T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SÖZDİZİMSEL AKTARIMA DAYALI MAKİNE ÇEVİRİSİ

TUĞBA ÇALIŞ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Yrd. Doç. Dr. Özlem AYDIN

EDİRNE - 2017

TUĞBA ÇALIŞ'ın hazırladığı "SÖZDİZİMSEL AKTARIMA DAYALI MAKİNE ÇEVİRİSİ" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir Yüksek lisans tezi olarak kabul edilmiştir.

Jüri Üyeleri : İmza

Yrd. Doç. Dr. Özlem AYDIN .....

Doç. Dr. M. Tolga SAKALLI .....

Yrd. Doç. Dr. E. Serdar GÜNER .....

Tez Savunma Tarihi: 22/12/2017

Bu tezin Yüksek Lisans olarak gerekli şartları sağladığını onaylarım.

İmza

Yrd. Doç. Dr. Özlem AYDIN
Tez Danışmanı .....

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

Prof. Dr. Murat YURTCAN
Fen Bilimleri Enstitüsü Müdürü

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI
DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

22 / 12 / 2017

Tuğba Çalış



Yüksek Lisans Tezi
Sözdizimsel Aktarıma Dayalı Makine Çevirisi
T.Ü. Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Makine çevirisi yapay zekâ bilim alanının bir alt dalı olan Doğal Dil İşleme alanının önemli bir dalıdır. Bilgisayımsal teknolojinin ilerlemesiyle ve insanlık tarihinde farklı dillerin gelişimiyle çeviriye olan ihtiyaç artmıştır. Bu tezin amacı, İngilizce'den Türkçe'ye sözdizimsel aktarım yöntemi ile makine çevirisi yapan çekirdek bir uygulama geliştirmektir.

Çalışmada öncelikle İngilizce varoluş cümlelerinin (existential sentences) sözdizim kuralları araştırılmış ve öbek yapı kuralları belirlenmiştir. Ardından, sözdizim kuralları, dilbilim teorem ve hipotezleri ışığında ayrıştırma ağaçlarına aktarılmıştır. Sözcük anlamları için ise bir sözlük oluşturulmuştur. Geliştiricinin oluşturduğu bu sözlük ve dil çiftinin öbek yapı kuralları mantık programlama dili olan Prolog ile kodlanmıştır. Ortaya konan çeviri uygulamasının, kurallı cümle yapılarında sözdizimsel aktarım yöntemi ile başarılı sonuçlar elde ettiği görülmüştür.

Yıl : 2017

Sayfa Sayısı : 46

Anahtar Kelimeler : Makine Çevirisi, Kural Tabanlı Makine Çevirisi, Sözdizimsel Aktarım Yöntemi

Master's Thesis
Machine Translation Based on Syntactic Transfer
Trakya University Institute of Natural Sciences
[Department of Computer Engineering]

ABSTRACT

Machine translation is an essential branch of natural language processing which is a subfiled of artificial intelligence. As computational technology advances and with the development of the different languages in the history of mankind, demand for an effective translation system is increased. The aim of this thesis is to develop a core application that translates machines from English into Turkish by using syntactic transfer method.

In the study, first of all, the syntax rules of English existential sentences are researched and the structure rules of phrases are determined. Then, syntactic rules, linguistic theorems and hypotheses are transmitted to the parse trees. A dictionary is created for the meaning of the words. This dictionary created by the developer and the phrase structure of the language pair are coded in Prolog, which is the logic programming language. It has been seen that the syntactic transfer method in the case of regular sentence structures has achieved successful results.

Year : 2017

Number of Pages : 46

Keywords : Machine Translation, Rule-Based Machine Translation, Syntactic Transfer

TEŞEKKÜR

Bu tezin hazırlanmasında her konuda bana destek olan değerli hocam Yrd. Doç. Dr. Özlem AYDIN'a çok teşekkür ederim. Tezin uygulama kısmında yol gösteren ve desteğini esirgemeyen kıymetli hocam Prof. Dr. Yılmaz KILIÇASLAN 'a çok teşekkür ederim.

İÇİNDEKİLER

ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
İÇİNDEKİLER vii
ÇİZELGELER DİZİNİ ix
ŞEKİLLER DİZİNİ x
BÖLÜM 1 1
GİRİŞ 1
BÖLÜM 2 2
MAKİNE ÇEVİRİSİ 2
2.1 TARİHÇE 2
2.2 MAKİNE ÇEVİRİSİ YAKLAŞIMLARI 4
2.2.1 İstatistiksel Yaklaşım 4
2.2.2 Kural Tabanlı Yaklaşım 5
2.2.2.1 Doğrudan Çeviri Yöntemi 6
2.2.2.2 Aktarım Yöntemi 7
2.2.2.3 Aradil Yöntemi 7
2.3 MAKİNE ÇEVİRİSİNDE KARŞILAŞILAN ZORLUKLAR 8
2.3.1 Biçimbilgisel Düzeyde Zorluklar 8
2.3.2 Sözcük Düzeyinde Karşılaşılan Zorluklar 8
2.3.3 Sözdizim Düzeyinde Karşılaşılan Zorluklar 10
2.3.4 Söylem Düzeyinde Karşılaşılan Zorluklar 10
2.4 MAKİNE ÇEVİRİSİ UYGULAMALARI 11
BÖLÜM 3 12
SÖZDİZİMSEL AKTARIMA DAYALI MAKİNE ÇEVİRİSİ 12
3.1 TÜRKÇE'DE SÖZDİZİM ANALİZİ 12
3.2 KAVRAMLAR 13
3.2.1 Yüklemler ve Özellikleri 13

3.2.2 Fiiler ve Özellikleri 14
3.2.3 Tamamlayıcı Özneler 14
3.3 TEOREMLER VE HİPOTEZLER 15
3.3.1 X-Etiketi Teorisi (X-Bar Theory) 15
3.3.2 Fiil Öbeği İç Öznesi Hipotezi (Verb Phrase Internal Subject Hypothesis) 16
3.3.3 Katmanlı Fiil Öbeği Hipotezi (VP-Shell Hypothesis) 16
3.3.4 Sözdiziminde Küçük "v" (Little v in Sytntax) 17
3.4 UYGULAMA 18
3.4.1 PROLOG UYGULAMASI 24
3.4.1.1 Ötele-İndirge Ayrıştırıcısı 24
3.4.1.2 Birleştirme 33
3.4.1.3 Özellik Yapısı 34
3.4.1.4 Geri İzleme 35
BÖLÜM 4 37
DEĞERLENDİRME VE SONUÇ 37
BÖLÜM 5 39
EKLER 39
KAYNAKLAR 44

ÇİZELGELER DİZİNİ

Çizelge 2.4 Bazı popüler makine çevirisi uygulamaları 11
Çizelge 4.1 Makine çevirisi yöntemlerini karşılaştırma 37

ŞEKİLLER DİZİNİ

Şekil 2.1 Makine Çevirisi Yaklaşımları 4
Şekil 2.2 Vauquois üçgeni. 6
Şekil 3.1 Genel Bir X-Etiketi Yapısı 15
Şekil 3.2 Bir fiil öbeği oluşturma 16
Şekil 3.3 İngilizce cümle yapısı 17
Şekil 3.4 İngilizce oldurgan cümle yapısı 17
Şekil 3.5 Yeni bir çözümleyici oluşturma 19
Şekil 3.6 Sözlükte sözcük bilgisi tanımlama 19
Şekil 3.7 İsim öbeği oluşturma 20
Şekil 3.8 Prolog derleyicisinde isim öbeği oluşturma 21
Şekil 3.9 Fiil öbeği ayrıştırma ağaçları 21
Şekil 3.10 Prolog derleyicisinde fiil öbeği oluşturma 22
Şekil 3.11 Zaman öbeği ayrıştırma ağacı 22
Şekil 3.12 Prolog derleyicisinde cümle oluşturma 23

BÖLÜM 1

GİRİŞ

Dilde çeviri, bir dildeki bir metnin başka bir dile aktarılması işlemidir. Zamanla farklı dillerin oluşmasıyla birlikte, çeviriye gereksinim duyulmaya başlanmıştır. Farklı dili konuşan toplumlar arasında resmi anlaşmalar, ilk çeviri gereksinimlerini oluşturmuştur. Buna ek olarak, dinsel metinlerin başka dillere aktarılma isteği de çeviri gereksinimini artırmıştır. Bir dilde yazılmış olan bir metni diğer bir dilde yazılmış bir metne çevirme işleminde bilgisayarın da kullanılmasıyla makine çevirisi alanı ortaya çıkmıştır. Makine çevirisi yaklaşımları, istatistiksel ve kural tabanlı yaklaşımlar olarak iki ana grup altında toplanmıştır. Son zamanlarda bu iki yöntemi bir arada kullanan hibrit çalışmalar da yapılmaya başlanmıştır. Bu çalışmada, varoluş cümlelerinde (existential sentences) kural tabanlı bir yaklaşım olan sözdizimsel aktarım yöntemi ile yapılan bir makine çevirisi geliştirilmesi hedeflenmektedir.

BÖLÜM 2

MAKİNE ÇEVİRİSİ

Makine çevirisi, çeviri işinin bilgisayar kullanımı yoluyla otomatikleştirilerek yapılması işidir. Çeviri işinin otomatikleştirilmesi sürecinde 2 ana yaklaşım ortaya atılmıştır:

- İstatistiksel makine çevirisi
- Kural tabanlı makine çevirisi

Çeviri kalitesinin iyi olması için, hem kaynak dilin hem de hedef dilin yapısının iyi incelenmesi gerekmektedir. Bu bakımdan kaynak dilin kurallarını hedef dile bir makine aracılığıyla aktarabilme işi farklı güçlükler doğurmaktadır.

2.1 TARİHÇE

Çeviri işinin otomatikleşmesi sürecinde ilk patent, 3 aşamalı modeli ile Petr Petrovich Troyanskii tarafından alınmıştır. Bu modelde ilk olarak, kaynak dildeki ifadelerin bir insan tarafından mantıksal çözümlemesi yapılmıştır. Daha sonra, bu çözümleme makineye girdi olarak verilmiştir. Son olarak da elde edilen sonuç insan tarafından yeniden düzenlenmiştir (Hutchins & Lovtskii, 2000, s. 187– 221).

20. yüzyılda ortaya çıkan sanayi devrimi sonrasında uluslararası ilişkilerin çoğalması çeviri ihtiyacını artırmıştır. Soğuk Savaş döneminde ise çeviri bağımsız bir bilim alanı olmuştur. Georgetown-IBM deneyi gibi bu dönemde yapılan çalışmaların neredeyse tamamı, Rusça – İngilizce dil çifti arasında çevirilere odaklanmıştır (Hutchins,1954).

