bu da kavram latisleri ile gerçekleşir. Kavram latisleri ile Türkçe'nin anlamsal çözümlenmesine yönelik bilgisayarlı uygulama çalışması ise bir sonraki bölümde ayrıntılı bir şekilde ele alınmıştır.

BÖLÜM 4

TÜRKÇE'NİN ANLAMSAL ÇÖZÜMLENMESİNİN BİLGİSAYARLI GERÇEKLEŞTİRİMİ

Bu bölümün amacı, FrameNet üzerinden bir gerçeklik modelinin bilgisayarlı gerçekleştirimini tanıtmaktır. Uygulama, biçimsel kavram analizi çerçevesinde derlem tabanlı işaretleme yöntemiyle geliştirilmiştir. Kullanılan derlem 1033 tümceden oluşmaktadır. Bu derlem bire bir Türkçe karşılığı olduğu için, kutsal kitap İncil'in Yuhanna'ya giriş kısmından alınmıştır.[9] Özel uygulama alanı olarak Türkçe'nin anlamsal çözümlenmesi hedeflenmiştir.

İşaretleme ise, verilen tümcelerin parçalara ayrılarak, her bir parçanın veya öğenin anlamsal ve biçimbilimsel / dizimsel özelliklerinin örneklenmesidir.

Kendi derlemimizden seçilen (54) nolu tümceye ait işaretleme Tablo 4.1'de görülmektedir.

(54) Şölen başkan-ı, şarab-a dönüş-müş su-yu tat-tı.

[9] Derleme http://incil.info adresinden ulaşılabilir.

Tablo 4.1 Örnek işaretleme

[Theme tematik rolü]	=>	‘suyu’
[Agent tematik rolü]	=>	‘Şölen başkanı’
[Goal tematik rolü]	=>	‘şaraba’
[T]	=>	‘tı’
[VP]	=>	‘tat’
[Case]	=>	‘yu’
[NP]	=>	‘su’
[ASP]	=>	‘müş’
[VP]	=>	‘dönüş’
[Case]	=>	‘a’
[NP]	=>	‘şarab’
[Case]	=>	‘ı’
[NP]	=>	‘başkan’
[NP]	=>	‘Şölen’

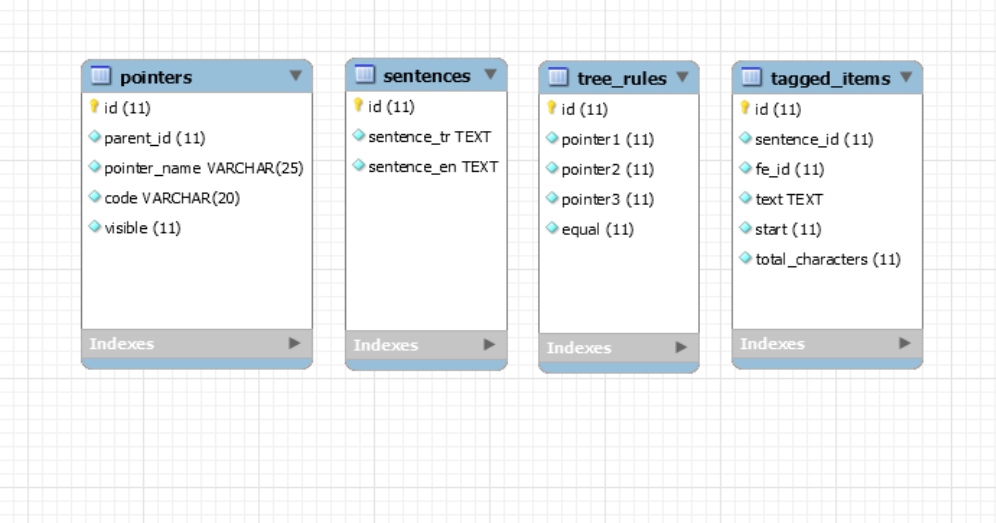
Tablo 4.1‘de görüldüğü üzere tümcenin öğelerinin tematik rol ve biçimbilimsel / dizimsel işaretlemeleri yapılmıştır. Bu tümcenin biçimbilimsel / dizimsel özellikleri ise şunlardır:

- İsim sözcük öbeği –NP
- Eylem sözcük öbeği –VP
- Hal eki- Case
- Zaman eki-T
- Görünüş eki-ASP

Bu tip işaretlemeler için bilgisayar ortamında hazırlanmış olan işaretleme arayüzü (annotation tool) PHP (Hypertext Preprocessor) dili ve MySQL’de (My Sequential Query Language) oluşturulan veritabanı kullanılarak gerçekleştirilmiştir.[10] Bu arayüz (tool) web ortamında çalışan bir sayfa konumundadır.[11]

[10] yelda_tool.sql veri tabanına http://www.yelda1367.com:2082/ adresinden ulaşılabilir.
[11] Arayüze http://www.yelda1367.com/member/login adresi üzerinden erişilebilir.

Uygulama çalışmamızda yelda_tool.sql isimli veritabanı dosyası mevcuttur. Bu veri tabanı dosyası içinde Şekil 4.1'de görüldüğü üzere dört tane tablo mevcuttur.



Şekil 4.1 yelda_tool.sql veritabanına ait tablolar

Şekil 4.1'de görülen tablolardan;

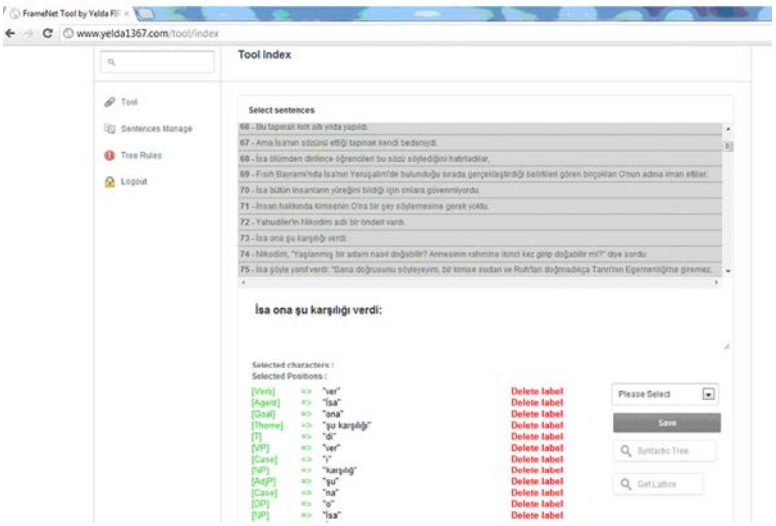
Pointer tablosu: Ağaç yapılarının çıkarılmasında kullanılan tümce öğelerinin detaylarını tutan alanlardan oluşan tablodur.

Sentences tablosu: Türkçe ve İngilizce tümceleri tutan alanları kapsayan tablodur.

Tree_rules tablosu: Pointerların veya tümce öğelerinin meydana getirdiği ağaç yapılarını oluşturan kuralları tutan tablodur.

Tagged_items tablosu: İşaretlenen tümcenin öğelerine ait alanları tutan tablodur.

Şekil 4.2'de gösterilen arayüz ile elimizdeki derlemeye ait Türkçe tümceler parçalara ayrılarak tematik rolleri işaretlenmiş, biçimbilimsel / dizimsel analizleri yapılmıştır.



Şekil 4.2 Biçimbilimsel / dizimsel - tematik rol işaretleme ara yüzü

Şekil 4.2'ye ait görüntünün javaScript kodları:

```
<script language="JavaScript" type="text/javascript" src="<?php echo base_url();
?>js/Selection.js"></script>
<script>
/*
Move JS fonksiyonu seçilen tümceyi sentence_field id li textbox'a taşır. Tümceye
ait id numarasını ise sentence_id id numaralı fielda taşır. check_selected_items()
fonksiyonunu kendi içerisinde tetikler. Seçili alanı boşaltır. Move fonksiyonu
parametre olarak tümcenin id sini alır ona göre işlem yapar.
*/
function move(id)
{
        var sentence = $('#s'+id).text();
        $('#sentence').val(sentence);
        $('#sentence_field').val(sentence);
        $('#sentence_id').val(id);
        check_selected_item();
        $('#selected-characters').empty();
        $('#tree').attr('href','<?php echo base_url(); ?>tool/tree/'+id);
```

```
    $('#lattice').attr('href','<?php echo base_url(); ?>lattice/get/'+id);
}
```

/*

open_selectbox fonksiyonu parametre olarak işaretlenen öğenin türünü alır. (FE, LU)

Ona göre ilgili kategorinin alt selectboxını açar.

***/**

```
function open_selectbox( value )
{
        var selection = $('#selected').val();
                if(value == 'lu')
        {
                $('#selected-characters').html('<input type="text" name="lu" value="' +
selection + '" />');
        }
                if(value == 'fe')
        {
                $('#fe').fadeIn('normal');
                $('#selected-characters').html(selection);
        }
        Else
        {
                $('#fe').fadeOut('normal');
        }
}
```

/*

Save fonksiyonu Tool controller'ının Save methoduna, seçili olan kelime veya kelime grubunun seçilen kategorilere göre kaydedilmesi için bir ajax isteği gönderir. #tool id li form içerisindeki tüm form elementlerini serialize ederek POST eder. Ajax isteği göndermeden önce istek tekrar tekrar gönderilmesin diye submit butonunun olduğu container div elementinin içeriğinde bir loading imajı gösterir. Ajax isteğinin yanıtı geldiğinde submit butonu tekrar yerine koyar.

#ajax_message id li div içerisine Ajax isteğinden gelen mesajı yazar. Ardından check_selected_items() javascript fonksiyonunu tetikler.

***/**

```
function save()
{
                $.ajax({
        type: "POST",
        url: "<?php echo base_url('tool/save'); ?>",
data: $("#tool").serialize(),
                        beforeSend:function(msg){ $("#submitbutton").html('<img
src="<?php echo base_url('img/loading.gif'); ?>" />'); },
        success: function(msg)
{
        $("#ajax_message").html(msg);
        $("#submitbutton").html('<input type="button" style="width:160px;margin-
top:10px;" onclick="save();" name="button" id="button" value="Save" class="st-
button"/>');
        check_selected_item();
                        }
          });
}
```

/*

check_selected_items fonksiyonu tümcenin id numarasını #sentence_id isimli gizli bir elementten alır. Tool controllerının check_selected_item methoduna, seçilen kelime veya kelime öbeğinin daha önce işaretlenip işaretlemediğini kontrol etmek için bir ajax isteği yapar.