1966'da ALPAC2 (Automatic Language Processing Advisory Committee) tarafından çeviri sürecini değerlendiren bir rapor yayınlanmıştır (Pierce & Carroll,1966). Raporda aşağıdaki konulara değinilmiştir:

- Kaliteli bir çeviri elde edebilmek için insan gereksinimi olduğu ifade edilmiştir.
- Çeviriye yardımcı araçların geliştirilmesi gerektiği vurgulanmıştır.

- Başarısızlığın uygulamadan kaynaklandığının görülmesi dilbilim çalışmalarını başlatmıştır.

70'li yıllarda, hesaplamalı sistemler dilbilimsel kuramlara dayandırılmıştır. METEO (Chevalier, Dansereau & Poulin, 1978), EUROTRA (King & Perschke, 1982, s. 27-32), SYSTRAN (Toma, 1977, s. 569-581) gibi çeviri sistemleri makine çevirisi alanında bu yıllarda geliştirilen projelerdendir.

80'li yıllarda ise makine çevirisi ticari bir şekil almaya başlamıştır. Örneğin, Trados firması, ticari makine çevirisi yazılımları üretmeye bu dönemde başlamıştır. Donanımlı bilgisayarlar ve İnternet'in ortaya çıkışı ile 90'lı yıllarda makine çevirisi sistemleri kullanım alanları yaygınlaşmıştır.

Günümüzde, Google Translate gibi İnternet üzerinden erişilebilen büyük ticari makine çevirisi uygulamalarının yaygın kullanım alanları bulunmaktadır.

2.2 MAKİNE ÇEVİRİSİ YAKLAŞIMLARI

Makine çevirisi yaklaşımları, Şekil 2.1'deki gibi alt başlıklar halinde toplanabilir.



Şekil 2.1 Makine Çevirisi Yaklaşımları

2.2.1 İstatistiksel Yaklaşım

İstatistiksel Makine Çevirisi (İMÇ), iki dilli metin derlemlerinin incelenmesi yoluyla gerçekleştirilen ve istatistiksel modelleri kullanan bir makine çevirisi yaklaşımıdır.

Bu yaklaşıma ilişkin ilk düşünceler, Claude Shannon'ın bilgi kuramına bağlı kalarak Warren Weaver tarafından 1949 yılında ortaya atılmıştır. Ancak o yıllarda bilgisayarların depolama gereksinimleri sağlanamadığından ve işlem kapasiteleri yeterli olmadığından bu yöntem kullanılamamıştır. Doksanlı yıllarda, güçlü depolama alanı ve bu yöntemi gerçekleştirebilecek kapasitede bilgisayarlara sahip olunmasıyla birlikte bu yöntem IBM araştırma laboratuvarlarında yeniden gündeme gelmiştir. Son yıllarda, oldukça popüler olan bu yöntem, çeviride yapılması gereken işleri oldukça azaltmıştır.

Ancak etkin bir çeviri sonucu elde edebilmek için çok büyük paralel metinlere ihtiyaç vardır.

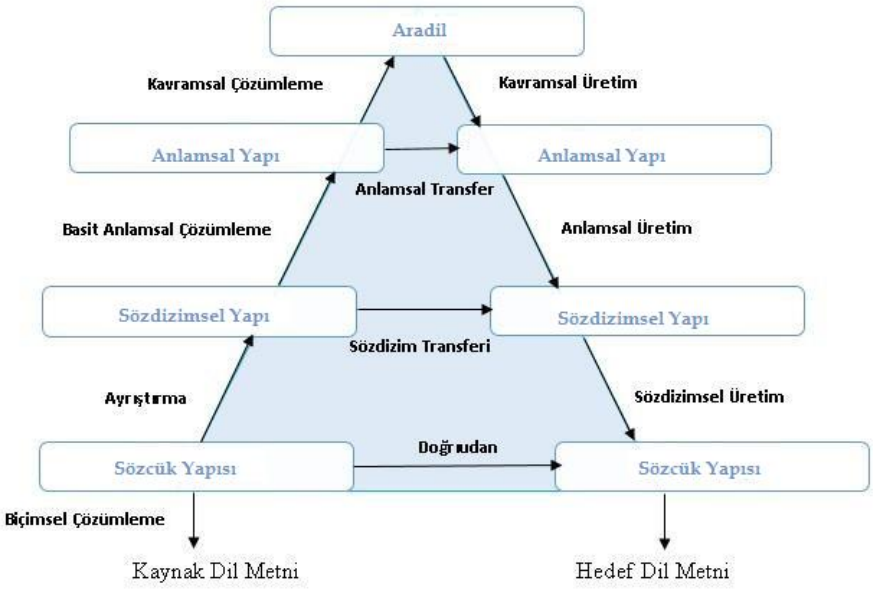
İMÇ, sistem kurulduktan sonra otomatik olarak çalıştığı ve elle kural girme zorunluluğu olmadığı için gerekli iş gücünü oldukça azaltmaktadır. Ancak, etkin çalışabilmesi için çok büyük paralel metinlere ihtiyaç duyulması, bu paralel metinlerin çok az dil çifti için var olması ve alana özgü olmalarından dolayı kısıtlı alanlarda başarım sağlanabilmektedir.

2.2.2 Kural Tabanlı Yaklaşım

Kural tabanlı makine çevirisi, kaynak dildeki sözcükler ve dilbilgisi kurallarından alınan dilsel bilgileri, hedef dildeki sözcüklere ve dilbilgisi kurallarına aktarma işidir. Bu yaklaşım, 3 ana başlık altında incelenebilir:

- Doğrudan çeviri yöntemi
- Aktarım yöntemi
- Aradil yöntemi

Kaynak dilden hedef dile çeviri yapılması sürecinin şematik olarak gösterimi Şekil 2.2'de verilen Vauquois Üçgeni ile yapılır. Kaynak dilden hedef dile geçiş yapılabilecek aşamalar üçgen üzerinde gösterilir. Doğrudan çeviri aşamasından aradil aşamasına ilerlendikçe yapılması gereken çözümleme miktarı artar, ihtiyaç duyulan bilgi azalır.



Şekil 2.2 Vauquois üçgeni.

2.2.2.1 Doğrudan Çeviri Yöntemi

Kaynak dildeki metni sözcük sözcük inceleyen en basit çeviri düzeyidir. Sözcüklerin hedef dildeki karşılıklarını bulmak için çift dilli bir sözlükten faydalanılır.

Bu düzeyde herhangi bir dilbilgisel işleme veya kural analizi yapmaya ihtiyaç duyulmaz. Çözümleme yapmadan kaynak dildeki her sözcük hedef dile birebir çevrilir. Bu yüzden sözlük kapasitesi yeterince fazla olmalıdır.

Hedef dildeki sözcük ile kaynak dildeki sözcük arasında basit bir yer değiştirmeye başvurularak çeviri elde edilebilir. Örneğin İngilizce'de genel olarak özne-yüklem-nesne şeklinde sıralı bir yapı varken, bu yapıya karşılık Türkçe'de özne-nesne-yüklem gibi bir yapı mevcuttur. Bu yöntemi örneklendirmek gerekirse:

İngilizce → Ayşe loves cats.

Türkçe → Ayşe kedileri sever.

Bu yöntem her sözcüğün her iki dilde birebir karşılığının olduğu yapılarda oldukça başarılıdır.

Doğrudan çeviri yönteminin avantajları şunlardır:

- Hızlıdır.
- Kolay ve basit uygulanabilir.
- Düşük maliyetlidir.

Doğrudan çeviri yönteminin dezavantajları şunlardır:

- Çeviri yapılacak iki dil arasında yapısal farklılıkların bulunması başarılı sonuçlar elde edilememesine neden olur.
- Büyük öbekler içeren ifadelerde ve uzak yer değiştirmelerde yetersiz kalır.

2.2.2.2 Aktarım Yöntemi

Etkin bir otomatik makine çevirisi yapılabilmesi için kaynak dil yapısının hedef dil yapısına uygun aktarım yapılması gerekir. Sözcük-sözcük çeviri yapan doğrudan çeviri yöntemine ek olarak öbek bilgilerinin ve yapısal bilgilerin eklenmesiyle daha başarılı sonuçlar elde edilmiştir.

Kaynak dil ifadesi biçimsel bir kurama dayalı olarak çözümlenir. Çözümleme işleminin makineye aktarılması hedef dilin dilbilgisi kurallarıyla gerçekleşmelidir. Çözümleme aşamasında elde edilen yapının türü aktarımın nasıl yapılacağını da belirler.

2.2.2.3 Aradil Yöntemi

Analizde en üst düzey olan *aradil (Interlingua)* yöntemi ile kaynak dilden bir ara gösterime anlamsal çözümleme yapılarak soyut bir gösterim elde edilir. Hedef dil ifadesi bu gösterimden faydalanarak oluşturulur.

Kaynak dil ifadesi gerekli çözümlemeler yapılarak dile bağlı kalmaksızın oluşturulabilirse anlamsal bilgiye ihtiyaç duyulmaz. Hedef dil de çözümlenen bu soyut ifadeden elde edilebilir. Dilden bağımsız oluşturulan bu soyut ifade *aradil* olarak adlandırılır.

Ara gösterim yüklem mantığı veya durum anlam bilgisi (situation semantics) yaklaşımlarıyla oluşturulabilir.

2.3 MAKİNE ÇEVİRİSİNDE KARŞILAŞILAN ZORLUKLAR

Bir dildeki bir metin başka bir dile aktarılırken kaynak metnin tamamı anlaşılmalıdır. Bunun için de çeviri yapılan dile ve o dilin unsurları olan dilbilgisi, sözdizimi, deyimler, anlambilim ve kültürel çerçeveye tam anlamıyla hâkim olunmalıdır. Bu sebeple de, hedef dilin tüm kuralları bilinmelidir. Çeviri yapılırken kaynak metnin üslup, tonlama, hedef kitle vb. özellikleri dikkate alınmalıdır. Bu bağlamda, çeviri oldukça karmaşık bir süreçtir.

Çeviri kalitesinin iyi olması için, kaynak dilin yapısı, hedef dilin yapısı ve diller arasındaki farklılıklar detaylı bir şekilde incelenmelidir.

Kaynak dilin kurallarını, hedef dile bir makine aracılığıyla aktarabilmek ise farklı güçlükler doğurmaktadır. Bu güçlükler, aşağıdaki başlıklar altında toplanabilir:

- Biçimbilgisel (morphological) düzeyde zorluklar
- Sözcük (lexical) düzeyinde zorluklar
- Sözdizim (syntax) düzeyinde zorluklar
- Söylem (discourse) düzeyinde zorluklar

2.3.1 Biçimbilgisel Düzeyde Zorluklar

Biçimbilgisi (morfoloji), sözcüklerin içyapısının çözümlemesini inceleyen bir bilim dalıdır. Bir dilin anlam taşıyan en küçük parçalarını araştırır. Bir dilin anlam taşıyan en küçük birimine biçimbirim (morfem) denir. Biçimbirimler, bağımlı ve bağımsız biçimbirimler olarak 2 ana başlık altında toplanırlar. Bağımsız biçimbirimler tek başlarına kullanılabilirler. Bağımlı biçimbirimler ise sadece bağımsız biçimbirimlerle birlikte kullanılabilirler. Örneğin, ‘köpek’ bir bağımsız biçimbirimdir. Ama ‘köpekler’ sözcüğündeki ‘-ler’ eki bir bağımlı biçimbirimdir ve yalnız kullanılamaz.

Türkçe, sondan eklemeli bir dil olması nedeniyle biçimbilgisel bakımdan oldukça zengin bir dildir. Dolayısı ile Türkçe ile ilgili yapılan herhangi bir çeviri çalışmasında, biçimbirim kurallarının titizlikle uygulanması gereklidir.

2.3.2 Sözcük Düzeyinde Karşılaşılan Zorluklar

Çeviri yapılan dillerin sözcükleri arasında meydana gelebilecek birtakım düzey farklılıkları, çeviri kalitesini olumsuz yönde etkileyebilmektedir. Dolayısı ile çevirinin kalitesini artırmak için bu tür düzey farklılıklarının mümkün olduğunca azaltılması

gerekmektedir. Örneğin, kaynak dildeki bir sözcüğün hedef dildeki karşılığı birden fazla olabilmektedir. Bunun için çeviriye konu olan sözcüklerin anlamlarının belirginleştirilmesi gerekmektedir.

Sözcük Anlamı Belirginleştirme (SAB) yöntemiyle birden fazla anlamı olan bir sözcüğün bulunduğu bağlamdaki anlamı belirlenir. Bu sorunun daha iyi analiz edilebilmesi için örneklendirmek gerekirse, İngilizce ad kategorisindeki "iron" sözcüğünün birbirinden farklı Türkçe anlamları vardır. Bu anlamlar şöyledir:

- Demir
- Ütü
- Golf sopası
- Zincir
- Güç

Örneğin, İngilizce 'de *"see"* yüklemi "*görmek"* anlamında kullanabileceği gibi "*anlamak"* anlamında da kullanılabilir. Oysa Türkçe'de bu durumu ifade eden iki ayrı fiil kullanılmaktadır.

** İngilizce → I saw a dog in the garden.

Türkçe → Bahçede bir köpek gördüm.

** İngilizce → I see what you mean.

Türkçe → Ne demek istediğini anlıyorum.

İngilizcede cinsiyet içeren sözcükler Türkçe'de tek bir sözcüğe karşılık gelebilmektedir.

Türkçe → *Onun* ne yaptığını bilmiyorum.

İngilizce → I don't know what *he/she* did.

Türkçe → *O* İngilizce konuşuyor.

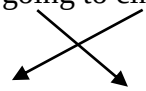
İngilizce → *He/she* speaks English.

Kaynak dilde mevcut bir sözcüğün hedef dilde bir karşılığı olmayabilir, bu durumda sözcüksel boşluk (lexical gap) oluşur. Türkçe 'de "*gümüşservi*" sözcüğü (TDK Türkçe Sözlük, 2015) ayın suya yansımasıyla oluşan parıltılı görünüm manasına gelen bir ifadedir ve İngilizce 'de sözcük karşılığı bulunmamaktadır.

2.3.3 Sözdizim Düzeyinde Karşılaşılan Zorluklar

Diller sözdizimsel olarak özne, yüklem, nesne ve tümleç gibi yapılardan oluşur. Bu yapıların cümle içindeki konumları dilden dile değişebilmektedir. Bu zorluğu örneklendirmek gerekirse:

İngilizce → Ayşe is going to cinema.



Türkçe → Ayşe sinemaya gidiyor.

İngilizce cümlede öge dizilişi özne (Ayşe), yüklem (is going), tümleç (to cinema) şeklindedir. Çeviri cümlede ise, öge dizilişi özne tümleç yüklem şeklindedir. Cümle içindeki bu konum değişiklikleri çeviri yapılırken dikkate alınmalıdır.

2.3.4 Söylem Düzeyinde Karşılaşılan Zorluklar

Çeviride sözcüklerin doğru yerde kullanılmamasından kaynaklı zorluk yaşanabilir. Aşağıdaki cümle bu zorluğu örneklemektedir.

Türkçe → Hırsız çocuğu kovaladı.

İngilizce → The thief chased the child.

→ Someone chased the thief child.

"*The thief chased the child*" cümlesi, kovalayanın hırsız olduğu durumdur. *"Someone chased the thief child"* cümlesi ise birisinin "hırsız çocuğu" kovaladığı durumu ifade etmektedir.

2.4 MAKİNE ÇEVİRİSİ UYGULAMALARI

Bazı popüler makine çevirisi uygulamalarının üzerinde çalıştıkları platformlar ve geliştirilme yaklaşımları incelenerek aşağıdaki Çizelge 2.4'te oluşturulmuştur (Wikipedia, 2015).

Çizelge 2.4 Bazı popüler makine çevirisi uygulamaları

İSİM	PLATFORM	AÇIKLAMA
Xerox Easy Translator	Web Uygulaması	Hibrit makine çevirisidir. Son düzenlemeler bir insan aracılığıyla yapılmıştır.
Asia Online	Windows, Linux, Web	Hibrit makine çevirisidir. Bir sunucu platformu ile çeviriye yardımcı masaüstü araçlar sunmuştur.
Apertium	Debian	Kural tabanlıdır. Dil ile ilgili bilgiler açık kaynaklıdır.
Anusaaraka	Unix uyumlu	Kural tabanlıdır. Derin ayrıştırıcı tabanlıdır (deep parser based). Dil ile ilgili bilgiler açık kaynaklıdır.
IBM	Platform bağımsız	IBM tarafından geliştirilen hibrit makine çevirisidir. Sadece kural tabanlı halinin WebSphere Çeviri Sunucusu'nda bir ürünü mevcuttur.
Google Translate	Web uygulaması	İstatistiksel makine ve sinirsel makine çevirisidir.
Bing Translator	Web uygulaması	Microsoft'un dilbilim kurallarına uygun olan istatistiksel makine çevirisini kullanır.
SAIC Omnifluent	Windows, Linux, Web, i0S	Hibrit makine çevirisidir. Otomatik konuşma tanıma ile tek bir platform üzerinde metin ve ses dosyalarını çevirebilmektedir.
SYSTRAN	Platform bağımsız	Hibrit makine çevirisidir.

Yukarıda bahsedilen popüler çeviri uygulamalarından İngilizce'den Türkçe'ye makine çevirisi yapabilenler şunlardır:

- SYSTRAN
- Bing Translator
- Google Translate

BÖLÜM 3

SÖZDİZİMSEL AKTARIMA DAYALI MAKİNE ÇEVİRİSİ

Sözdizim (syntax), doğal dillerde cümle kurma prensiplerini inceleyen bir dilbilimi dalıdır. Bir dilin sözdizimi, her sözcüğe bir görev atayan ve bu sözcükler arasında ilişki kuran prensiplerden oluşur. Sözdizimsel analiz aracılığıyla doğal dillerde cümleyi oluşturan kurallar çözümlendikten sonra bu kuralların makineye aktarılması, sözdizimsel aktarıma dayalı makine çevirisi işinin bir parçası olmuştur.

Sözdizimsel aktarım, sözdizimi veya cümleyi oluşturan morfolojik ögelerin hiyerarşik kurallara uyumunu karşılaştırarak ölçümlemektir (Wikipedia, 2015).

Bu çalışmada, sözdizimsel aktarım ile makine çevirisi hedeflenmiştir. Bu yöntemle, hedef dilin sözcüksel ve sözdizimsel ifadeleri kaynak dilden ayrıştırılır. Daha sonra, dönüşüm kuralları ağaç veri yapısı ile hedef dile aktarılır. Çift dilli bir sözlük kullanılarak aktarım gerçekleşir.

3.1 TÜRKÇE'DE SÖZDİZİM ANALİZİ

Türkçe'de cümleler en genel şekliyle özne, nesne ve yüklem bileşenlerinden oluşur. Buna ek olarak, cümle anlamı kuvvetlendirilmek istenirse tümleç, bağlaç, edat gibi bileşenler de eklenebilir. Türkçe'de özne ile yüklem temel bileşenler olup genelde tüm cümlelerde yer alırlar.

Sözdizimsel analizde amaç, bir giriş cümlesini alarak bu cümleyi temsil eden ayrıştırma ağaçlarını oluşturmaktır (Erenler, 2005).

Sözdizimsel analizin yapılabilmesi için leksikona, dilbilgisi kurallarına ve ayrıştırma algoritmasına gerek vardır. Leksikon, sözcüklerin köklerini ve sözcük türü (part of speech) bilgilerini içeren sözlük manasına gelmektedir. Bir dildeki cümle türleri ve cümledeki ögelerin diziliş sırası kurallarını içeren formel tanımlamalar dilbilgisi kurallarıdır. Ayrıştırma algoritmaları, giriş cümlesini alarak dilbilgisi kurallarına uygun biçimde ayrıştırma ağaçlarını oluşturma amacını taşımaktadır.

3.2 KAVRAMLAR

Sözdizimsel aktarıma dayalı çalışmada kullanılan kavramlar, yüklemler ve özellikleri, fiiller ve özellikleri, tamamlayıcı özneler (dummy subjects) başlıklarıyla açıklanmıştır.

3.2.1 Yüklemler ve Özellikleri

Yüklem, cümlede özne tarafından gerçekleştirilen eylemi, oluşu, hareketi veya bir olguyu ya da yargıyı bildiren sözcük veya sözcük grubudur. Özne ile birlikte cümlenin ana öğelerinden biridir.