***/**

```
function check_selected_item()
{
        var sentence_id = $('#sentence_id').val();
        $.ajax({
        type: "POST",
```

```
        url: "<?php echo base_url('tool/check_selected_item'); ?>",
                        data: $("#tool").serialize(),
                        beforeSend:function(msg){ $("#tagged_item").html('<img
src="<?php echo base_url('img/loading_small.gif'); ?>" />'); },
                        success: function(msg){ $('#tagged_item').html(msg); }
        });
}

/*
```

delete_label fonksiyonu parametre olarak gönderilen label id numarasına göre işaretlenen bir kelime veya kelime grubunu silmek için Tool controlleri'nın delete_label methoduna bir ajax isteği gönderir.

```
*/
function delete_label(label_id)
{
        $.ajax({
        type: "GET",
        url: "<?php echo base_url('tool/delete_label'); ?>/" + label_id,
                        success: function(msg){ if(msg == 'ok'){
$('#tagged_item').empty(); check_selected_item(); } }
        });
}
/*
```

getSelected fonksiyonu seçilen kelime veya kelime grubunu döndürür.

```
*/
function getSelected() {
  if(window.getSelection) { return window.getSelection(); }
  else if(document.getSelection) { return document.getSelection(); }
else {
    var selection = document.selection&& document.selection.createRange();
    if(selection.text) { return selection.text; }
    return false;
```

```
}
  return false;
}
$(document).ready(function() {
/*
```

#sentence id numaralı tümcenin bulunduğu textboxda mouse up eventi tetiklendiğinde seçili olan kelime veya kelime grubu alınır. Ve tümcenin kaçıncı karakterinden başladığı kaç karakter içerdiği hesaplanır. Gerekli divlere yazılır.

```
*/
  $('#sentence').mouseup(function() {
var selection = getSelected();
          var sentence  = $('#sentence').val();
          var counter   = sentence.length;
    if(selection && (selection = new String(selection).replace(/^\s+|\s+$/g,''))) {
                    var selected = selection.substring(0,counter);
        $('#selected-characters').html(selected);
                    $('#selected').val(selected);
                    evalOnce();
}
  });
});
var timer;
var buffer = "";
/*
```

Gönderilen element checklenmiş ise evalStart fonksiyonunu checklenmemişse evalStop fonksiyonunu çalıştırır.

```
*/
function evalSwitchClicked(e) {
    if (e.checked) {
        evalStart();
} else {
        evalStop();
```

```
}
}
/*
```
evalStart fonksiyonu eval_time idli elementten time değeri alır ve setTimeout ile alınan değer kadar sonra timerHandler fonksiyonunu tetikler.

```
*/
function evalStart() {
   var o = document.getElementById("eval_time");
   timer = setTimeout(timerHandler, o.value);
}
/*
```
evalStop fonksiyonu evalStart fonksiyonunda tanımlanan timer ı iptal eder.

```
*/
function evalStop() {
   clearTimeout(timer);
}
/*
```
timerHandler fonksiyonu çalıştığında timer ı iptal eder. eval_switch id li elementin checked olup olmadığına bakar. Checked durumundaysa evalOnce fonksiyonunu çalıştırır. Artından evalStart ile yeni bir timer tanımlar.

```
*/
function timerHandler() {
   clearTimeout(timer);
   var sw = document.getElementById("eval_switch");
   if (sw.checked) {
      evalOnce();
      evalStart();
}
}
/*
```
evalOnce fonksiyonu ilk önce sentence id li textboxın element objesini parametre olarak Selection nesnesine verir ve yeni bir örneğini oluşturur. s adında bir

değişkene kendi içerisindeki create fonksiyonunu verir. Artık seçili kelime yada kelime grubunun tümcede kaçıncı karakterden başlayıp kaç karakterde bittiğini s.start ve s.end alarak alabilecek haline getirir.

***/**

```
function evalOnce() {
   try {
      var selection = new Selection(document.getElementById("sentence"));
      var s = selection.create();
      var result = "[" + s.start + ":" + s.end + "]";
      buffer += result;
      flush();
} catch (ex) {
      buffer = ex;
      flush();
}
}
```

/*

clear fonksiyonu selected-positions id li seçili öğenin kaçınca karakterde başlayıp kaçıncı karakterde bittiğini bilgi amaçlı gösteren elementin içini temizler. Positions idli elementin değerini de boşaltır.

***/**

```
function clear()  {
   var out = document.getElementById("selected-positions");
out.innerHTML = "";
   $('#positions').val('');
}
```

/*

print fonksiyonu root scope'da buffer değişkenine parametre olarak gelen str değişkenin değerini değişkeni bozmadan ekleme yapar.

***/**

```
function print(str)  {
   buffer += str + "\n";
```

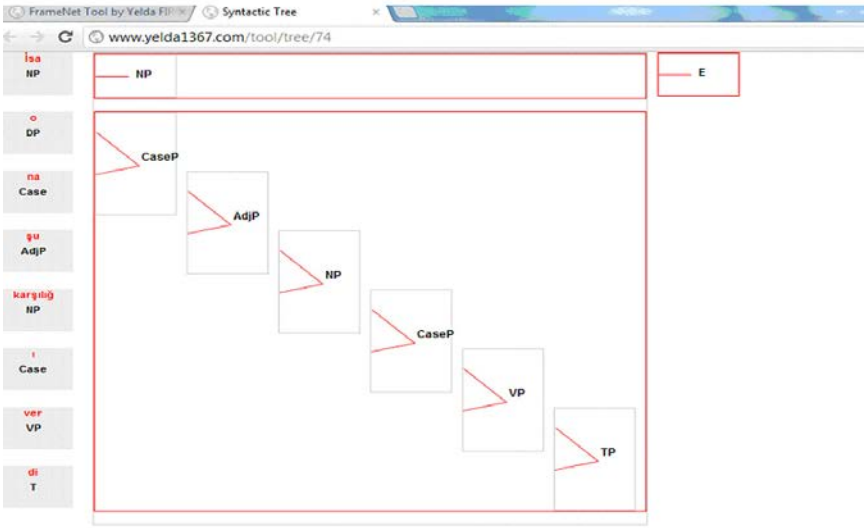
```
}
/*
flush fonksiyonu selected-positions idli elementin içerisine root scope daki buffer değişkeninin içerini doldurur. Bir de positions idli elementin değerine buffer değişkeninin değerini yazar. Ve buffer değişkenini tamamen boşaltır.
*/
function flush()  {
   var out = document.getElementById("selected-positions");
out.innerHTML = buffer;
   $('#positions').val(buffer);
   buffer = "";
}
</script>
```

Bu işaretlemelerden elde edilen bilgilerle oluşturulan veri tabanı, tümcelere ait öğelerin tanımını içerir ve geleneksel sözlüklerden farklı olarak çok miktarda bilgi sağlar [46].

Bizim oluşturduğumuz veri tabanı da bir başka deyişle yukarıda tanımlanan tablolar tümcedeki her bir öğenin biçimbilimsel / dizimsel ve anlamsal gösterimlerini tanımlayan karakteristik özellikleri kapsar.

Kodlamada yer alan model ve kontroller sınıflarından model sınıfı veri tabanına bağlanıp bize istediğimizi getiren bir sınıftır. Kontroller sınıfı ise istek yaptığımız, asıl işlemleri yapan sınıftır. Bir başka deyişle tümcenin id numarası kontroller sınıfına istek göndererek hesaplama süreci başlar. Böylece yapılan kodlamada model, istenen şartlara göre veriyi veri tabanından çekerek kontroller'a atar ve kontroller gereken gruplamaları yaparak grafiği obje haline getirip, view dosyasına göndererek ekrana html görüntüsünü verir.

Bu anlamda Şekil 4.3'de derlemden seçilen örnek bir tümcenin kural ağacının otomatik olarak çıkarılmış görüntüsü verilmiştir. Şekil 4.4'de ise seçilen örnek tümceye ait öğelerin tematik rollerine ulaşmak için otomatik olarak çıkarılmış latis diyagramı gösterilmiştir.



Şekil 4.3 İşaretlenmiş örnek tümcenin otomatik olarak çıkarılmış kural ağacı

Şekil 4.3'de gösterilen işaretlenen tümcelerin öğelerini eşleştiren ve gerekli kontrolleri yaparak grafiğe döken metodun kodları:

/*

tree methodu parametre olarak id alır. Bu method işaretlenmiş bir tümcenin ağacını ekrana çizer. ID olarak bir sentence id numarası gönderilmelidir. Gönderilen id ye göre Tool Modelinden ilgili tümcenin detayı istenir.

Subject'in başlangıç ve bitişi Tool Modelindeki exists_subject methodu çağrılarak istenir. Tümcede Conj ve C var mı diye yine aynı şekilde Tool modelinden istenir.

Tümcedeki kelime sayısı kadar dönecek bir for döngüsü başlatılır. Her dönüşte o kelimenin ve bir öncesindeki kelimeyle türünün ne olduğuna göre bir ağaç dalı oluşturulur ve tree arrayine kaydedilir. Bu ağacın haritasını taşır.

Her dönen kelime veya kelime grubunun bağlı olduğu bir kural olması gerekir. Ağaç haritası çıkarılıp ekrana çizdirilmeye başladığında sürekli Modeldeki check_rule methoduna sorulur. Eğer ağacı çizilecek olan tümcede bir eşleşmezlik

veya kural eksikliği var ise ağaç çıkarılmaz ve "Kural eksikliği vardır" şeklinde hata bastırılır.

Aynı zamanda bir sonraki kelimenin kuralının bir önceki ağacı çıkarılan kuralla eşleşmezliği varsa da ağaç çıkarılmasında hata oluşur.

***/**

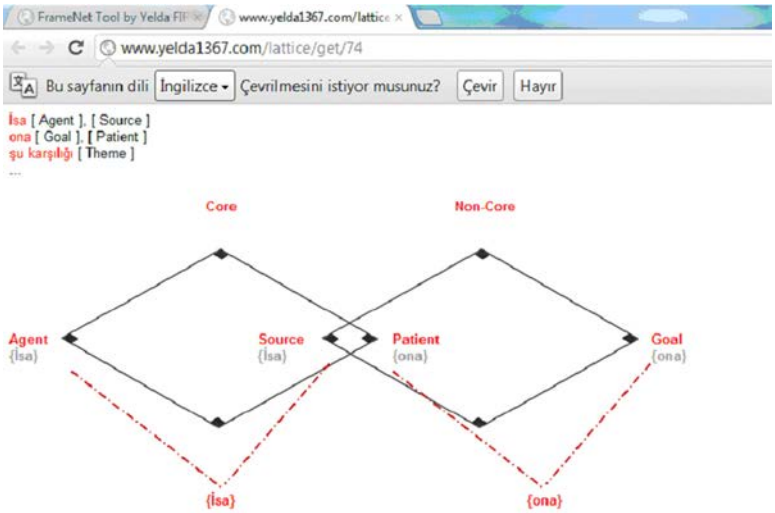
```
function tree( $id )
{
// böyle bir sentence var mı kontrol edilir. varsa işleme devam ettirilir.
echo '<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> ';
$words = $this->tool_model->get($id);
$need   = null;
$tree   = array();
 // Tümcede temel özelliklerin olup olmadığını bulan fonksiyonlar
$subject_end    = $this->tool_model->exist_subject($id,'end');
$subject_start  = $this->tool_model->exist_subject($id,'start');
$conj    = $this->tool_model->exist_conj($id);
$c       = $this->tool_model->exist_c($id);
$counter = 0;
$conj_counter = 0;
for($i=0;$i<=count($words)-1;$i++)
{
          if( $conj > 0 || $c > 0)
{
              if( $words[$i]->fe_id == 75 ||  $words[$i]->fe_id == 39)
{
                  $type = 0;
                  $tree[] = $type.'|'. $words[$i]->id; $i++;
                  $counter = 0;
                  $need = null;
}
}
```

```
        // subjectimiz varsa
        if( $subject_end > 0 )
{
            // döngüde dönen şuanki kelime de subject içerisindeki bir kelimeyse
            if($words[$i]->start>= $subject_start && $words[$i]->total_characters <=
$subject_end )
{
                // Subjectimiz bir kelimelik bir subject ise
                if( $this->tool_model->subject_counter($id,$subject_start, $subject_end)
== 1)
{
                    $type = 0;
                    $tree[] = $type.'|'. $words[$i]->id .'|null' ; $i++;
}
else // birden fazla kelimeden oluşuyorsa
{
                    $type = 1;
}
                // şuanki dönen kelime subjectin başladığı yerse
                if( $words[$i]->start == $subject_start )
{
                    $counter = 0;
}
}
else
{
                $type = 2; // subject değil
}
}
else
{
            $type = 2; // subject değil
```

```
}
// tümce conj veya c si bulunmayan standart tümce (özneli veya öznesiz farketmez)
if(count($tree) > 0){ $array_end_type = substr(end($tree),0,1); }else{ $array_end_type
= 10; }
            if($counter == 0)
{
                if($words[$i+1]->fe_id == 75 ||  $words[$i+1]->fe_id == 39)
                                        {
                                                $tree[] = '0|'. $words[$i]->fe_id .'|null';
                                        }
                                        else
                                        {
                                                $need = $words[$i]->id; $counter++;
                                        }
}else{
                if( $array_end_type != $type )
                                        {
        if( $this->tool_model->check_rules( $need, $words[$i]->id ) ){
                $tree[] = $type.'|'. $this->tool_model->get_equal( $need, $words[$i]->id
) .'|null';
                                        }
                                        else
                                        {
                    die('Kural eksikliği var. Ağaç çıkarılamıyor.'.$this->tree_model-
>get_rule_name($need).','.$this->tree_model->get_rule_name($words[$i]->id));
                                        }
                                        }
                                        else
                                        {
                                           $reverse = $tree;
                $reverse = array_reverse($reverse);
                $reverse = explode('|',$reverse[0]);
```

```
                if( $this->tool_model->check_rules( $reverse[1], $words[$i]->id ) )
{
                    $tree[] = $type.'|'.$this->tool_model->get_equal( $reverse[1],
$words[$i]->id ) .'|null';
}
else
{
                    die('Kural eksikliği var. Ağaç çıkarılamıyor.'.$this->tree_model-
>get_rule_name($reverse[1]).', '.$this->tree_model->get_rule_name($words[$i]->id));
}
}
        // düz tümceler
        // öznesi olan veya öznesi null olan tümceler
        if( $subject_end > 0 )
{
            // döngüde dönen şuanki kelime de subject içerisindeki bir kelimeyse
            if($words[$i]->start>= $subject_start && $words[$i]->total_characters <=
$subject_end )
{
                // Subjectimiz bir kelimelik bir subject ise
                if( $this->tool_model->subject_counter($id,$subject_start, $subject_end) >
1)
                 // birden fazla kelimeden oluşuyorsa
{
                    // şuanki dönen kelime subjectin başladığı yerse
                    if( $words[$i]->total_characters == $subject_end )
{
                        $counter = 0;
}
}
}
}
```

```
}
        $data['tree'] = $tree;
        $data['words'] = $words;
        $data['subject'] = $subject_end;
        $data['c'] = $c;
        $this->load->view('tool/tree_view',$data);
}
```



Şekil 4.4 İşaretlenmiş örnek tümcenin otomatik olarak çıkarılmış latis diyagramı

Şekil 4.4'de gösterilen işaretlenen tümcelerin aldığı tematik rollere göre latis diyagramlarını gösteren metoda ait kodlar:

```
function get($id)
{
        echo '<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> ';
        echo '<style>
        body{ font-family:Arial, Helvetica, sans-serif;font-size:13px; }
        .double-lattice{ background:url('.base_url('img/doublelattice.jpg').') no-repeat;
width:700px; height:177px; background-position:50px 0px; }
        .triple-lattice{ background:url('.base_url('img/triplelattice.jpg').') no-repeat;
width:1000px; height:177px; background-position:50px 0px; }
```

```
.lattice-title{ margin-top:80px; color:red; width:50px; font-weight:bold; float:left;}
.lattice-title div{ color : #999; }
.lattice-header{ color:red; width:70px; font-weight:bold; float:left; margin-left:170px;}
.lattice-header div{ color : #999; }
.header{ height:50px; margin-top:20px; }                    .dashed{ background:url('.base_url('img/dashed.png').') no-repeat;  width:250px; height:150px; float:left; }
</style>';
$words = $this->lattice_model->get($id);
$lattice = 0;
for($i = 0; $i<=count($words)-1; $i++)
{
   if( array_key_exists($i+1,$words) )
{
      if( $words[$i]->text == $words[$i+1]->text )
{
         echo '<span style="color:red;">'.$words[$i]->text.'</span> [ '.$this->lattice_model->get_pointer_name( $words[$i]->fe_id ).' ], [ '.$this->lattice_model->get_pointer_name( $words[$i+1]->fe_id ).' ]<br>'; $i++; $lattice = 1;
}
else
{
         echo '<span style="color:red;">'.$words[$i]->text.'</span> [ '.$this->lattice_model->get_pointer_name( $words[$i]->fe_id ).' ]<br>';
}
}
else
{
      echo '<span style="color:red;">'.$words[$i]->text.'</span> [ '.$this->lattice_model->get_pointer_name( $words[$i]->fe_id ).' ]<br>';
}
```

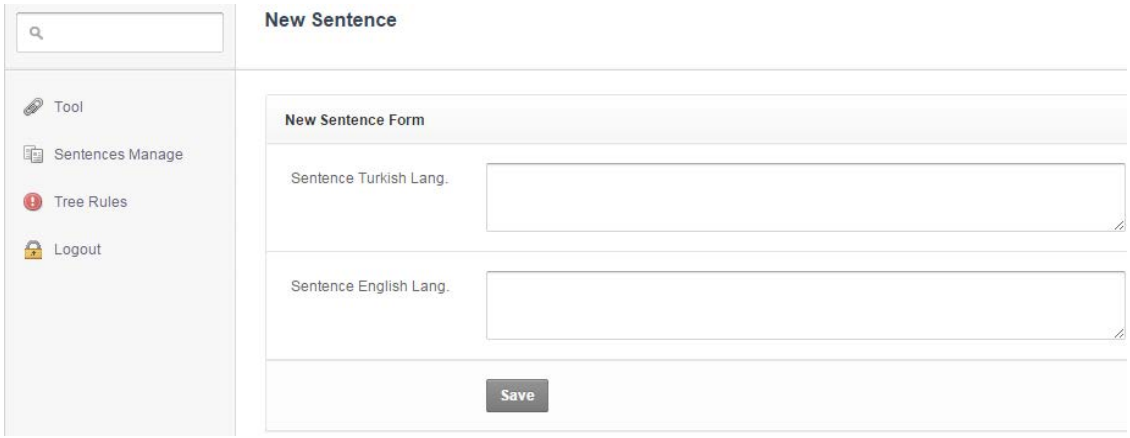
```
}
        echo '...<br>';
        if( $this->lattice_model->exists_manner($id) )
{
            echo '
<div class="header">
<div class="lattice-header" style="margin-left:190px;">Core <div></div></div>
<div class="lattice-header">Non-Core <div></div></div>
<div class="lattice-header">Non-Core <div></div></div>
</div>
<div class="triple-lattice">
<div class="lattice-title">Agent <div>{'.$this->lattice_model-
>exists_agent($id).'}</div></div>
<div class="lattice-title" style="margin-left:190px;">Source <div>{'.$this-
>lattice_model->exists_source($id).'}</div></div>
<div class="lattice-title" style="margin-left:80px;">Patient <div>{'.$this-
>lattice_model->exists_patient($id).'}</div></div>
<div class="lattice-title" style="margin-left:70px;">Manner <div>{'.$this-
>lattice_model->exists_manner($id).'}</div></div>
<div class="lattice-title" style="margin-left:80px;">Goal <div>{'.$this->lattice_model-
>exists_goal($id).'}</div></div>
<div class="lattice-title" style="margin-left:200px;">State <div>{'.$this-
>lattice_model->exists_state($id).'}</div></div>
<div class="dashed" style="margin-left:360px;"><div class="lattice-title"
style="margin:125px 0 0 130px;">{'.$this->lattice_model->exists_patient($id).'}
<div></div></div></div></div>
</div>';
}
else
{
            echo '
<div class="header">
```

```
<div class="lattice-header" style="margin-left:190px;">Core <div></div></div>
<div class="lattice-header">Non-Core <div></div></div>
</div>
<div class="double-lattice">
<div class="lattice-title">Agent <div>{'.$this->lattice_model-
>exists_agent($id).'}</div></div>
<div class="lattice-title" style="margin-left:190px;">Source <div>{'.$this-
>lattice_model->exists_source($id).'}</div></div>
<div class="lattice-title" style="margin-left:80px;">Patient <div>{'.$this-
>lattice_model->exists_patient($id).'}</div></div>
<div class="lattice-title" style="margin-left:200px;">Goal <div>{'.$this-
>lattice_model->exists_goal($id).'}</div></div>
<div class="dashed" style="margin-left:60px;"><div class="lattice-title"
style="margin:125px 0 0 130px;">{'.$this->lattice_model->exists_source($id).'}
<div></div></div></div></div>                    <div class="dashed" style="margin-
left:60px;"><div class="lattice-title" style="margin:125px 0 0 130px;">{'.$this-
>lattice_model->exists_goal($id).'} <div></div></div></div>
</div>';
}
}
```

Ayrıca uygulamamızda derlemimize ait tümcelerin güncellenmesi, yeni tümcelerin eklenmesi ve silinmesi için gerekli arayüzler de tasarlanmış ve kodlanmıştır. Bu arayüzler Şekil 4.5 ve Şekil 4.6'da görülür.