Greg Carlson, İngilizce üzerine yaptığı çalışmalarında, yüklemlerin öz nitelikler (individual-level) ile geçici nitelikler (stage-level) taşıdığını savunmuştur. Bu özelliklerin arasındaki ayrımın iyi yapılması gerektiğini vurgulamıştır. Geçici nitelikler, zaman dilimi içinde var olurlar. Yani zaman kavramıyla varlığı meydana gelen özelliklerdir. Öz nitelikler, bireylerin sahip olduğu özellikleri ifade ederken, geçici nitelikler bireylerin özelliklerinin anlık sahnelenmesini ifade etmektedir. Örneğin, kapının ardındaki bir köpeğin havlaması (1a) geçici nitelikli yüklemdir (Stage Level Predicate - SLP). Köpeğin kuyruğunun olması (1b) ise öz nitelikli yüklemdir (Individual Level Predicate - ILP) (Carlson, 1977).

(1a) There is a dog barking (SLP)

(1b) The dog has a tail. (ILP)

Bu ayrımın birçok dilde dilbilgisel etkisi vardır. Örneğin, İspanyolca'da "to be" fiilinin 2 farklı kullanımı vardır. "Ser" öz nitelikli bir yüklemken (2a), "estár" geçici nitelikli bir yüklemdir (2b).

(2a) La casa es roja (ILP)

Bu cümle "Ev kırmızıdır" manasına gelmektedir. Öz nitelikli yüklemlere bir örnektir.

(2b) Yo estoy en Taksim. (SLP)

Bu cümle “Ben Taksim’deyim” manasına gelmektedir. Geçici nitelikli yüklemlere bir örnektir.

İngilizce’de öz nitelikli yüklemler genel olarak isim fiil yapan “–ing” (gerund) ekini almazlar. (3a)’daki durum buna bir örnektir. (3b)’de ise geçici nitelikli bir yüklem olan “bark” sözcüğü için örnek bir cümle verilmiştir.

(3a) I know English. (ILP)
(3b) The dog is barking. (SLP)

Yüklemlerin öz nitelikli ya da geçici nitelikli olma durumu varoluş cümlelerinin yapısını etkilemiştir.

3.2.2 Fiiler ve Özellikleri

Fiiller bir iş, bir oluş, bir hareket bildiren sözcüklerdir. Nesnesine göre fiiller, geçişli (transitive) ve geçişsiz (intransitive) olmak üzere 2 ana gruba ayrılmaktadır. Geçişli fiiler nesne alan fiillerdir. Cümlede ne, neyi, kimi sorularının cevabı varsa o cümlede geçişli fiil var demektir. Geçişsiz fiiller ise, nesne almayan fiillerdir. Cümlede ne, neyi, kimi sorularının cevabı bulunmuyorsa o cümlede geçişsiz fiil var demektir. Örneğin, (4a) durumunda geçişli fiil, (4b) durumunda geçişsiz fiil örneklenmiştir.

(4a) O’na çiçek aldım.
(4b) Akşam sinemaya gidiyorum.

Fiilerin nesne alıp alamaması durumu, sözdizimsel aktarım kurallarını oluşturan yapıların başında gelmektedir.

3.2.3 Tamamlayıcı Özneler

İngilizce’de bir cümlede fiille ilişkili bir özne bulunmadığı durumlarda tamamlayıcı (dummy) özneye ihtiyaç duyulur. Aslında gerçek özne cümleciğin (clause) herhangi bir yerindedir. “There” ve “it” özneleri, İngilizce’deki tamamlayıcı öznelerdendir.

Örneğin, “There is a cat chasing a dog” cümlesindeki gerçek özne bir kedidir (a cat) ve kedi köpek kovalıyordur (chasing a dog). “There”, bu cümlede boş özne olarak kullanılmıştır.

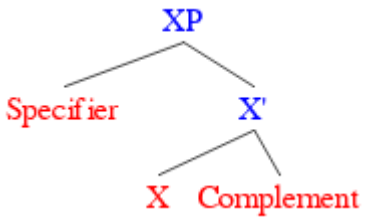
3.3 TEOREMLER VE HİPOTEZLER

Sözdizimsel aktarım yönteminin makineye aktarılmasında çeşitli teoremlerden ve hipotezlerden faydalanılmaktadır.

3.3.1 X-Etiketi Teorisi (X-Bar Theory)

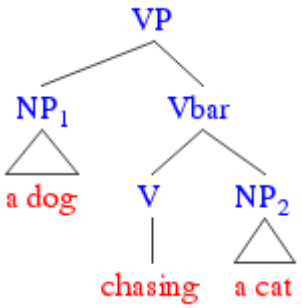
Tüm dillerde ortak olan özellikleri belirlemeye çalışan X-Etiketi Teorisi (XET) dilbilim teorisinin bir parçasıdır. Tüm dillerin aynı temel sözdizimsel yapıları veya benzer dilbilgisel yapıları paylaştığını varsayar (Chomsky, 1970, s. 184-221). Standart dilbilgisi kuralları bu teoremin tekrar düzenlenmiş şeklidir.

XET, sözdizimsel kategori oluşumunun bir teorisidir. Bu teoriye göre her X öbeği (X phrase) , bir belirtece (specifier) ve X’in bir yansımasına (projection) sahip olmalıdır. X yansıması ise X’in kendisi ve onun tamamlayıcısından (complement) oluşur. X harfi sözcük kategorisi belirtmek için kullanılır. “X” olarak isimlendirilen etiket ismin, fiilin, sıfatın veya edatın yerini tutar. X-Etiketi bu yapının gösteriminden türemiştir. Şekil 3.1’de genel bir X-Etiketi yapısı örneklenmiştir.



Şekil 3.1 Genel Bir X-Etiketi Yapısı.

XET cümleyi bileşenlerine ayırma metodudur. Sözdizimsel analizde, bileşenler bir sözcükten ya da hiyerarşik yapıda tek bir işlevi olan sözcük grubundan oluşurlar. Bu durum, bir fiil öbeği ayrıştırma ağacıyla Şekil 3.2’de örneklenmiştir.



Şekil 3.2 Bir fiil öbeği oluşturma

Bu ağaç yapısına göre bir fiil öbeği (VP), isim öbeği belirteci ve fiil öbeği yansımasından (Vbar) oluşmuştur. Fiil öbeği yansıması, bir fiil ve isim öbeği ile beraber oluşmuştur. İsim öbeği bir artikel (a) ve bir ismin (cat) bir araya gelmesiyle oluşmuştur.

3.3.2 Fiil Öbeği İç Öznesi Hipotezi (Verb Phrase Internal Subject Hypothesis)

XET'ye göre, özneler fiil öbeği belirtecinde yer alırlar. Ancak Fiil Öbeği İç Öznesi Hipotezi (FÖİÇH), var oluş cümlelerinde öznelerin zaman öbeğinin (TP) belirteci pozisyonuna yükselmesiyle aynı durumu ifade ettiğini iddia eder (Radford, 1988). Bu durum, aşağıdaki örnek cümlede açıklanmıştır.

There is a dog chasing a cat.

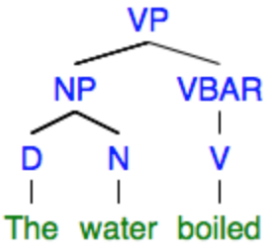
Bu cümlede "there is" yapısı cümlenin belirtecidir. Aynı zamanda boş öznedir, dolayısıyla anlamsızdır. Asıl özne "a dog" ifadesidir. Dolayısıyla " a dog" zaman öbeğinin belirtecine yükselir. Yani, "a dog" öznesi hem fiil öbeği belirteci (spec- VP), hem de zaman öbeği belirteci (spec-TP) olabilir. Bu yüzden, "There is a cat chasing a dog" cümlesi ile "A cat is chasing a dog" cümlesi aslında anlam olarak aynıdır.

3.3.3 Katmanlı Fiil Öbeği Hipotezi (VP-Shell Hypothesis)

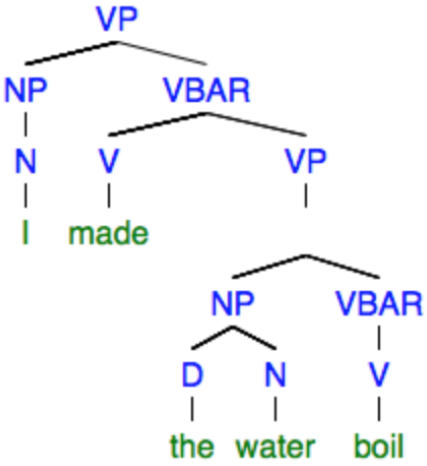
Katmanlı Fiil Öbeği Hipotezi (KFÖH) , fiilin iki nesne arasında bulunduğu durumda ayrıştırma ağacında yukarı doğru hareket ettiği yeni bir yapıyı ortaya çıkarmıştır. Bu hipotez, oldurgan fiil (ergatives) bulunan cümle yapılarını daha iyi analiz edebilmek için geliştirilmiştir. Türkçe'de geçişsiz bir fiile yapım ekleri getirilerek elde edilen geçişli fiiller oldurgan fiillerdir. İngilizce'de bu yapı KFÖH ile açıklanır.

"The water boiled." cümlesi "Su kaynadı" şeklinde Türkçe'ye çevirilir.

"I made the water boil" cümlesi ise oldurgan yapılıdır ve "Suyu kaynattım" şeklinde Türkçe'ye çevrilir. Bu yapıyı açıklayan durum Şekil 3.3'te ve Şekil 3.4'te ağaç yapıları ile verilmektedir.



Şekil 3.3 İngilizce cümle yapısı



Şekil 3.4 İngilizce oldurgan cümle yapısı

3.3.4 Sözdiziminde Küçük "v" (Little v in Sytntax)

Küçük "v" (little v) sözdizimi tarihinde en çok tartışılan başlıklardan biridir. "v" kafası (the v head) ilk olarak fiilin harici argümanı barındıran kafası olarak Chomsky (Chomsky, 1995) tarafından ve daha sonra da Kratzer (Kratzer, 1996, s.109-137) tarafından önerilmiştir. Kısa süre sonra, v'nin 'fiil yapıcı' (verbalizer) yani bir kökü bir fiile dönüştüren kafa olduğu varsayılmıştır. Örneğin "There is a dog chasing a cat" cümlesinde asıl fiil "chasing" fiilidir. Bu başlığa göre "chasing a cat" ifadesi fiile dönüşen kafa olarak ifade edilir. Bu yüzden, küçük "v" başlığı ile asıl fiilin "to be" fiili olmadığı vurgulanmıştır. KFÖH'ye göre geçişli olmayan fiiller geçişli hale getirilirken bu konudan yararlanılmıştır. Küçük "v" öbekleri (LVP) oluşturularak geçişli yapılan geçişsiz fiiller katmanlı bir yapıyla ayrıştırma ağacını oluşturmuştur.