Şekil 4.5 Tümce güncelleme ve silme arayüzü



Şekil 4.6 Tümce ekleme arayüzü

Şekil 4.5 ve Şekil 4.6'daki arayüzlere ait tümce işlemleri ile ilgili class'a ait kodlar:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Sentences extends CI_Controller {
        function __construct()
        {
                parent::__construct();
                $this->load->model('sentences_model');
                $this->load->library('form_validation');
                check_login();
        }
```

```
/*
Sentences controlleri'nın içerisinde bulunan index methodu Sentences modelinden
tümce listesini ister ve list_view isimli view dosyasına paslar.
*/
        public function index()
    {
        $data['sentences'] = $this->sentences_model->sentences_list(); // Model
aracılığıyla tümceleri aldık data dizisine gönderdik.
        $this->load->view('block/header');
        $this->load->view('block/sidebar');
        $this->load->view('sentences/list_view',$data);
        $this->load->view('block/footer');
    }
/*
Create methodu yeni tümce ekleme formunu içeren viewi yükler.
*/
        function create()
    {
        $this->load->view('block/header');
        $this->load->view('block/sidebar');
        $this->load->view('sentences/create_view'); // Yeni tümce ekleme formu
        $this->load->view('block/footer');
    }
        function update( $sentence_id = null)
    {
        if( is_numeric($sentence_id) )
        {
            $data['sentence'] = $this->sentences_model->get_sentence(
$sentence_id );
            $this->load->view('block/header');
            $this->load->view('block/sidebar');
            $this->load->view('sentences/update_view',$data);
```

```
                $this->load->view('block/footer');
        }
                }
/*
create_process methodu formdan ajax çağrısıyla gelen Türkçe ve İngilizce tümcenin validation filtresinden geçirerek veritabanına ekleme işlemini yapar.
*/
function create_process()
        {
                $rules = array(
                                        array(
                                                'field' => 'sentence_tr',
                                                'label' => 'Turkish sentence',
                                                'rules' =>
'required|min_length[5]|xss_clean|htmlspecialchars|trim'
                                                ),
                                        array(
                                                'field' => 'sentence_en',
                                                'label' => 'English sentence',
                                                'rules' =>
'required|min_length[5]|xss_clean|htmlspecialchars|trim'
                                                )
                                );

                $this->form_validation->set_rules( $rules );
                if( $this->form_validation->run() == FALSE )
                {
                        echo '<div class="albox
errorbox"><ul>'.validation_errors('<li>','</li>').'</ul></div>';
                }
                else
                {
```

```
                        if( $this->sentences_model->create_sentence() )
                        {
                                $this->session->set_flashdata('message','<div class="albox
succesbox">Create succesful</div>');
                                echo '<script>location.href =
'".base_url('sentences/index')."';</script>';
                        }
                }
        }
/*
```
update_process methodu formdan ajax çağrısıyla gelen Türkçe ve İngilizce tümcenin validation filtresinden geçirerek veritabanına güncelleme işlemini yapar.
```
*/
        function update_process()
        {
                $rules = array(
                                                array(
                                                        'field' => 'sentence_tr',
                                                        'label' => 'Turkish sentence',
                                                        'rules' =>
'required|min_length[5]|xss_clean|htmlspecialchars|trim'
                                                        ),
                                                array(
                                                        'field' => 'sentence_en',
                                                        'label' => 'English sentence',
                                                        'rules' =>
'required|min_length[5]|xss_clean|htmlspecialchars|trim'
                                                        ),
                                                array(
                                                        'field' => 'id',
                                                        'label' => 'Sentence ID',
```

```
                                                    'rules' =>
'required|numeric|xss_clean|htmlspecialchars|trim'
                                                    )
                                            );
                        $this->form_validation->set_rules( $rules );
        if( $this->form_validation->run() == FALSE )
        {
                echo '<div class="albox
errorbox"><ul>'.validation_errors('<li>','</li>').'</ul></div>';
        }
        else
        {
                if( $this->sentences_model->update_sentence( $this->input-
>post('id') ) )
                {
                        $this->session->set_flashdata('message','<div class="albox
succesbox">Update succesful</div>');
                        echo '<script>location.href =
'".base_url('sentences/index')."';</script>';
                }
        }
    }
/*
delete methodu parametre olarak bir tümce id numarası alır ve ilgili id numaralı
tümceyi veritabanından siler.
*/
function delete( $sentence_id )
    {
        if( is_numeric( $sentence_id ) )
        {
                if( $this->sentences_model->delete( $sentence_id ) )
                {
```

```
                    $this->session->set_flashdata('message','<div class="albox
succesbox">Delete Succesful</div>');
                    redirect(base_url('sentences/index'),'refresh');
                }
                else
                {
                    $this->session->set_flashdata('message','<div class="albox
errorbox">Sentence can not be found</div>');
                    redirect(base_url('sentences/index'),'refresh');
                }
            }
        }
}
```

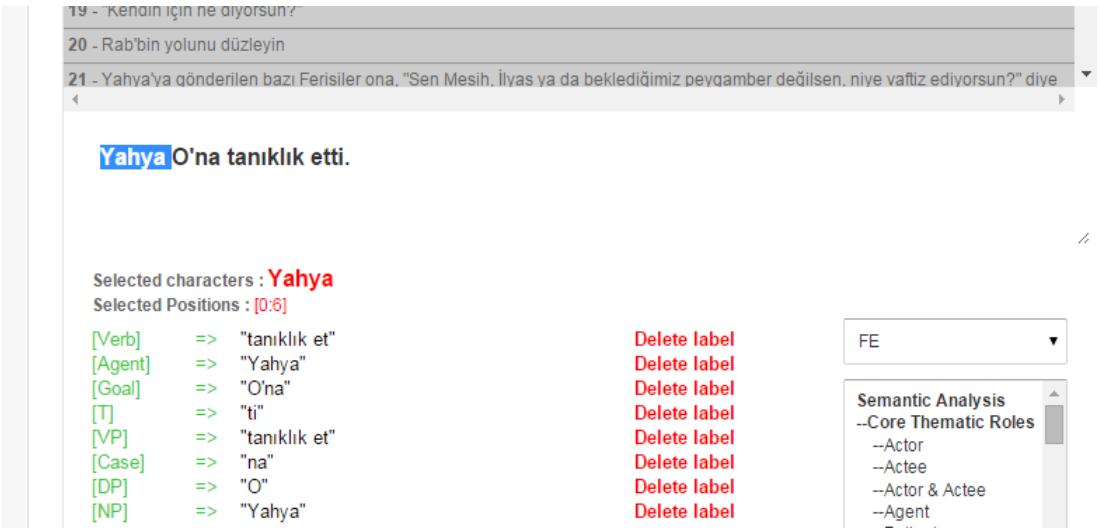
Tümce ekleme çıkarmanın veritabanı işlemlerini yapan model class'ına ait kodları:

```
class sentences_model extends CI_Model
{
/*
Modeldeki sentences_list methodu tümceleri veritabanından çekecek query
tümcesini çalıştırır ve dönen sonuçları döndürür.
*/
function sentences_list()    {
        $sentences = $this->db->query("SELECT * FROM sentences ORDER
BY id asc")->result();
        return $sentences;
    }
/*
Modeldeki create_sentence methodu gelen tümcenin veritabanına ekleme işlemini
yapar.
*/
```

```
function create_sentence()
	{
		$data = array(
'sentence_tr' => $this->input->post('sentence_tr'),
'sentence_en' => $this->input->post('sentence_en')
);

		return $this->db->insert('sentences',$data);
	}
/*
Modeldeki update_sentence methodu gelen tümcenin veritabanında güncelleme işlemini yapar.
*/
function update_sentence( $sentence_id )
	{
		$data = array(
'sentence_tr' => $this->input->post('sentence_tr'),
'sentence_en' => $this->input->post('sentence_en')
);
		$condition = array('id' => $sentence_id);
		return $this->db->update('sentences', $data, $condition);
	}
/*
delete methodu tümce id sini parametre olarak alır ve ilgili tümceyi veritabanından silme işlemini yapar.
*/
function delete( $sentence_id )
	{
```

```
            $this->db->where('id' , $sentence_id);
            return $this->db->delete('sentences');
        }
            function get_sentence( $sentence_id )
        {
            return $this->db->get_where('sentences',array('id' => $sentence_id))-
>row();
        }
            function sentence_check($sentence)
        {
            return $this->db->get_where('sentences',array('sentence_tr' =>
$sentence))->num_rows();
        }
}
```



Şekil 4.7 Örnek bir tümcenin işaretleme arayüzü

Şekil 4.7'deki arayüzün işaretleme yapma class'ına ait kodları:

```
class Tool extends CI_Controller {
    function __construct()
    {
        parent::__construct();
```

```
                $this->load->model('tool_model');
                $this->load->model('sentences_model');
        $this->load->model('tree_model');
                $this->load->library('form_validation'); // Form doğrulama kütüphanesini
çağırıyoruz.
                check_login();
        }
                public function index()
        {
                $data['sentences'] = $this->sentences_model->sentences_list(); // Model
aracılığıyla tümceleri aldık data dizisine gönderdik.
                $this->load->view('block/header'); // Üst kısım
                $this->load->view('block/sidebar'); // Yan kısım
                $this->load->view('tool/index_view',$data);
                $this->load->view('block/footer'); // alt kısım
        }
        function save()
        {
                $selection = $this->input->post('selected');
                $first     = $this->input->post('first');
        $positions = $this->input->post('positions');
                if($first == '' || $selection == '')
                {
                        echo '<span style="font-weight:bold;color:red;">Please
select</span>';
                }
                elseif($first == 'lu')
                {
                        if( $this->tool_model->save_lexical_unit() )
                        {
                                echo '<span style="font-weight:bold;color:green;">Save
Succesful <br>['.$this->input->post('selected').']</span>';
```

```
                        }
                }
                elseif($first == 'fe' && $this->input->post('fe') == '')
                {
                        echo '<span style="font-weight:bold;color:red;">Select frame
element</span>';
                }
        elseif( $positions == '' )
{
        echo '<span style="font-weight:bold;color:red;">Positon error</span>';
}
                else
                {
                        if( $this->tool_model->save_frame_element() )
                        {
                                echo '<span style="font-weight:bold;color:green;">Save
Succesful <br>['.$this->input->post('selected').']</span>';
                        }
                }
        }
/*
check_selected_item methodu taglenen öğeleri Tool Modelinden alır ve listesini
döndürür.
*/
function check_selected_item()
        {
                $item_data = $this->tool_model->get_tagged_items();
                $items = '<table cellspacing="5">';
                foreach($item_data as $item)
                {
                        $items .= '<tr><td>['.$item->pointer_name.']</td><td> 
=>  </td><td width="285">  <span style="color:#000;">"'.$item-
```

```
>text.'"</span>  </td><td> <a href="javascript:void(0)"
onclick="delete_label('.$item->id.')">Delete label</a></td></tr>';
                }
                echo $items.'</table>';
        }
/*
```

delete_label methodu id numarası verilen tagleme işlemini siler. Önce var olup olmadığına bakar daha sonrada silip "ok" döndürür.

```
*/
function delete_label( $label_id )
        {
                if( $this->tool_model->exist_label_id($label_id) > 0 )
                {
                        if( $this->tool_model->delete_label($label_id) )
                        {
                                echo 'ok';
                        }
                }
        }
        function tree( $id )
        {
                // böyle bir sentence var mı kontrol edilir. varsa işleme devam ettirilir.
                echo '<meta http-equiv="Content-Type" content="text/html; charset=utf-
8" /> ';
      $words = $this->tool_model->get($id);
                $subject = $this->tool_model->exist_subject($id);
                $conj    = $this->tool_model->exist_conj($id);
      $c       = $this->tool_model->exist_c($id);
      $tree = array();
      $counter = 0;
                $conj_counter = 0;
      $need = null;
```

```
      if($c == 0 && $conj == 0)
{
         for($i=0;$i<=count($words)-1;$i++)
{
            if($counter == 0)
{
                                    $need = $words[$i]->id;
}
else
{
               if(count($tree) == 0)
{
                                          if( $this->tool_model->check_rules( $need,
$words[$i]->id ) )
{
                     $tree[] = $this->tool_model->get_equal( $need, $words[$i]->id );
}
else
{
                     print_r($tree);
                     die('Kural eksikliği var. Ağaç çıkarılamıyor.'.$this->tree_model-
>get_rule_name($need).','.$this->tree_model->get_rule_name($words[$i]->id));
}
}
                                 else
                                 {
                  if( $this->tool_model->check_rules( end($tree), $words[$i]->id ) )
{
                     $tree[] = $this->tool_model->get_equal( end($tree), $words[$i]->id );
}
else
{
```

```
                    print_r($tree);
                                                    die('Kural eksikliği var. Ağaç çıkarılamıyor.'.$this->tree_model->get_rule_name(end($tree)).', '.$this->tree_model->get_rule_name($words[$i]->id));
}
}
}
            $counter++;
}
}
      print_r($tree);
      //$this->load->view('tool/tree_view',$data);
}
/*
İd si verilen tümcenin kelimelerini Tool Modelindeki get methodundan ister.
*/
function check($id)
        {
                print_r($this->tool_model->get($id));
        }
        function sub()
        {
                echo $this->tool_model->exist_subject(5);
        }
        function aktar()
        {
                $rules = $this->db->get('tree_rules')->result();
                $i = 1;
                foreach($rules as $rule)
                {/*
                        if($rule->pointer1 != '')
                        {
```

```
                $this->db->where('pointer_name',$rule->pointer1);
                $row = $this->db->get('pointers')->row();
                if(empty($row))
                {
                        echo 'aranan : '.$rule->pointer1.' - Bulunan : --------
--------------------------<br>';
                }
                else
                {
                        //echo 'aranan : '.$rule->pointer1.' - Bulunan :
'.$row->pointer_name.' ('.$row->id.')<br>';
                        $this->db->update('tree_rules',array('pointer1' =>
$row->id),array('pointer1' => $rule->pointer1)); echo 'eklendi --<br>';
                }
        }
        if($rule->pointer2 != '')
        {
                $this->db->where('pointer_name',$rule->pointer2);
                $row = $this->db->get('pointers')->row();

                if(empty($row))
                {
                        echo 'aranan : '.$rule->pointer2.' - Bulunan : --------
--------------------------<br>';
                }
                else
                {
                        //echo 'aranan : '.$rule->pointer1.' - Bulunan :
'.$row->pointer_name.' ('.$row->id.')<br>';
                        $this->db->update('tree_rules',array('pointer2' =>
$row->id),array('pointer2' => $rule->pointer2)); echo 'eklendi --<br>';
                }
```

```
                        }
                        if($rule->pointer3 != '')
                        {
                                $this->db->where('pointer_name',$rule->pointer3);
                                $row = $this->db->get('pointers')->row();
                        if(empty($row))
                                {
                                        echo 'aranan : '.$rule->pointer3.' - Bulunan : --------
--------------------------<br>';
                                }
                                else
                                {
                                        //echo 'aranan : '.$rule->pointer1.' - Bulunan :
'.$row->pointer_name.' ('.$row->id.')<br>';
                                        $this->db->update('tree_rules',array('pointer3' =>
$row->id),array('pointer3' => $rule->pointer3)); echo 'eklendi --<br>';
                                }
                        }
                        */
                        if($rule->equal != '')
                        {
                                $this->db->where('pointer_name',$rule->equal);
                                $row = $this->db->get('pointers')->row();
                                if(empty($row))
                                {
                                        echo 'aranan : '.$rule->equal.' - Bulunan : ----------
----------------------<br>';
                                }
                                else
                                {
                                        //echo 'aranan : '.$rule->equal.' - Bulunan : '.$row-
>pointer_name.' ('.$row->id.')<br>';
```

```
                                        $this->db->update('tree_rules',array('equal' =>
$row->id),array('equal' => $rule->equal)); echo 'eklendi --<br>';
                                    }
                                }
                            }
                        }
}
```

Kontrol paneline ve ağaçlara ait kuralların belirlenmesine yönelik metot ve fonksiyonlara ait kodlar aşağıda verilmiştir:

```
class Member extends CI_Controller {
/*
Member fonksiyonu Member classıyla aynı ismi taşıdığı için constructer görevini
yapmakta. Sınıf instance edilirken user_model'ini ve form_validation
kütüphanesini class içerisine yüklemektedir. Class içerisindeki bir methodda bu
kütüphane ve modeldeki methodlara load yapılmadan erişilebilir.
*/
   function Member()
{
        parent::__construct();
        $this->load->model("user_model");
        $this->load->library('form_validation');
}
/*
Member dizinine girildiğinde member/login e yönlendirme yapmaktadır.
*/
        function index()
        {
                redirect(base_url('member/login'),'refresh');
        }
   /*
```

Login methodu is_login() methodunu kullanır. is_login true ise giriş sayfasını göstermek yerine Framenet projesi anasayfasına yönlendirir. Değilse login view dosyasını dahil eder.

***/**

```
   function login()
{
                if( is_login() == 1 ){ redirect(base_url('tool/index'),'refresh'); }
      $this->load->view('login');
}
```

/*

Login_process methodu ajax isteğiyle gelen form değerlerini belirli validation kurallarına tabi tutarak kontrollerini gerçekleştirir. validation hataları varsa hataları geri döndürür. E-mailin geçerli yazılıp yazılmadığını, sistemde öyle bir e-mail adresinin olup olmadığını, parolanın yeterli karakterde olup olmadığını, e-mail ile parolanın sistemdeki bilgilerle birebir eşleşip eşleşmediğini kontrol eder.

Tüm validation kuralları geçerli ise session için bir array hazırlar ve session'a bunları set eder. Bundan sonra kimlik doğrulama işlemleri bu set edilen değerlere göre yapılacaktır. En sonda da /tool/index linkine yönlendirme yapar.

***/**

```
   function login_process()
{
      if( $this->form_validation->run('login_user') == FALSE )
{
         echo '<div class="albox errorbox"><ul>' . validation_errors('<li>','</li>') .
'</ul></div>';
}
      elseif( $this->user_model->exists_email( $this->input->post('email') ) == 0 )
{
         echo '<div class="albox errorbox">You must create an account. </div>';
}
      elseif( $this->user_model->check_user_detail() == FALSE )
```

```
{
        echo '<div class="albox errorbox">Invalid E-mail or password</div>';
}
else
{
        $userdata = $this->user_model->user_data( $this->input->post('email') );
        $session_data = array(
                        "username"   => $userdata->email,
                        "userhash"   => md5( $userdata->password.$this->config-
>item('password_hash')  ),
                        "department" => $userdata->department
                        );

        $this->session->set_userdata($session_data);
        echo '<script>location.href="'.base_url('tool/index').'";</script>';
}
}
/*
Logout methodu session değerlerini sıfırlar ve login sayfasına yönlendirme yapar.
*/
   function logout()    {
      $session_items = array(
                                                'username' => '',
                                                'userhash'     => '',
                    'department'     => '',
                                                'logged_in'=> FALSE

                                                );
                $this->session->unset_userdata($session_items);
                redirect(base_url()."member/login","refresh");
}
}
```

/*

Sayfalar açılırken hep check_login() fonksiyonu tetiklenir. Bu fonksiyon user_modelini içine dahil eder. Session daki username ve userhash değerlerini okur. İkisinden biri bile boş olsa login sayfasına yönlendirir. Doluysa mailin ve userhash'inin geçerli olup olmadığını kontrol ederek gerekli yönlendirme işlemlerini gerçekleştirir.

***/**

```
 function check_login()
{
      $CI =& get_instance();
      $CI->load->model("user_model");
      $username = $CI->session->userdata('username');
        $userhash = $CI->session->userdata('userhash');
      if($username == '' || $userhash == '')
        {
                redirect(base_url().'member/login','refresh');
                exit();
        }
        elseif( $CI->user_model->exists_email( $username ) == 0 )
        {
                redirect(base_url().'member/login','refresh');
                exit();
        }
        else
        {
                $userdata  = $CI->user_model->user_data( $username );
                $hash_db  = md5($userdata->password.$CI->config-
>item('password_hash'));
                if($userhash != $hash_db)
                {
                        redirect(base_url().'member/login','refresh');
                        exit();
```

```
                }
        }
}
/*
is_login fonksiyonu login olunup olunmadığını boolean tipinde döndürür.
*/
 function is_login()
        {
      $CI =& get_instance();
                $CI->load->model("user_model");
                $username = $CI->session->userdata('username');
                $userhash = $CI->session->userdata('userhash');
                if( $username == '' || $userhash == '' )
                {
                        return 0;
                }
                else
                {
                        $userdata  = $CI->user_model->user_data( $username );
                $hash_db  = md5($userdata->password.$CI->config->item('password_hash'));
                        if( $CI->user_model->exists_email( $username ) == 0 )
                        {
                                return 0;
                        }
                        elseif( $userhash != $hash_db )
                        {
                                return 0;
                        }
                        elseif( ( $CI->user_model->exists_email( $username ) == 1 ) && $userhash == $hash_db )
                        {
```

```
                    return 1;
                }
            }
        }
```

**/*
Userdata fonksiyonu parametre olarak verilen column name lerine göre login olan kullanıcının bilgilerini alma işlemini üstlenir. Eğer parametre verilmezse default olarak kullanıcı id numarasını döndürür.
*/**

```
  function userdata( $field = 'id' )    {
      $CI =& get_instance();
            $CI->load->model("user_model");
            $username = $CI->session->userdata('username');
      $userdata = $CI->user_model->user_data( $username );
      return $userdata->$field;
}
```

**/*
rule_name fonksiyonu parametre olarak verilen kuralın id sine göre o kuralın kısa ismini döndürür.
*/**

```
  function rule_name($rule_id)
{
      $CI =& get_instance();
      $CI->load->model('tree_model');
      return $CI->tree_model->get_rule_name($rule_id);
}
class User_model extends CI_Model {
   function __construct()
{
      parent::__construct();
}
```

```
/*
```
exists_email parametre olarak bir e-mail stringi alır. Ve bunu db deki sayısını döndürür. 0 ya da 1 döndürecektir.

```
*/
   function exists_email( $email )
{
                $email_count = $this->db->get_where('users',array('email' => $email))->num_rows();
                return $email_count;
}
/*
```
Formdan gelen e-mail ve şifrenin db dekiler ile uyuşup uyuşmadığını true ya da false olarak döndürür.

```
*/
   function check_user_detail()
{
      $username = $this->input->post('email');
      $password = $this->input->post('password');
      $userdata = $this->user_data( $username );
      if( $userdata->email == $username && $userdata->password == md5($password) && $userdata->status == 1)
{
         return true;
}
else
{
         return false;
}}
/*
```
get_user methodu parametre olarak kullanıcı id si alır. İlgili kullanıcının bilgileri bulunduğu satırı obje olarak döndürür.

```
*/
```

```
function get_user( $user_id )
        {
                return $this->db->get_where('users',array('id' => $user_id))->row();
        }
/*
```

check_user_id methodu parametre olarak kullanıcı id numarası alır. Sonuç olarak kullanıcının sistemde olup olmadığını döndürür.

```
*/
function check_user_id( $user_id )
{
      return $this->db->get_where('users',array('id' => $user_id))->num_rows();
}
/*
```

user_data methodu parametre olarak e-mail stringi alır ve o e-maile ait kullanıcının bilgilerini obje olarak döndürür.