3.4 UYGULAMA

Sözdizimsel aktarım yöntemi uygulanan bu çalışmada, mantıksal programlama dili olan Prolog kullanılmıştır. Prolog dilinde yazılan kod satır satır yorumlanır ve sonuç üretilir. Prolog'ta veritabanı çok önemlidir çünkü sonuçlar veritabanındaki verilerden elde edilir. Veritabanında kurallar belirlenir. Tanımlanan bir kuralın bir diğer kural ile ilişkisi belirlenerek program akışı düzenlenir. Sorgu yapıldığı zaman, kurallara uygun olan sonuç kullanıcıya gönderilir. Prolog'ta bu işlemleri yerine getirmek için Karar Motoru (Inference Engine) denilen bir yapı mevcuttur.

Prolog'ta kodlar yazılırken önce nesneler ve bu nesneler arasındaki ilişkiler belirlenir. Bu nesneler kullanılarak yüklem mantığı yapısı oluşturulur. Bu nesneler gelen sorgu arasından seçilir ve geriye kalan sorgu cümlesi olduğu gibi bırakılır. Böylece hem hızlı hem de amaca uygun sonuçlar elde edilmiş olur.

Prolog derleyicisi, Prolog yüklemlerini bir ağaç yapısı oluşturarak yukarıdan aşağıya doğru yorumlar. Bu yapının dezavantajları şunlardır:

* Kimi zaman sonsuz döngüye sokabilmektedir.

* Eksik veya yanlış çeviriye sebep olabilmektedir.

Bu problemleri en aza indirmek için Şekil 3.5'teki gibi yeni bir ayrıştırıcı (parser) tanımlanmıştır.

```
parse(Expression,Str):- parse1(Expression,R),
 list_to_string(R,Str).
parse1(Expression,R):-
tokenize(Expression,L),
sr_parse([],L,R).
sr_parse([phrase(loc:(cat:_,trans:Trans,pred:_),nonloc:_)],[],Trans).
sr_parse([Y,X|Rest],Words,R):-
 (Z ==> (X,Y)),

sr_parse([Z|Rest],Words,R).
sr_parse([X|Rest],Words,R):-
(Y ==> X),
sr_parse([Y|Rest],Words,R).
```

```
sr_parse(Stack,[Word|Words],R):-
(Cat => [Word]),
sr_parse([Cat|Stack],Words,R).
```

Şekil 3.5 Yeni bir çözümleyici oluşturma

Sözlükte her sözcüğün sözcük türü *cat* adlı değişkene özellik bilgisi olarak tanımlanmıştır. *Cat* değeri isim, fiil, sıfat, zarf ve İngilizce 'deki artikel bilgisi içerebilir. Bu sözcük türü bilgilerine göre cat değişkeni aşağıdaki değerleri alabilir:

Cat : n → Sözcük türünün isim olduğunu ifade eder.

Cat : v → Sözcük türünün fiil olduğunu ifade eder.

Cat : d → Sözcüğün artikel olduğunu ifade eder.

Cat : adv → Sözcük türünün zarf olduğunu ifade eder.

Cat : adj → Sözcük türünün sıfat olduğunu ifade eder.

Sözlükteki her sözcük *word* yüklemi ile Şekil 3.6 'daki gibi tanımlanmıştır. *word* yüklemi ile sözcüklerin çeviri karşıklıları, yüklem özellikleri ve sözcük türleri bilgileri elde edilmiştir. Her İngilizce sözcüğün Türkçe karşılığı *trans* özelliği ile belirlenmiştir. Bu özellik, örneğin "dog" sözcüğünün çeviride karşılığının "köpek" sözcüğü olduğunu ifade etmek için kullanılmıştır. "dog" sözcüğünün sözcük türü isim olduğu için *cat* özelliği isim ifade eden "n" ile belirlenmiştir. *loc* özelliği ise adın kalma durumu (locative) ifadesi için kullanılmıştır. Kalma durumunun aksinin ifadesi için ise *nonloc* özelliği kullanılmıştır. *pred* özelliği ise yüklem türünü belirtmek için oluşturulmuştur. Bu özelliğin "individual" olması bu yüklemin öz nitelikli olmasını ifade ederken, "stage" olması bu yüklemin geçici nitelikli olmasını ifade etmektedir. s*ubj* özelliği ise öznenin yüklemin türüne göre hangi değişkeni alacağını ifade etmektedir.

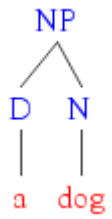
```
word(loc:(cat:n,trans:[köpek],pred:individual)) => [dog].
word(loc:(cat:n,trans:[taş],pred:individual)) => [stone].
word(loc:(cat:n,trans:[kedi],pred:individual))=> [cat].
```

Şekil 3.6 Sözlükte sözcük bilgisi tanımlama

Sözlük oluşturulduktan sonra, İngilizce cümlenin Türkçe karşılığını oluşturan öbek yapıları dilbilgisi kuralları ile programda kodlanır.

Sözlükteki sözcükler birleşerek bir öbek (phrase) yapısı oluşturabilir. İngilizce dilbilgisi kuralları ışığında, bu öbek yapıları isim öbeği (noun phrase), fiil öbeği (verb phrase) şeklinde oluşabilir.

Bir isim öbeği yapısı (NP), sıfat (adj) ile isimden (N) ya da artikel (D) ile isimden oluşabilir. Artikeller, İngilizce'de isimleri genelleştiren ve "herhangi bir" anlamı veren sözcüklerdir. Bu ekler İngilizce dilbilgisinde "article" diye adlandırılır. Türkçe dilbilgisinde bu yapının karşılığı yoktur. İsim öbeği sadece zamirden de (PN) doluşabilir. Şekil 3.7 'deki örnek çeviri cümlesindeki "a dog" yapısı isim öbeğidir.



Şekil 3.7 İsim öbeği oluşturma

Sözcük türü artikel olan ve trans özelliği *Trans1* olan bir sözcük ile sözcük türü isim olan ve trans özelliği *Trans2* olan bir sözcüğün Şekil 3.8 'deki gibi birleşmesiyle bir isim öbeği oluşturulmuştur. Bu isim öbeğinin trans özelliği ise *Trans1* ve *Trans2*'nin birleşmesiyle *Trans* olarak oluşturulup öbek seviyesine trans bilgisi taşınmıştır.

phrase yüklemi, sözcük türü ya da öbek türü, trans özelliği ve yüklem türü bilgilerini içermektedir. Aşağıdaki Prolog kod parçasında iki sözcüğün bir araya gelmesiyle bir isim öbeği oluşturulması örneklenmiştir. Bir artikel ile bir isim, bir sıfat ile bir isim bir araya gelerek isim öbekleri oluşturulmuştur. Bir isim öbeği, bir artikel ve bir isim öbeğinden de oluşabilmektedir.

```
phrase(loc:(cat:np,trans:Trans,pred:Pred)) ==>
word(loc:(cat:d,trans:Trans1)),
word(loc:(cat:n,trans:Trans2,pred:Pred)):-
append(Trans1,Trans2,Trans).
```

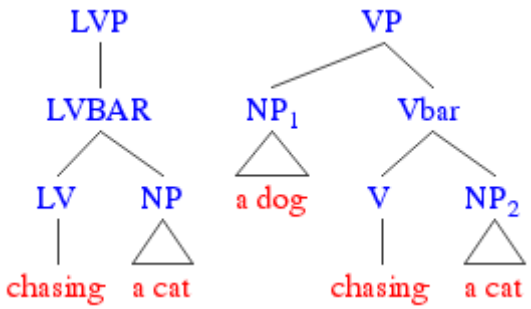
```
phrase(loc:(cat:np,trans:Trans,pred:Pred)) ==>
word(loc:(cat:adj,trans:Trans1,pred:ind)),
word(loc:(cat:n,trans:Trans2,pred:Pred)):-
append(Trans1,Trans2,Trans).
phrase(loc:(cat:np,trans:Trans,pred:Pred)) ==>
word(loc:(cat:d,trans:Trans1)),
phrase(loc:(cat:np,trans:Trans2,pred:Pred)):-
append(Trans1,Trans2,Trans).
```

Şekil 3.8 Prolog derleyicisinde isim öbeği oluşturma

Bir fiil öbeği yapısı sadece fiilden, fiil ile zarftan, fiil ile isimden ya da fiil ile isim öbeğinden oluşabilir. FÖİH'ye göre ve KFÖH'ye göre oluşturulan ayrıştırma ağaçları Şekil 3.9'da verilmiştir.

Uygulamaya göre iki şekilde fiil öbeği oluşturulabilir:

1. Bir isim öbeği (NP) ve küçük "v" (v) ile fiilleşen yapı küçük "v" etiketine (LVBAR) dönüşmüştür. Küçük "v" etiketi de küçük "v" öbeğine (LVP) dönüşmüştür.
2. Bir isim öbeği ve fiil birleşerek (V) "v" etiketini (VBAR) oluşturmuştur. V etiketi (Vbar) ile yine bir isim öbeği birleşerek fiil öbeği (VP) oluşturmuştur.



Şekil 3.9 Fiil öbeği ayrıştırma ağaçları

Sözcük türü fiil olan ve trans özelliği *Trans1* olan bir sözcük ile trans özelliği *Trans2* olan bir isim öbeğinin Şekil 3.10'daki gibi birleşmesiyle bir fiil öbeği oluşturulmuştur. Bu fiil öbeğinin trans özelliği ise *Trans1* ve *Trans2*'nin birleşmesiyle

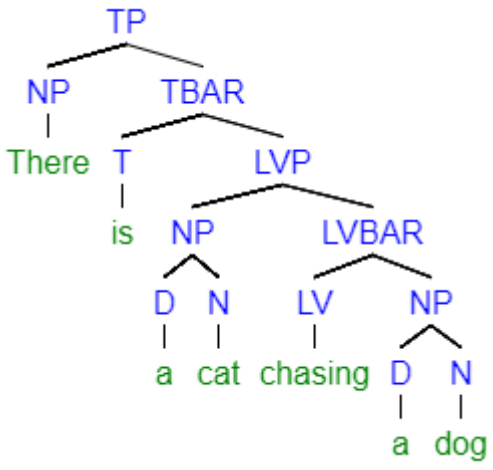
Trans olarak oluşturulup öbek seviyesine trans bilgisi taşınmıştır. Trans özelliği *Trans1* olan isim öbeği ile trans özelliği *Trans2* olan fiil öbeği bir cümle oluşabilir.

```
phrase(loc:(cat:vp,trans:Trans,pred:_),nonloc:subj:stage) ==>
phrase(loc:(cat:np,trans:Trans1,pred:_)),
phrase(loc:(cat:vbar,trans:Trans2,pred:_)):-
append(Trans1,Trans2,Trans).
phrase(loc:(cat:vp,trans:Trans,pred:_),nonloc:subj:individual) ==>
phrase(loc:(cat:vbar,trans:Trans,pred:_)).
```

Şekil 3.10 Prolog derleyicisinde fiil öbeği oluşturma

İngilizce varoluş cümle yapısının zaman öbeği ayrıştırma ağacı aşağıdaki Şekil 3.11'de verilmiştir. Ağaç, Küçük "v", Fiil Öbeği İç Öznesi Hipotezi, Katmanlı Fiil Öbeği Hipotezi ve diğer dilbilim kavramları ve teorileri ışığında oluşturulmuştur.