```
*/
function user_data( $username )
{
      return $this->db->get_where('users',array('email' => $username))->row();
}
}
class tree_model extends CI_Model {
/*
```

Tree Model classındaki rules methodu kuralların bir listesini objeler halinde döndürür.

```
*/
function rules()
        {
                $this->db->order_by('id','desc');
                return $this->db->get('tree_rules')->result();
        }
```

```
/*
get_rule_name methodu parametre olarak id numarası verilen kuralın bilgilerinin bulunduğu satırı obje olarak döndürür.
*/
function get_rule_name( $rule_id )
{
		$row = $this->db->get_where('pointers',array('id' => $rule_id))->row();
		return $row->pointer_name;
}
/*
pointers methodu pointers tablosundan kayıtları döndürür.
*/
function pointers()
	{
		$this->db->where_not_in('id',array(1,2,3,4,5,6,7,8,80,42,43));
		 $this->db->where_not_in('parent_id',array(4,5,6,7,42,43));
		return $this->db->get('pointers')->result();
	}
/*
check_pointer_id methodu parametre olarak pointer id numarasını alır. Ve o pointerın veritabanında olup olmadığını döndürür.
*/
function check_pointer_id($pointer_id)
	{
		return $this->db->get_where('pointers',array( 'id' => $pointer_id ))->num_rows();
	}
/*
check_rule_id methodu parametre olarak kuralın id numarasını alır. Ve o kuralın db de olup olmadığını kontrol eder.
*/
function check_rule_id($id)
```

```
        {
                return $this->db->get_where('tree_rules',array( 'id' => $id ))-
>num_rows();
        }
```
**/*
create_rule methodu gerekli validationları sağladıktan sonra yeni kuralı database'e ekler.
*/**

```
function create_rule()
        {
      if( $this->input->post('text_equal') )
{
        $equal = $this->new_pointer();
}
else
{
        $equal = $this->input->post('equal');
}
      $data = array(  'pointer1' => $this->input->post('pointer1'),
                                                'pointer2' => $this->input->post('pointer2'),
                                                'pointer3' => $this->input->post('pointer3'),
                                                'equal' => $equal
                                                );

      if( $this->db->get_where('tree_rules',$data)->num_rows() > 0)
{
        return false;
}
else
{
        return $this->db->insert('tree_rules',$data);
}
```

```
                        }
/*
Formdan gelen kural verilerini db deki kuralın üzerine yazar ve güncelleme
işlemini gerçekleştirir.
*/
   function update_rule()
         {
                  if( $this->input->post('text_equal') )
{
         $equal = $this->new_pointer();
}
else
{
         $equal = $this->input->post('equal');
}
      $data = array(  'pointer1' => $this->input->post('pointer1'),
                                                'pointer2' => $this->input->post('pointer2'),
                                                'pointer3' => $this->input->post('pointer3'),
                                                'equal' => $equal
                                                );

      if( $this->db->get_where('tree_rules',$data)->num_rows() > 0)
{
         return false;
}
else
{
         return $this->db->update('tree_rules',$data,array('id' => $this->input-
>post('rule_id')));
}

         }
```

```
/*
Parametre olarak kural id si verilir ve o kuralı veritabanından siler.
*/
function delete_rule( $id )
        {
                return $this->db->delete('tree_rules',array('id' => $id));
        }
/*
get_rule methodu parametre olarak kuralın idsini alır ve o kuralın detayını obje olarak döndürür.
*/
function get_rule($id)
{
      return $this->db->get_where('tree_rules',array('id' => $id))->row();
}
/*
new_pointer methodu formdan gelen verilerin validationlarını yaptıktan sonra yeni pointer olarak veritabanına ekler.
*/
function new_pointer()
{
      $data = array('parent_id' => 0,
                'pointer_name' => $this->input->post('text_equal'),
                'visible' => 1);
      $this->db->insert('pointers',$data);
      return $this->db->insert_id();
}}
class tool_model extends CI_Model {
/*
Tümcenin id numarasına göre frame elementi olup olmadığını kontrol eder.
*/
function exist_frame_element()
```

```
{
                return $this->db->get_where('tagged_items',array('text' => $this->input->post('selected'), 'fe_id' => $this->input->post('fe'),'sentence_id'=>$this->input->post('sentence_id')))->num_rows();
        }
/*
```

Frame elementi veritabanına kaydeder.

```
*/
function save_frame_element()
        {
      $find      = array('[',']');
      $replace   = array("","");
            $positions = str_replace($find,$replace,$this->input->post('positions'));
      list($start,$end) = explode(':',$positions);
                $sentence_data = $this->db->get_where('sentences',array('id' => $this->input->post('sentence_id')))->row();
                $data = array('sentence_id' => $this->input->post('sentence_id'),
                                        'fe_id'       => $this->input->post('fe'),
                                        'text'        => $this->input->post('selected'),
                                        'start'             => $start,
                                        'total_characters' => $end);

                return $this->db->insert('tagged_items',$data);
        }
/*
```

İşaretlenen kelime veya kelime grubuna göre veritabanında arama yapar daha önce o tümceye işaretleme yapılmış mı diye kontrol eder.

```
*/
function check_pointer()
        {
                return $this->db->get_where('tagged_items',array('sentence_id' => $this->input->post('sentence_id'), 'text' => $this->input->post('selected') ))->num_rows();
```

```
        }
/*
get_tagged_items methodu tümcenin id numarasına göre işaretlemeleri obje olarak döndürür.
*/
function get_tagged_items()
        {
                $this->db->order_by('tagged_items.id','asc');
                $this->db->where( array('tagged_items.sentence_id' => $this->input->post('sentence_id')) );
                $this->db->select('tagged_items.id, tagged_items.text, pointers.pointer_name');
        $this->db->from('tagged_items');
        $this->db->join('pointers', 'tagged_items.fe_id = pointers.id');
        return $this->db->get()->result();
        }
/*
exist_label_id methodu parametre olarak label id numarasını alır ve veritabanında o id ye göre bir label var mı sonucunu döndürür.
*/
        function exist_label_id( $label_id )
        {
                return $this->db->get_where('tagged_items',array('id' => $label_id))->num_rows();
        }
/*
delete_label methodu parametre olarak verilen label idsine göre labelı veritabanından siler.
*/
function delete_label( $label_id )
        {
                return $this->db->delete('tagged_items',array('id' => $label_id));
```

```
        }
/*
lexical unit (sözcük birimin) in veritabanına kaydını gerçekleştirir.
*/
function save_lexical_unit()
        {
                $find      = array('[',']');
      $replace   = array("","");
            $positions = str_replace($find,$replace,$this->input->post('positions'));
      list($start,$end) = explode(':',$positions);
                $sentence_data = $this->db->get_where('sentences',array('id' => $this-
>input->post('sentence_id')))->row();
                $data = array('sentence_id' => $this->input->post('sentence_id'),
                                        'fe_id'       => 74,
                                        'text'        =>  $this->input->post('lu'),
                                        'start'             => $start,
                                        'total_characters' => $end);

                return $this->db->insert('tagged_items',$data);
        }
/*
Tümcenin işaretlenen öğelerinin tümünü objeler halinde döndürür.
*/
function get($id)
        {
                $condition = array('tagged_items.sentence_id' => $id);
                $not_in = array(0,4,5,6,7,42,43,80);
                $this->db->order_by('tagged_items.start','asc');
                $this->db->where( $condition );
                $this->db->where_not_in('pointers.parent_id',$not_in);
```

```
                $this->db->select('tagged_items.sentence_id, tagged_items.fe_id,
tagged_items.text, tagged_items.start, tagged_items.total_characters, pointers.id,
pointers.pointer_name');
        $this->db->from('tagged_items');
        $this->db->join('pointers', 'tagged_items.fe_id = pointers.id');
return $this->db->get()->result();
            }
/*
Parametre olarak verilen kuralların doğruluğunu veritabanından kontrol eder ve
sonucunu döndürür.
*/
    function check_rules( $first, $second = 0 , $third = 0 )
{
        return $this->db->get_where('tree_rules',array('pointer1' => $first, 'pointer2' =>
$second, 'pointer3' => $third))->num_rows();
}
/*
get_pointer_name methodu parametre olarak işaretçi numarası alır ve
veritabanındaki ismini döndürür.
*/
        function get_pointer_name($pointer_id)
{
        $row = $this->db->get_where('pointers',array('id' => $pointer_id))->row();
        return $row->pointer_name;
}
/*
exist_subject verilen tümce id numarasına ait bir subject var mı kontrol edip
sonucunu döndürür.
*/
function exist_subject($sentence_id,$field = 'end')
            {
```

```
                if( $this->db->get_where('tagged_items',array('sentence_id' =>
$sentence_id,'fe_id' => 81))->num_rows() > 0)
                {
                        $row = $this->db->get_where('tagged_items',array('sentence_id'
=> $sentence_id,'fe_id' => 81))->row();

        if($field == 'start')
{
    return $row->start;
}
else
{
    return $row->total_characters;
}
                }
                else
                {
                        return '0';
                }
        }
/*
exist_conj verilen tümce id numarasına ait bir conj var mı kontrol edip sonucunu
döndürür.
*/
function exist_conj($sentence_id)
        {
                if( $this->db->get_where('tagged_items',array('sentence_id' =>
$sentence_id,'fe_id' => 75))->num_rows() > 0)
                {
                        $row = $this->db->get_where('tagged_items',array('sentence_id'
=> $sentence_id,'fe_id' => 75))->row();
```

```
                return $row->total_characters;
            }
            else
            {
                return '0';
            }
        }
/*
exist_c verilen tümce id numarasına ait bir c var mı kontrol edip sonucunu döndürür.
*/
 function exist_c($sentence_id)
        {
            if( $this->db->get_where('tagged_items',array('sentence_id' =>
$sentence_id,'fe_id' => 39))->num_rows() > 0)            {
                $row = $this->db->get_where('tagged_items',array('sentence_id'
=> $sentence_id,'fe_id' => 39))->row();
                return $row->total_characters;
            }
            else
            {
                return '0';
            }
        }
/*
Tümcedeki işaretlenmiş olan subject sayısını döndürür.
*/
   function subject_counter($id,$start,$end)
   {
        $condition = array('sentence_id' => $id, 'start >=' => $start, 'total_characters
<=' => $end);
            $not_in = array(0,4,5,6,7,42,43,80);
```

```
                $this->db->where( $condition );
                $this->db->where_not_in('pointers.parent_id',$not_in);
                $this->db->select('tagged_items.sentence_id, tagged_items.text,
tagged_items.start, tagged_items.total_characters, pointers.id, pointers.pointer_name');
        $this->db->from('tagged_items');
        $this->db->join('pointers', 'tagged_items.fe_id = pointers.id');
        return $this->db->get()->num_rows();
    }
}
class lattice_model extends CI_Model {
/*
get_tree_words methodu verilan id numarasına göre o tümceye ait ağaç öğelerini
obje olarak döndürür.
*/
function get_tree_words($id)
        {
                $condition = array('tagged_items.sentence_id' => $id);
                $not_in = array(0,4,5,6,7,42,43,80);
                $this->db->order_by('tagged_items.start','asc');
                $this->db->where( $condition );
                $this->db->where_not_in('pointers.parent_id',$not_in);
                $this->db->select('tagged_items.sentence_id, tagged_items.fe_id,
                tagged_items.text, tagged_items.start, tagged_items.total_characters,
                pointers.id, pointers.pointer_name');
                $this->db->from('tagged_items');
        $this->db->join('pointers', 'tagged_items.fe_id = pointers.id');

        return $this->db->get()->result();                }
/*
exist_agent methodu id numarası verilen tümcede agent var mı kontrolünü yapıp
döndürür. Agent var ise agent in adını döndürür.
*/
```

```
function exists_agent($id)
   {
      $record = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 47))->num_rows();
      if($record > 0)
      {
         $data = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 47))->row();
                  return $data->text;
      }
      else
      {
         return 0;
      }
   }
/*
exists_source parametre olarak tümce id si alır ve o tümcede source var mı
kontrolünü yapar. Var ise ilgili source un detayını obje olarak döndürür.
*/
   function exists_source($id)
   {
      $record = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 52))->num_rows();
            if($record > 0)
      {
         $data = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 52))->row();
         return $data->text;
      }
      else
      {
         return 0;
```

```
        }
    }
/*
```
exists_patient parametre olarak tümce id si alır ve o tümcede patient var mı kontrolünü yapar. Var ise ilgili patient in detayını obje olarak döndürür.

***/**

```
    function exists_patient($id)
    {
        $record = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 48))->num_rows();
        if($record > 0)
        {
            $data = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 48))->row();
            return $data->text;
        }
        else
        {
            return 0;
        }
    }
/*
```
exists_goal parametre olarak tümce id si alır ve o tümcede goal var mı kontrolünü yapar. Var ise ilgili goal un detayını obje olarak döndürür.

***/**

```
    function exists_goal($id)
    {
        $record = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 53))->num_rows();

        if($record > 0)
```

```
        {
            $data = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 53))->row();
                    return $data->text;
        }
        else
        {
            return 0;
        }
    }
/*
exists_manner parametre olarak tümce id si alır ve o tümcede manner var mı
kontrolünü yapar. Var ise ilgili manner un başlığını döndürür.
*/
    function exists_manner($id)
    {
        $record = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 66))->num_rows();

        if($record > 0)
        {
            $data = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id'
=> 66))->row();
            return $data->text;
        }
        else
        {
            return 0;
        }
    }
```

```
/*
exists_state parametre olarak tümce id si alır ve o tümcede state var mı kontrolünü yapar. Var ise ilgili state in başlığını döndürür.
*/
   function exists_state($id)
   {
      $record = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id' => 71))->num_rows();