Küçük "v" bir kökü bir fiile dönüştüren kavramdır. Bu yüzden, cümledeki ana fiil "chasing" olmuştur. "There" tamamlayıcı özne görevinde olup çeviride boş olarak tanımlanmıştır. "to be" fiili olan "is" ise zaman (tense) ifadesidir.



Şekil 3.11 Zaman öbeği ayrıştırma ağacı

İsim öbeği olan ve trans özelliği *Trans1* olan bir öbek ile fiil öbeği ve trans özelliği *Trans2* olan bir fiil öbeğinin Şekil 3.12'deki gibi birleşmesiyle oluşan öbek *cümle*

oluşturur. Bu *cümle* trans özelliği ise *Trans1* ve *Trans2*'nin birleşmesiyle *Trans* olarak oluşturulup *cümle* seviyesine trans bilgisi taşınmıştır.

```
phrase(loc:(cat:tp,trans:Trans,pred:Pred),
nonloc:(subj:no,comp:Comp)) ==>
phrase(loc:(cat:np,trans:Trans1,pred:Pred)),
phrase(loc:(cat:tbar,trans:Trans2,pred:_),
nonloc:(subj:Pred,comp:Comp)):-
append(Trans1,Trans2,Trans).
```

Şekil 3.12 Prolog derleyicisinde cümle oluşturma

3.4.1 PROLOG UYGULAMASI

Prolog programlama diliyle geliştirilen uygulamada Ötele-İndirge Ayrıştırıcısı (Shift-Reduce Parser), Birleştirme (Unification), Özellik Yapısı (Feature Structure) , Geri İzleme (Backtracking) konularından yararlanılmıştır.

3.4.1.1 Ötele-İndirge Ayrıştırıcısı

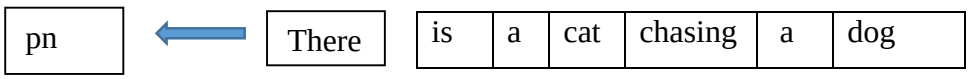
Bir ötele-indirge ayrıştırıcısı, dilbilgisi sembollerini içeren bir ayrıştırma yığını (stack) kullanır. Yığın, depolanan veriler içinde en son verinin ilk ulaşılabilir olduğu veri yapısıdır (LIFO).

Bu ayrıştırıcı, aşağıdan yukarıya doğru bir algoritma olduğundan (Bottom-up algorithm), sözcüklere ve kategorilerine bakarak başlar. Geçerli bir kural varsa bu bilgiler ışığında cümle kurmaya çalışır. Öteleme ve indirgeme olarak iki farklı işlemle bunu gerçekleştirir. Öteleme işlemiyle cümledeki ilk çözümlenmemiş sözcüğü alır, onun sözlük kuralına bakar ve sözcüğün kategorisinin bir simgesini yığının üzerine koyar. İndirgeme işlemiyle ayrıştırıcı dilbilgisi kuralının sağ tarafında bulunan ve yığında en üstte bulunan kategori simgelerini eşleştirmeye çalışır. Bu başarılı olursa, simgeleri kaldırır ve bunun yerine kuralın solundaki kategori için bir simgeyi yığının üzerine koyar. Aşağıdaki örnek cümle ile bu ayrıştırıcının nasıl çalıştığı açıklanmaktadır.

Örnek cümle: There is a cat chasing a dog.

Öteleme işlemi

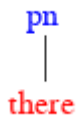
Yığın:



Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

Öteleme işleminde ilk olarak “there” sözcüğü alınır ve sözlükten sözcük kategorisi bulunur daha sonra bulunan kategori simgesi yığına atılır.

Çıktı



İndirgeme işlemi

Yığın:

np	is	a	cat	chasing	a	dog

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

İndirgeme işleminde np→ pn dilbilgisi kuralına uyan bir yapı elde edilir. Yığında zamir (pn) olarak var olan sözcük kategorisi isim öbeği (np) yapısına dönüştürülür.

Çıktı



Öteleme işlemi

Yığın:

np	t



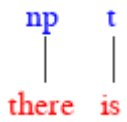
a

a	cat	chasing	a	dog

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

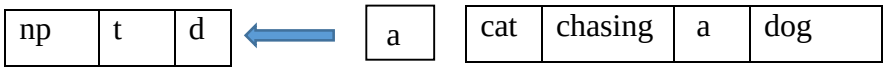
Öteleme işleminde “is” sözcüğü alınır ve sözlükten sözcük kategorisi bulunur daha sonra bulunan kategori simgesi yığına atılır.

Çıktı



Öteleme işlemi

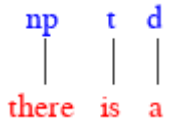
Yığın:



Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

Öteleme işleminde “a” sözcüğü alınır ve sözlükten sözcük kategorisini bulunur daha sonra bulunan kategori simgesi söz yığına atılır.

Çıktı



Öteleme işlemi

Yığın:

np	t	d	n



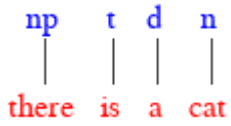
cat

chasing	a	dog

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

Öteleme işleminde “cat” sözcüğü alınır ve sözlükten sözcük kategorisi bulunur daha sonra bulunan kategori simgesi yığına atılır.

Çıktı



İndirgeme işlemi

Yığın:

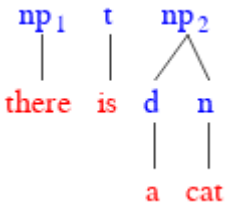
np	t	np

chasing	a	dog

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

İndirgeme işleminde np → d, n dilbilgisi kuralına uyan bir yapı elde edilir. Yığında, artikel (d) ve isim (n) olarak var olan sözcük kategorisi isim öbeği (np) yapısına dönüştürülür.

Çıktı



Öteleme işlemi

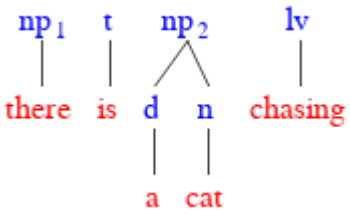
Yığın:



Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

Öteleme işleminde “chasing” sözcüğü alınır ve sözlükten sözcük kategorisi bulunur daha sonra bulunan kategori simgesi yığına atılır.

Çıktı



Öteleme işlemi

Yığın:

np	t	np	lv	d



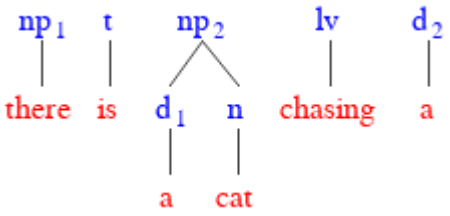
a

dog

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

Öteleme işleminde “a” sözcüğü alınır ve sözlükten sözcük kategorisi bulunur daha sonra bulunan kategori simgesi yığına atılır.

Çıktı



Öteleme işlemi

Yığın:

np	t	np	lv	d	n

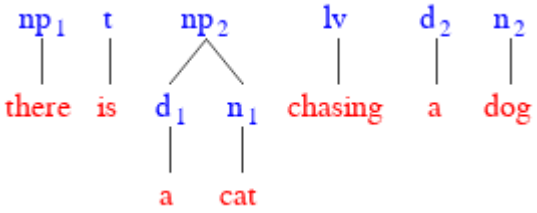


dog

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

Öteleme işleminde "dog" sözcüğü alınır ve sözlükten sözcük kategorisi bulunur ve daha sonra bulunan kategori simgesi yığına atılır.

Çıktı



İndirgeme işlemi

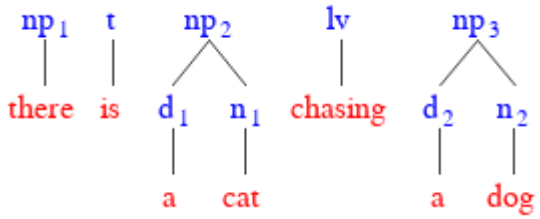
Yığın:

np	t	np	lv	np

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

İndirgeme işleminde np → d, n dilbilgisi kuralına uyan bir yapı elde edilir. Yığında, artikel (d) ve isim (n) olarak var olan sözcük kategorisi isim öbeği (np) yapısına dönüştürülür.

Çıktı



İndirgeme işlemi

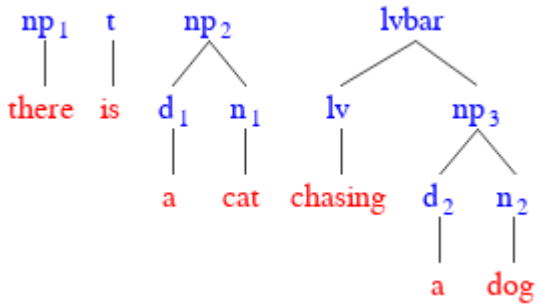
Yığın:

np	t	np	lvbar

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

İndirgeme işleminde lvbar →lv, np dilbilgisi kuralına uyan bir yapı elde edilir.

Çıktı



İndirgeme işlemi

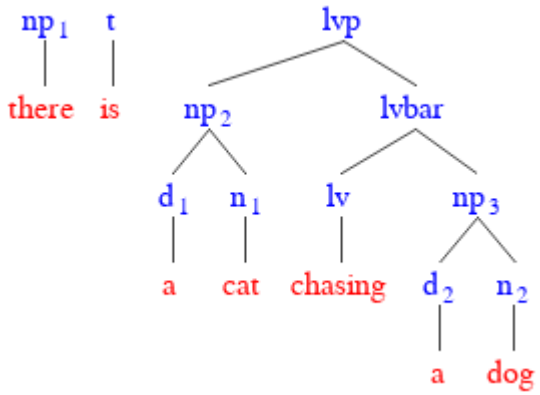
Yığın:

np	t	lvp

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp
d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

İndirgeme işleminde lvp → np, lvbar dilbilgisi kuralına uyan bir yapı elde edilir.