      if($record > 0)
      {
         $data = $this->db->get_where('tagged_items', array('sentence_id' => $id, 'fe_id' => 71))->row();
      return $data->text;
      }
      else
      {
         return 0;
      }
   }
}
```

BÖLÜM 5

SONUÇLAR VE TARTIŞMA

Dilbiliminde tümceye ait öğelerin sınıflandırılmasında anlamsal roller büyük önem taşır. Sözcüklerin ya da öğelerin biçimbilimsel / dizimsel ve anlamsal olarak tematik rol latisleri ile modellenmesi bizi gerçek anlamlara götürür. Bu modellerden elde edilen bilgiler, bilgi tabanlarında tutularak FrameNet gibi sözlüksel kaynaklara entegre edilir. Bu şekilde elde edilen anlamsal çıkarımlar doğal dil işleme ve hesaplamalı dilbilmi çalışmalarına önemli bir kaynak teşkil eder.

Bir tümceye ait sözcüklerin gerçek çıkarımlarını elde etmede bu öğelerin biçimbilimsel / dizimsel ve anlamsal rollerinin işaretlenmesi gerekir. Bu anlamda gerçeklik modellerine ihtiyaç duyulur. Bizim tarafımızdan geliştirilen gerçeklik modelinde Biçimsel Kavram Analizi yönteminin kullanıldığı hiyerarşik bir yapı vardır. Tematik roller ise bu yapının temel taşlarıdır. Dolayısıyla matematikselleştirilmiş gerçeklik modelleri elde edilir. Oluşturulan bu matematiksel yapıların uygulanabilmesi için dilin sözdizimsel ve anlamsal analizlerinin sunulduğu bir kaynağa ihtiyaç duyulur. Bunu en iyi sunan kaynak FrameNet'tir. FrameNet sözlüğü anlamsal çerçeveler üzerine oturtulmuş leksigokrafik bir sözlüktür. Bu sözlük anlamsal rolleri dildeki sözcüklerin anlamsal karşılıklarını belirlemek suretiyle oluşturur. Eğer dilden görece bağımsız anlamsal yapılar istenirse bizim oluşturduğumuz gerçeklik modellerine bakmak gerekir. Oluşturduğumuz bu model ile FrameNet'in dil ile etkileşimi sağlanarak insana ait zihin yapısını veren ontolojiler ile dil arasındaki bağ da açık bir şekilde kurulur. Yine bu çalışmayla Türkçe'nin anlamsal çözümlenmesine de katkısı sağlanarak Türkçe sözcüklerin mevcut FrameNet sözlüğündeki çerçevelerle eşleştirmesi yapılır.

Bu tezde yapılan uygulama çalışması, derleme ait tümcelerin işaretlenmesi ile biçimsel ontoloji tabanlı tematik rol latislerinin otomatik olarak çıkarılmasını kapsar. Böylece Türkçe'deki sözcük birimlerin öğe yapıları ve değerlik değişimleri göz önüne alınarak biçimbilimsel / dizimsel ve anlamsal arayüzü ile Türkçe tümcelerin dilbilgisi

oluşturulur. Bu çalışmayla, Türkçe üzerine çalışan araştırmacılar ileri düzeyde veri tabanına erişme imkanı elde ederler. Yine bu çalışmayla, Türkçe'nin eklemeli ve özne-nesne-yüklem dizilişine sahip olmasından dolayı makine öğrenimindeki ve çevirisindeki güçlüklerin ortadan kaldırılması sağlanır. Böylece hesaplamalı dilbilim uygulamaları için gerekli ön çalışmanın da tamamlanması gerçekleştirilir.

KAYNAKÇA

[1] Jackendoff, R., Semantic Interpretation in Generative Grammar, Cambridge: MIT Press, 1972.

[2] Gruber, J., Studies in Lexical Relations, MIT Dissertation, Bloomington: Indiana University Linguistics Club, [Reprint 1970], 1965.

[3] Fillmore, C. J., Frame semantics and the nature of language, In Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, Volume 280, 20-32, 1976.

[4] Dowty, D., Thematic Proto-Roles and Argument Selection, Language 67, 547-619, 1991.

[5] Jackendoff, R., On the Role of Conceptual Structure in Argument Selection, A Reply to Emonds, Natural Language and Linguistic Theory 11, 279-312, 1993.

[6] Gildea, D., ve Jurafsky, D., Automatic Labeling of Semantic Roles. In: Computational Linguistics, Volume:28, No:3, 245-288, 2002.

[7] Kasper, S., A comparison of 'thematic role' theories, Unpublished master thesis, The Philips University of Malburg, 2008.

[8] Butt, M., Theories of Case. Cambridge, Cambridge University Press, 2005.

[9] Blake, F. R., A Semantic Analysis of Case, In: Curme Volume of Linguistic Studies, Ed. by J. T. Hatfield, W. Leopold, A. J. Friedrich Zieglschmid, [Reprint of Language Monograph No. 7. Published by the Linguistic Society of America], 34-49, 1930.

[10] Chomsky, N., Sprache und Geist, Frankfurt: Suhrkamp, 1970.

[11] Jackendoff, R., The Status of Thematic Relations in Linguistic Theory, Linguistic Inquiry 18, 369-411, 1987.

[12] Croft, W., Syntactic Categories and Grammatical Relations, Chicago/London: University of Chicago Press, 1991.

[13] Wunderlich, D., Predicate Composition and Argument Extension as General Options, A Study in the Interface of Semantic and Conceptual Structure, In: Lexicon in Focus. Ed. By B. Stiebels and D. Wunderlich. Berlin: Akademie, 247-270, 2000.

[14] Pinker, S., Learnability and Cognition: The Acquisition of Argument Structure, Cambridge: MIT Press, 1989.

[15] Fillmore, C., The Case for Case, In: Universals in Linguistic Theory, Ed. by E. Bach and R. Harms. New York, 1968.

[16] Kılıçaslan, Y., A situation-theoretic approach to case marking semantics in Turkish, Lingua, Elsevier, 2006.

[17] DeLancey, S., Notes on Agentivity and Causation, Studies in Language 8, 181-213, 1984.

[18] Van Valin, R. D. jr., The Case for "Effector": Case Roles, Agents and Agency Revisited, In: Grammtical Constructions, Ed. by M. Shibatani and S. Thompson, Oxford: Oxford University Pres, 289-322, 1996.

[19] Butt, M., The Dative-Ergative Connection, In: Empirical Issues in Syntax and Semantics 6. Ed. by O. Bonami and P. Cabredo Hofherr, 69-92, 2006.

[20] Cooper, R., The role of situations in generalized quantifiers. In Lappin, s. (Ed.), Handbook of Contemporary Semantic Theory, Blackwell, 1995.

[21] Barwise, J. ve Perry, J., Situations and Attitudes, MIT Press, Cambridge, MA, 1983.

[22] Goldberg, A., Constructions: A Construction Grammar Approach to Argument Structure, Chicago: University of Chicago Press, 1995.

[23] DeLancey, S., The Universal Basis of Case, Logos and Language 1(2), 1-15, 2000.

[24] Leonard, T., Semantic *S*tructures in English and Atsugewi. Ph.D dissertation, Department of Linguistics, University of California, Berkeley, 1972.

[25] Chomsky, N., Lectures on Government and Binding, Dordrecht: Foris, 1981.

[26] Jackendoff, R., Semantic Structures, Cambridge: MIT Press, 1990.

[27] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html.

[28] Culicover, P. W., ve Wilkins, W., Control, PRO, and the Projection Principle, Language 62, 120-153, 1986.

[29] Foley, W. A., ve Van Valin, R. D. jr., Functional Syntax and Universal Grammar, Cambridge University Press, Cambridge, 1984.

[30] Valverde, F. J., ve Albacete, Explaining the Structure of FrameNet with Concept Lattices, Formal Concept Analysis:Third International Conference, ICFCA 2005, Lens, Volume 3403, 79-94, France, 2005.

[31] Obitko, M., Snasel, V., ve Smid, J., Ontology Design with Formal Concept Analysis, CLA, ISBN 80-248-0597-9, 111-119, Technical University of Ostrava, Dept. Of Computer Science, 2004.

[32] Benes, M., ve Snasel, V., Deducing Design Class Hierarchy from Object Properties, ISM'2002, Rožnov pod Radhoštěm. Czech Republic, 2002.

[33] Cruse, D. A., Some Thoughts on Agentivity, Journal of Linguistics 9, 1-204, 1973.

[34] Fillmore, C., The Case for Case Reopened, In: Syntax and Semantics 8: Grammatical Relations. Ed. by P. Cole and J. Sadock, New York: Academic Press, 59-82, 1977.

[35] MacKenzie, J. L., Places and things, In: Anstey, M. P. and MacKenzie, J. L. (Eds.), Crucial Readings in Functional Gramma, 141-165, Berlin and New York NY: Mouton de Gruyter, Reprint , 2005.

[36] Dzikovska, M. O., Swift, M. D., ve Allen, J. F., Building a computational lexicon and ontology with FrameNet, 2004, http://homepages.inf.ed.ac.uk/mdzikovs/papers/lf-framenet.pdf.

[37] Ganter, B., ve Wille, R., Formal Concept Analysis, Mathematical Foundation, Berlin: Springer, Verlag, 1999.

[38] Pease, A., The sigma ontology development environment, In working notes of the IJCAI-2003 Wkshp. on Ontologies and Distributed System, Acapulco, Mexico, 2003. URL http://sigmakee.sourceforge.net.

[39] Kılıçaslan, Y., ve Güner, E. S., Filtering Machine Translation Results with Automatically Constructed Concept Lattices, Proceedings of 8th International Conference on Concept Lattices and Their Applications (CLA 2011), Nancy, France, 2011.

[40] Basili, R., Cao, D. D., Croce, D., Coppola, B., ve Moschitti, A., Cross-Language Frame Semantics Transfer in Bilingual Corpora, In Computational Linguistics and Intelligent Text Processing, 332-345, 2009.

[41] Scheffczyk, J., Pease, A., ve Ellsworth, M., Linking FrameNet to the Suggested Upper Merged Ontology, Formal Ontology in Information Systems-Proceedings of the Fourth International Conference, Volume:150, 289-300, 2006.

[42] Niles, I., ve Pease, A., Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology, In Proc. of the 2003 Int. Conf. on Information and Knowledge Engineering, 2003.

[43] Tonelli, S., ve Pianta, E., Three issues in cross-language frame information transfer, International Conference RANLP, Borovets, Bulgaria, 441-448, 2009.

[44] Narayanan, S., Baker, C., Fillmore, C., ve Petruck, M., Framenet Meets the Semantic Web:A DAML + OIL Frame Representation, Lecture Notes in Computer Scienece, Volume 2870, 771-787, 2003.

[45] Baker, C. F., Fillmore, C, J., ve Cowe, J. B., The Berkeley FrameNet Project, ACL '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1, 86-90, 1998.

[46] Fillmore, C. J., Johnson, C. R., ve Petruck, M. R. L., Background To FrameNet, Intertational Journal of Lexicography, Volume 16, No. 3, 2003.

[47] Boas, H. C., Bilingual Framenet Dictionaries For Machine Translation, Department of Germanic Studies, E. P. Schoch 3. 102, C 3300, University of Texas at Austin, U.S.A, 2002.

[48] Niles, I., ve Pease, A., Towards a standard upper ontology, In Proc. of the 2nd Int. Conf. on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine, 2001.

[49] Ruppenhofer, J., Sunde, J., ve Pinkal, M., Generating Framenets of various granularities: The Framenet Transformer, The seventh International Conference on Conguage Resources and Evaluation (LREC), Valletta, Malta, May 19-21, 2010.

[50] Kipper, K., Korhonen, A., Ryant, N., ve Palmer, M., Extending Verbnet with Novel Verb Classes, In Proceedings of 5th International Conference on Language Resources and Evaluation, Genova, Italy, 2006.

[51] Scheffczyk, J., Pease, A., ve Ellsworth, M., Linking FrameNet to the SUMO Ontology, International Conference on Formal Ontology in Information Systems, ISBN:1-58603-685-8, 289-300, FOIS 2006.

[52] https://framenet.icsi.berkeley.edu/fndrupal/.

[53] Kiparsky, P., The Rise of Positional Licensing, In: Parameters of Morphosyntactic Change, Ed. by A. v. Kemenade and N. Vincent, Cambridge: Cambridge University Press, 460-494, 1997.

[54] Tenny, Carol, The Aspectual Interface Hypothesis, In Lexical Matters, Ed. by Ivan A. Sag and Anna Szabolcsi, Stanford: CSLI, 1992.

[55] Arad, M., A Minimalist View of the Syntax-Lexical Semantics Interface, University College of London Working Papers in Linguistics, Volume 8, 215-242, 1996.

[56] Baker, M., Theta Theory and the Syntax of Applicatives in Chichewa, Natural Language and Linguistic Theory, Volume 6, Issue 3, 353-390, 1988.

[57] Fillmore, C. J., Ruppenhoffer, J., ve Baker, C. F., FrameNet and Representing the Link between Semantic and Syntactic Relations, In:Frontiers in Linguistics I. Language and Linguistics Monograph Series B. Academia Sinica, 19-62, 2004.

[58] Johnson, C. R., Petruck, M. R., Baker, C. F., Ellsworth, M., Ruppenhofer, ve Fillmore, C. J., FrameNet: Theory and Practice, 1.1 edn. International Computer Science Institute, 1947, CenterBerkeley, USA Available as: http://framenet.icsi.berkeley.edu/framenet/book/book.html, 2003.

[59] Gruber, T. R., A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, 5(2), 199-220, 1993.

[60] Curran, J. R., Clark, S., ve Bos, J., Linguistically Motivated Large-Scale NLP with C & C and Boxer, Proceedings of the ACL 2007 Demo and Poster Sessions, Association for Computional Linguistics, 33-36, 2007.

[61] Clark, S., ve Curan, J. R., Parsing the WSJ using CCG and log-linear models, In Proceedings of ACL-04, 104-111, Barcelona, Spain, 2004.

[62] Kamp, H., ve Reyle, U., From Discourse to Logic; An Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and DRT, Kluwer, Dordrecht, 1993.

[63] Krajca, P., Outrata, J., ve Vychodil, V., Parallel Recursive Algorithm for FCA, Proceedings of the Sixth International Conference on Concept Lattices and Their Applications, 433, 71-82, Olomouc, CEUR-WS.org, 2008.

[64] Petruck, M. R. L., Frame Semantics, In Verschueren, J., Östman, J. O., Blommaert, J., Bulcaen, C., eds.: Handbook of Pragmatics, John Benjamins, Philadelphia, 1996.

[65] Levin, B., Course LSA 116, LSA Linguistic Institute, University of California, Berkeley, July 2009.

[66] Chomsky, N., Syntactic Structures, The Hague/Paris: Mouton, 13, 1957.

[67] Radford, A., Minimalist Syntax: Exploring the structure of English, Cambridge University Press, Cambridge, 2004.

[68] Cook, W.A., S. J., Case Grammar Theory, Georgetown University Press, Washington, 1989.

[69] Langacker, R. W., Foundations of Cognitive Grammar, Descriptive Application, Stanford: Stanford University Press, Volume 2, 1991.

[70] Eijk, J. V., Computational Semantics with Functional Programming, Cambridge University Press, 2010.

[71] Jones, K. S., Words and Intelligence II: Semantic primitives: the tip of the iceberg, Essays in Honor of Yorick Wilks, Ed. K. Ahmad, C. Brewster and M. Stevenson, Computer Laboratory, University of Cambridge, Text, Speech and Language Technology Volume 36, 235-253, Berlin: Springer, 2007.

[72] Kılıçaslan, Y., ve Tuna, G., An NLP-based approach for improving humon-robot interaction, Journal of Artificial Intelligence and Soft Computing Research, Volume 3, No:3, 189-200, 2013.

[73] Kılıçaslan Y., Uçar, Ö., ve Özkan, Y., FrameNet Tekniği İle İki Dilli Ontoloji Hizalaması, NWSA-ENGINEERING SCIENCES, ISSN:1306-3111, e-Journal of New World Sciences Academy, Volume 7, Number 2, Article Number 1A0321, 2012 http://www.newwsa.com/.

[74] Fırat, Y., Uçar, Ö., ve Kılıçaslan, Y., Semantic Analysis of Turkish Through Formal Ontology Constructed in Computer, Journal of International Scientific Publications: Materials, Methods & Technologies, Volume 7, part 2, ISSN 1313-2539, Bulgaria, 2013, http://www.scientific-publications.net/.

[75] Fırat, Y., Kılıçaslan, Y., ve Uçar, Ö., Bilgisayar Ortamında Biçimsel Ontoloji Oluşturulması, Yüksek Öğretim ve Bilim Dergisi / Journal of Higher Education and Science, Cilt 3, Sayı 1, 080-084, Bülent Ecevit Üniversitesi, 2013, http://www.higheredu-sci.org/content.php3?id=8.

[76] Fırat, Y., Uçar, Ö., ve Kılıçaslan, Y., Semantic Analysis with a Lattice-Based FrameNet, Journal of International Scientific Publications, Language, Individual & Society, ISSN: 1314-7250, Volume 8, Bulgaria, 2014 http://www.scientific-publications.net/en/article/1000618/.

ÖZGEÇMİŞ

08.02.1972 yılında Karaman'da doğdum. İlk, orta ve lise eğitimimi Çanakkale'de tamamladım. 1994 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Öğretmenliğini (İngilizce) bitirdim. Aynı yıl içerisinde Elazığ Anadolu Teknik Lisesine teknik öğretmen olarak atandım. 1995 yılında Çanakkale Onsekiz Mart Üniversitesi'ne Öğretim Görevlisi olarak atandım. 2000 yılında Çanakkale Onsekiz Mart Üniversitesi Sosyal Bilimler Enstitüsüne bağlı İşletme Ana Bilim Dalına ait Yönetim Organizasyonunda yüksek lisansımı bitirdim. 2010 yılında Trakya Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliğinde Doktora Programına başladım. Halen Çanakkale Onsekiz Mart Üniversitesi Çanakkale Teknik Bilimler Meslek Yüksekokulunda Öğretim Görevlisi olarak çalışmaktayım.

TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER

1- Yılmaz KILIÇASLAN, Özlem UÇAR, Yelda ÖZKAN, FrameNet Tekniği İle İki Dilli Ontoloji Hizalaması, NWSA-ENGINEERING SCIENCES, ISSN:1306-3111, e-Journal of New World Sciences Academy 2012, Volume 7, Number 2, Article Number 1A0321 http://www.newwsa.com/

2- Yelda FIRAT, Özlem UÇAR, Yılmaz KILIÇASLAN, Semantic Analysis of Turkish Through Formal Ontology Constructed in Computer, Journal of International Scientific Publications: Materials, Methods & Technologies, Volume 7, part 2, ISSN 1313-2539, 2013, Bulgaria http://www.scientific-publications.net/

3- Yelda FIRAT, Yılmaz KILIÇASLAN, Özlem UÇAR, Bilgisayar Ortamında Biçimsel Ontoloji Oluşturulması, Yüksek Öğretim ve Bilim Dergisi / Journal of Higher Education and Science, Cilt 3, Sayı 1, Sayfa(lar) 080-084, 2013, Bülent Ecevit Üniversitesi http://www.higheredu-sci.org/content.php3?id=8

4- Yelda FIRAT, Özlem UÇAR, Yılmaz KILIÇASLAN, Semantic Analysis with a Lattice-Based FrameNet, Journal of International Scientific Publications, Language, Individual & Society, ISSN: 1314-7250, Volume 8, 2014, Bulgaria http://www.scientific-publications.net/en/article/1000618/