Çıktı



İndirgeme işlemi

Yığın:

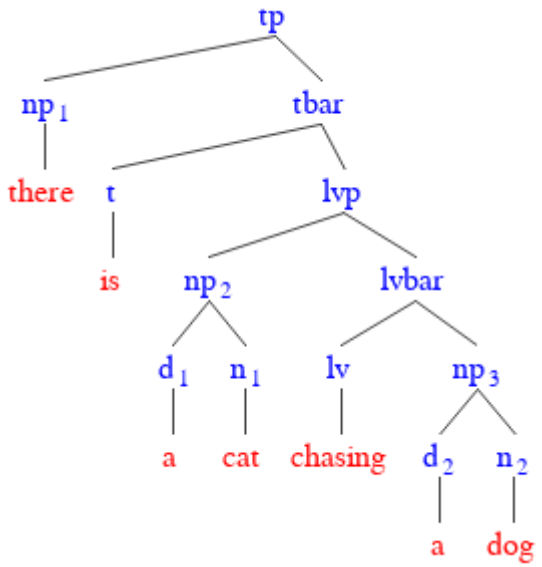
tp

Sözlük (Lexicon)	Dilbilgisi kuralları
n→cat	tp → np, tbar
t→is	np → pn
d →the	tbar→ t, lvp

d →a	lvp → np, lvbar
lv → chasing	np → d, n
pn →there	lvbar →lv, np

İndirgeme işleminde tp → np, tbar dilbilgisi kuralına uyan bir yapı elde edilir.

Çıktı



3.4.1.2 Birleştirme

Prolog'un iki terimle eşleşme biçimi, birleştirme olarak adlandırılır. Değişken örnekleme (variable instantiation) yoluyla aynı yapıya sahip iki ifade oluşturulabiliyorsa, birleştirme işi başarılı olur. Birleştirmenin amacı, bir dizi ifadelerin ortak durumlarını bulmak ve uygulamaktır. Birleştirme, '=' operatörüyle veya parametre aktarımı yoluyla gerçekleştirilebilir. Bu işlem, simetrik bir işlemdir ve atama işlemi ile aynı değildir. Birleştirme kuralları ve yöntemleri aşağıda verilmiştir.

- Herhangi bir değer kendisiyle birleştirilebilir.

Örnek: anne (ali) = anne (ali).

- Bir değişken, başka bir değişkenle birleştirilebilir. Bundan sonra iki değişken ismi aynı değişkeni referans alır.

Örnek: X = Y, X = 2, yaz (Y). İfadesi ekrana 2 değerini yazdırır.

- Bir değişken herhangi bir Prolog değeri ile birleştirilebilir, bu değişkeni örneklendirmek olarak adlandırılır. Bir değişken, değişken içermeyen bir değerle birleştirildiğinde, tam olarak örneklenir (Unification and instantiation, 2017).

Örnek: X = foo (bar, [1, 2, 3]). /*X tam örneklendi. * /

Örnek: Ali = koca (Ayşe). / * Ali kısmen örneklendi. * /

- Değerleri aynı hale getiren kurucu değişkenler için birleşmeler varsa, iki farklı değer birleştirebilir.

Örnek: anne (Ayşe, X) = anne (Y, baba (Z)). [Ayrıca Ayşe = Y ve X = baba (Z)] birleşimlerine neden olur.

- Bir değişkenin kendisini içeren bir ifade ile birleşebilir. Bununla birlikte, elde edilen değer yazdırılamaz.

Örnek: X = foo (X, Y)

3.4.1.3 Özellik Yapısı

Prolog'ta özellik yapısı (feature structure) genellikle özellik değeri matrisleri (attribute value matrix) olarak yazılmaktadır. Bu matris aşağıdaki gibi örneklendirilmiştir.

$$\begin{bmatrix} sayı & tekil \\ kişi & 3 \end{bmatrix}$$

Sayı özelliği, bir sözcüğün tekil ya da çoğul olduğunu belirten bir özelliktir. Kişi özelliği ise şahıs ekini ifade eden bir özeliktir.

Özellik yapısı kendi içinde özellik yapısı içerebilmektedir. Aşağıdaki matris bu durumu örneklemektedir.

$$\begin{bmatrix} Loc & \begin{bmatrix} cat & d \\ trans & Trans1 \end{bmatrix} \end{bmatrix}$$

Genel cümle yapılarında, bir özellik yapısı temel olarak bir öznitelik-değer çifti grubudur. Bir özellik yapısı, değişken değerlere ve değişken adlara giden yollara karşılık gelen düğümlerle, yönlendirilmiş bir DAG (directed acyclic graph) grafiği olarak da gösterilebilir (Feature Structures, 2017).

Geliştirilen uygulamadaki örnek bir kod satırının özellik değeri matrisi aşağıdaki gibidir:

$$\begin{bmatrix} Loc & \begin{bmatrix} cat & n \\ trans & köpek \\ pred & individual \end{bmatrix} \end{bmatrix}$$

```
word(loc:(cat:n,trans:[köpek],pred:individual)) => [dog].
```

Cat, trans, pred özelliklerinin aldığı değerler birleşip *loc* özelliğinin değerini oluşturmaktadır.

- *Cat* özelliği, sözcük kategorisini belirlemektedir. İsim, sıfat, zarf, fiil, zamir ve bunların oluşturduğu öbeklerden oluşabilmektedir.
- *Trans* özelliği, sözcüklerin çeviri karşılıklarını içermektedir.
- *Pred* özelliği, yüklemlerin türünü belirler. Yüklem geçici nitelikli ise *stage* değerini alır, öz nitelikli ise *individual* değerini alır.

3.4.1.4 Geri İzleme

Geri izleme (backtracking) temel olarak bir arama biçimidir. Örneğin, Prolog yorumlayıcısının hedef_1, hedef_2 gibi bir dizi hedefi gerçekleştirmeye çalıştığını varsayılsın. Prolog yorumlayıcısı, hedef_1'i gerçekleştirmek için izin verilen değişken bağlamalar kümesini bulduğunda, bu bağlamalara odaklanır ve sonra hedef_2'yi gerçekleştirmeyi amaçlar. Sonuçta iki şeyden biri olur:

1. hedef_2 tamamlandı
2. hedef_2 tamamlanamadı

Her iki durumda da, Prolog geri izler. Yani, hedef_1'i gerçekleştirmek için yaptığı değişken bağlamaları kaldırır ve hedef_1'in gerçekleştirilmesine izin veren farklı

değişken bağları arar. Eğer ikinci bir takım bağlar bulursa, onları işler ve yeni bağlamlarla tekrar hedef_2'yi gerçekleştirmeye çalışır. İlk durumda, Prolog tercümanı ekstra çözüm arar, ancak ikinci durumunda hala birinci çözümü arar. Bu yüzden geri izleme, bir soruna ilave çözümler bulmaya yarayabilir ya da ilk varsayımlar dizisi bir çözüme götürmediğinde ilk çözümü aramaya devam etmede yarayabilir. Diğer bir deyişle, birden çok alt hedefi bulunan bir hedef varsa ve prolog derleyicisi alt hedeflerden birinin doğruluğunu göstermekte başarısız olursa alternatif bir çözüm için bir önceki alt hedef yeniden ele alır.

BÖLÜM 4

DEĞERLENDİRME VE SONUÇ

Yapay zekâ alanında yapılan çalışmaların hız kazanmasıyla birlikte insanların yapmış olduğu çeviri işini bilgisayara aktarma işi dilbilim alanında ilgi çeken konulardan biri olmuştur. Çeviri işinin bilgisayarla otomatikleştirilmesi için çeşitli yaklaşımlar öne sürülmüştür. Bu yaklaşımlar, kural tabanlı ve istatiksel olmak üzere 2 ana grup altında toplanmıştır. Bu çalışmada, kural tabanlı makine çevirisi yaklaşımlarından sözdizimsel aktarım yöntemiyle bir makine çevirisi yapılmıştır.

Çizelge 4.1'deki varoluş cümle yapılarında sözdizimsel aktarım yöntemi ile geliştirilen uygulama sonucu çevirileri ile Google Translate çevirileri karşılaştırılmıştır. Google Translate istatistiksel makine çevirisi ve sinirsel makine çevirisi (neural machine translation) yöntemlerini kullanarak çeviri yapmaktadır.

Çizelge 4.1 Makine çevirisi yöntemlerini karşılaştırma

İngilizce Cümle	**Türkçe'ye Sözdizimsel Aktarım ile Çeviri**	**Türkçe'ye Google Translate ile Çeviri**
there is a cat	bir kedi var	orada bir kedi var
there is a dog barking	bir köpek havlıyor	havlayan bir köpek var
there is a cat chasing a dog	bir kedi bir köpek kovalıyor	bir kedi peşinde koşuyor
there is a cat rolling a stone	bir kedi bir taş yuvarlıyor	bir taş olan bir kedi var
a cat is rolling	bir kedi yuvarlanıyor	bir kedi yuvarlanıyor
a cat is rolling a stone	bir kedi bir taş yuvarlıyor	Bir kedi taş yuvarlamaktadır
the dog is black	köpek siyah	köpek siyah
the dog is a cat	köpek bir kedi	köpek bir kedi

Uygulama, sözcük türleri isim, sıfat, zarf, artikel, fiil, zamir olan 252 adet sözcük içerir. Sözlük genişledikçe, varoluş cümle yapılarındaki kurallara uyan daha çok sayıda cümle çevirebilmektedir.

Örneğin "there is a dog barking" cümlesini istatistiksel tabanlı çeviri yapan Google translate, "havlayan bir köpek var" şeklinde çevirmektedir. Kural tabanlı geliştirilen bu uygulamada varoluş cümleleri dilbilim teoremlerine hipotezlerine göre

kodlanmıştır. Buna göre “bir köpek havlıyor” şeklinde çevirmektedir. “There” tamamlayıcı öznedir ve anlamı yoktur. Dolasıyla asıl yüklem geçici nitelikli olan “bark” fiilidir.

Uygulamada kodlanan varoluş cümle yapılarına uyan cümlelerde çeviri başarılı sonuçlar elde etmiştir. Kural tabanlı makine çevirisi yapan bu uygulamaya ek olarak istatistiksel makine çevirisi yöntemi ile hibrit çalışmalar yapılırsa çok daha başarılı sonuçlar elde edilebilir.

BÖLÜM 5

EKLER

```
:- op(1100,xfy,=>).
:- op(1100,xfy,==>).
:- include('C:/Users/sonne/Desktop/masaüstü_son/masaüstü/MT/utilities.pl').

%% Sözlük

word(loc:(cat:n,trans:[köpek],pred:individual)) => [dog].
word(loc:(cat:n,trans:[kuş],pred:individual)) => [bird].
word(loc:(cat:n,trans:[taş],pred:individual)) => [stone].
word(loc:(cat:n,trans:[kedi],pred:individual))=> [cat].
word(loc:(cat:pn,trans:[],pred:stage))=>[there].
word(loc:(cat:d,trans:[bir])) => [a].
word(loc:(cat:d,trans:[])) => [the].
word(loc:(cat:adj,trans:[siyah],pred:ind)) => [black].
word(loc:(cat:v,trans:[havlıyor],pred:stage))=>[barking].
word(loc:(cat:lv,trans:[kovalıyor],pred:stage))=>[chasing].
word(loc:(cat:t,trans:[])) => [is].
word(loc:(cat:t,trans:[idi])) => [was].
word(loc:(cat:adv,trans:[çılgınca])) => [furiously].
word(loc:(cat:v,trans:[yuvarlanıyor],pred:stage)) => [rolling].
word(loc:(cat:lv,trans:[yuvarlıyor],pred:stage)) => [rolling].
word(loc:(cat:v,trans:[ötüyor],pred:stage))=>[singing].
word(loc:(cat:c,trans:[' mı?'])) => [is].
word(loc:(cat:c,trans:[' mıydı?'])) => [was].



%% Öbek Yapı Kuralları
```

```
phrase(loc:(cat:cp,trans:Trans,pred:_),
       nonloc:no) ==>
       word(loc:(cat:c,trans:Trans1)),
       phrase(loc:(cat:tp,trans:Trans2,pred:_),
              nonloc:(subj:no,comp:yes)):-
              append(Trans2,Trans1,Trans).

phrase(loc:(cat:tp,trans:Trans,pred:Pred),
       nonloc:(subj:no,comp:Comp)) ==>
      phrase(loc:(cat:np,trans:Trans1,pred:Pred)),
      phrase(loc:(cat:tbar,trans:Trans2,pred:_),
      nonloc:(subj:Pred,comp:Comp)):-
           append(Trans1,Trans2,Trans).

phrase(loc:(cat:tbar,trans:Trans,pred:Pred),
       nonloc:(subj:Subj,comp:no)) ==>
      word(loc:(cat:t,trans:Trans1)),
      phrase(loc:(cat:Cat,trans:Trans2,pred:Pred),nonloc:subj:Subj):-
           (Cat = vp; Cat = lvp),
           append(Trans2,Trans1,Trans).



phrase(loc:(cat:tbar,trans:Trans,pred:ind),
       nonloc:(subj:individual,comp:no)) ==>
      word(loc:(cat:t,trans:Trans1)),
      word(loc:(cat:adj,trans:Trans2,pred:ind)):-
         append(Trans2,Trans1,Trans).

phrase(loc:(cat:tbar,trans:Trans,pred:_),
       nonloc:(subj:individual,comp:no)) ==>
```

```
    word(loc:(cat:t,trans:Trans1)),
    phrase(loc:(cat:np,trans:Trans2,pred:_)):-
        append(Trans2,Trans1,Trans).

phrase(loc:(cat:lvp,trans:Trans,pred:_),nonloc:subj:stage) ==>
    phrase(loc:(cat:np,trans:Trans1,pred:_)),
    phrase(loc:(cat:lvbar,trans:Trans2,pred:_)):-
        append(Trans1,Trans2,Trans).

phrase(loc:(cat:lvp,trans:Trans,pred:stage),nonloc:subj:individual) ==>
    phrase(loc:(cat:lvbar,trans:Trans,pred:_)).

phrase(loc:(cat:lvbar,trans:Trans,pred:_)) ==>
    word(loc:(cat:adv,trans:Trans1,pred:_)),
    phrase(loc:(cat:lvbar,trans:Trans2,pred:stage)):-
        append(Trans1,Trans2,Trans).

phrase(loc:(cat:lvbar,trans:Trans,pred:_)) ==>
    word(loc:(cat:lv,trans:Trans1,pred:_)),
    phrase(loc:(cat:np,trans:Trans2,pred:_)):-
        append(Trans2,Trans1,Trans).

phrase(loc:(cat:vp,trans:Trans,pred:_),nonloc:subj:stage) ==>
    phrase(loc:(cat:np,trans:Trans1,pred:_)),
    phrase(loc:(cat:vbar,trans:Trans2,pred:_)):-
        append(Trans1,Trans2,Trans).

phrase(loc:(cat:vp,trans:Trans,pred:_),nonloc:subj:individual) ==>
    phrase(loc:(cat:vbar,trans:Trans,pred:_)).
phrase(loc:(cat:vp,trans:Trans,pred:_),nonloc:subj:stage) ==>
    phrase(loc:(cat:np,trans:Trans1,pred:_)):-
        append(Trans1,[var],Trans).
```

```
phrase(loc:(cat:vbar,trans:Trans,pred:Pred)) ==>
     word(loc:(cat:v,trans:Trans,pred:Pred)).

phrase(loc:(cat:np,trans:Trans,pred:Pred)) ==>
     word(loc:(cat:d,trans:Trans1)),
     word(loc:(cat:n,trans:Trans2,pred:Pred)):-
        append(Trans1,Trans2,Trans).

phrase(loc:(cat:np,trans:Trans,pred:Pred)) ==>
     word(loc:(cat:pn,trans:Trans,pred:Pred)).

%% PARSER
% Top
parse(Expression,Str):- parse1(Expression,R),
                        list_to_string(R,Str).

parse1(Expression,R):-
      tokenize(Expression,L),
      sr_parse([],L,R).
% Base
sr_parse([phrase(loc:(cat:_,trans:Trans,pred:_),nonloc:_)],[],Trans).

% Reduce
sr_parse([Y,X|Rest],Words,R):-
  (Z ==> (X,Y)),
  sr_parse([Z|Rest],Words,R).

sr_parse([X|Rest],Words,R):-
  (Y ==> X),
  sr_parse([Y|Rest],Words,R).
```

```
% Shift
sr_parse(Stack,[Word|Words],R):-
   (Cat => [Word]),
   sr_parse([Cat|Stack],Words,R).
test:-
  open('C:/Users/sonne/Desktop/MT/test.pl',read,Input),
  read_all(Input,0,0).
read_all(Input,SuccessIN,TotalIN):-
  read(Input,Term1),
  (Term1 == end_of_file ->
          (write(SuccessIN),
           write(' out of '),
           write(TotalIN),
           write(' words have been successfully parsed!'),
           close(Input));
          (TotalOUT is TotalIN + 1,
           read(Input,Term2),
           parse(Term1,Trans),
           ((Trans == Term2) -> SuccessOUT is SuccessIN + 1;
                        (SuccessOUT is SuccessIN, write(Term1),nl)),
            read_all(Input,SuccessOUT,TotalOUT))).
list_to_string([],'').
list_to_string([Word|Rest],String):-
    list_to_string(Rest,String1),
    (String1 == '' -> Word1 = Word;concat(Word,' ',Word1)),
    concat(Word1,String1,String).
```

KAYNAKLAR

Carlson, Gregory N. (1977) *Reference to Kinds in English*. Ph.D. thesis. University of Massachusetts. Amherst.

Chevalier, M., Dansereau, J., & Poulin, G. (1978). *TAUM-METEO: Description du Système*, Universite de Montreal.

Chomsky, N. (1970). Noam. *Remarks on nominalization*. In: R. Jacobs and P. Rosenbaum (eds.) *Reading in English Transformational Grammar*, 184-221. Waltham: Ginn.

Chomsky, N. (1995). *The Minimalist Program*. Cambridge, MA: MIT Press.

Erenler, Y. (2005). BLG 505 Doğal Dil İşleme Ders Notları, İTÜ Fen Bilimleri Enstitüsü, İstanbul.

Feature Structures, http://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node81.html#l11.sec.fs, Erişim Tarihi: 15.12.2017

Hutchins, J., & Lovtskii, E. (2000). *Petr Petrovich Troyanskii (1894-1950): A forgotten pioneer of mechanical translation*, Machine Translation, 15(3), 187–221.

Hutchins, W. J., *The Georgetown-IBM Experiment Demonstrated in January 1954*, Machine Translation: From Real Users to Research, Springer Berlin.

King, M., & Perschke, S. (1982). .*EUROTRA and Its Objectives*, Multilingua-Journal of Cross-Cultural and Interlanguage Communication, 1(1), 27-32.

Kratzer, A. (1996). Severing the external argument from its verb. *In Phrase structure and the lexicon*, Rooryck & Zaring (eds),. Dordrecht, The Netherlands: Kluwer Academic Publishers. 109 – 137.

Pierce, J. R., & Carroll, J. B. (1966). *Language and Machines: Computers inTranslation and Linguistics*, National Academy of Sciences, National Research Council, Washington, D.C.

Radford, A. (1988).*English Syntax An Introduction*. Cambridge, Cambridge University Press..

TDK Güncel Türkçe Sözlük, http://tdk.gov.tr/index.php?option=com_gts&arama=gts&guid=TDK.GTS.5677e7aaec7055.71589339 , Erişim Tarihi: 21.12.2015

Toma, P. (1977). *SYSTRAN as a Multilingual Machine Translation System*, Proceedings of the Third European Congress on Information Systems and Networks, Overcoming the language barrier, 569-581.

Unification and instantiation, http://matuszek.org/prolog/prolog-unification.html, Erişim Tarihi: 15.12.2017

Wikipedia,*Vikipedi*,Adres:https://en.wikipedia.org/wiki/Comparison_of_machine_translation_applications, Erişim Tarihi: 21.12.2015

Wikipedia,*Vikipedi*,Adres:http://tr.wikipedia.org/wiki/Doğal_dil_işleme, Erişim Tarihi: 21.12.2015

ÖZGEÇMİŞ

Tuğba ÇALIŞ, 23 Nisan 1989 tarihinde Denizli'de doğdu. Lise öğrenimini Cumhuriyet Anadolu Lisesi'nde tamamladı. 2008 yılında girdiği Çankaya Üniversitesi Bilgisayar Mühendisliği (İngilizce -Tam burslu) Bölümü'nden, 2013 yılında mezun oldu. 2015 yılında Trakya Üniversitesi Bilgisayar Mühendisliği Bölümünde Yüksek Lisans çalışmalarına başladı. 2015 yılından beri Trakya Üniversitesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır.