

# TÜMEVARIMLI MANTIK PROGRAMLAMANNIN KELİME ANLAMI BELİRGİNLEŞTİRMEYE UYGULANABİLİRLİĞİNİN İNCELENMESİ

Özlem Aydın<sup>1</sup>

Trakya Üniversitesi  
Bilgisayar Mühendisliği Bölümü  
ozlema@trakya.edu.tr

Yılmaz Kılıçaslan<sup>2</sup>

Trakya Üniversitesi  
Bilgisayar Mühendisliği Bölümü  
yilmazk@trakya.edu.tr

## ÖZET

Bilgisayarlı dilbilimde, kelime anlamı belirginleştirme (Word Sense Disambiguation), bağlamdaki bir kelimenin doğru anlamının belirlenmesi olarak tanımlanır. Doğal dil uygulamaları arasında önemli bir yere sahiptir. Tümevarımlı mantık programlama (Inductive Logic Programming, ILP) ise makine öğrenmesi ve mantık programlama disiplinlerinin kesişim alanını kapsar. Bu sebeple tümevarımlı mantık programlama, makine öğrenmesi ve mantık programlama tekniklerinin her ikisini de kullanır. Tümevarımlı Mantık Programlama ile yapılan çalışmalarda elde edilen deneysel sonuçlar tümevarımlı mantık programlamanın artalan bilgisini (background knowledge) kullanmada başarılı olduğunu göstermiştir. Tümevarımlı mantık programlamanın bu yeteneğinin kelime anlamı belirginleştirme konusunda kullanılması bu alanda önemli bir gelişme sağlayabilir. Bu çalışmada, tümevarımlı mantık programlama konusu incelenmiş ve kelime anlamı belirginleştirmeye uygulanabilirliği Türkçe veriler üzerinden elde edilen deneysel örneklerle gösterilmiştir.

**Anahtar Kelimeler:** Kelime anlamı belirginleştirme, tümevarımlı öğrenme, tümevarımlı mantık programlama, makine öğrenmesi.

## 1. GİRİŞ

Doğal dil işleme çalışmalarında amaca uygun doğru sonuçlar elde etmek oldukça zordur. Çünkü dillerdeki esneklik nedeniyle farklı belirsizlik durumları ile karşılaşılır. Doğal dil işlemede farklı dillerdeki bu belirsizlik durumlarının çözülmesi gerekmektedir. Kelime anlamı belirsizliği bu belirsizlik tiplerinden biridir ve bir kelime çok anlamlı olduğu zaman ortaya çıkar. Bu belirsizlik durumunu bir örnek ile açıklayalım. Elimizde “el” kelimesinin içinde geçtiği bir tümce bulunsun.

“İnce uzun parmaklı elleri masanın üstündeydi.”

El kelimesinin Türk Dil Kurumu Türkçe Sözlük’ünden aldığımız en sık kullanılan üç anlamı aşağıdaki gibidir:

1. anlam: Kolun bilekten parmak uçlarına kadar olan, tutmaya ve iş yapmaya yarayan bölümü.
2. anlam: Sahiplik, mülkiyet.
3. anlam: İskambil oyunlarında her bir tur.

Görüldüğü gibi “el” kelimesinin birden fazla anlamı bulunmaktadır. Bu aşamada verilen tümcede 1. anlamı ile kullanıldığı belirlemek için KAB işlemi uygulanması gereklidir.

KAB hesaplamalı dilbilimde temel bir problem olarak ele alınmaktadır. Makine çevirisi (machine translation) ve yapay zekâ (artificial intelligence) alanlarına olan bilimsel ilginin artışı ile KAB bir araştırma alanı olarak ortaya çıkmıştır. Bu alandaki ilk

çalışmalar 1950’li yıllarda makine çevirisi için gerek duyularak başlamıştır [1][2]. 1960’lı yıllarda ise yapay zekânın doğal dil anlama uygulamalarında kullanılmaya başlamasıyla yeniden bir yükselişe geçmiştir [3][4][5]. 1980’li yıllardan itibaren büyük boyutlardaki sözlüklerin (dictionaries), gömü (thesauri) ve derlem (corpus) gibi sözlük benzeri kaynakların kullanılmasıyla KAB çalışmalarında artış olmuştur [6][7][8].

KAB, birden fazla anlamı olan bir kelimenin bulunduğu bağlamda hangi anlamıyla kullanıldığının belirlenmesi işlemidir. Bu işlem iki temel adıma ayrıştırılabilir [9]:

1. Metinde bulunan ilgili kelimenin tüm farklı anlamlarının belirlenmesi.
2. Uygun anlamın atanması.

Son yıllarda yapılan çoğu KAB çalışmaları birinci adım için aşağıda verilen bilgilerle anlamların önceden belirlenmesine dayanır.

- Günlük sözlüklerde de bulanabilen bir anlam listesi.
- Bir özellikler, kategoriler veya ilgili kelimeler kümesi (örneğin, bir gömüdeki eşanlamlılar).
- Bir makine çevirisi uygulaması için KAB işlemi yapılacaksa diğer dillere çevirileri içeren bir çeviri sözlüğündeki bir giriş.

İkinci adım kelimelere anlamların atanmasıdır. İki büyük bilgi kaynağı kullanılabilir: belirginleştirilecek

kelimenin bulunduğu bağlam ve harici bilgi kaynakları [9].

KAB birçok alana uygulanmaktadır. Bu uygulama alanlarından bazıları şunlardır: Makine Çevirisi (Machine Translation), Bilgi Geri Getirimi (Information Retrieval), Ses İşleme (Speech Processing), İnsan-Makine Haberleşmesi (Man-Machine Communication), Metin İşleme (Text Processing), İçerik ve Tematik Analiz (Content and Thematic Analysis) ve Gramer Çözümlemesi (Gramatical Analysis). KAB genelde doğal dil işleme uygulamalarında kullanılmasına rağmen son yıllarda Biyoenformatik (Bioinformatics) ve Semantik Web alanlarındaki önemi artmıştır.

KAB işlemi sürecinde faydalı bazı bilgi türleri de bulunmaktadır. Bunlar kategori bilgisi (Part of speech-POS), anlamların sıklıkları (frequency of senses), eşdizimlilikler (collocations), seçilebilir tercihler (selectional preferences), alt-ögeleme bilgisi (subcategorization information) ve semantik bağlam vb. olabilir. Bu bilgi türlerini elde etmek için kullanılan bazı bilgi kaynakları vardır. Bunlardan bazıları elektronik sözlükler, teknik sözlükler ve derlemelerdir. KAB sistemleri verilen hedef kelimenin bağlamsal özelliklerini bu bilgi kaynaklarından elde edip kelimenin farklı anlam karşılıklarını kıyaslamada kullanılır.

Günümüze kadar yapılan çalışmalarda kullanılan kaynaklar ve tekniklerin farklı birleşimlerinden çeşitli KAB yöntemleri ortaya çıkmıştır. Bu yöntemler genel olarak sınıflandırılacak olursa, bilgi tabanlı KAB ve derlem tabanlı KAB olmak üzere iki grup altında toplanabilir.

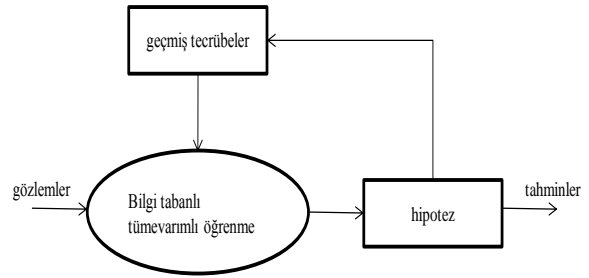
1. Bilgi tabanlı KAB'da sözlük benzeri kaynaklardan alınan bilgi kullanılmaktadır. Büyük boyutlardaki elektronik sözlüklerin hazırlanması ile 1980'li yıllarda bu sözlükler kullanılmaya başlamıştır. Bu kaynakların oluşturulması bilginin otomatik olarak çıkarılması çalışmalarını hızlandırmıştır.
2. Derlem tabanlı KAB'da en başarılı yaklaşımlar derlemden sınıflayıcı veya istatistiksel modelleri öğrenmek için kullanılan istatistiksel veya makine öğrenmesi algoritmalarıdır. Zaten bu yöntemlerin çoğu doğal dil işleme çalışmalarına uygulanmaktadır. Bu algoritma ve tekniklere olan ilgideki artış, makine öğrenmesi konusuna duyulan ilgiden kaynaklanmaktadır. Çünkü makine öğrenme algoritmaları çoğu doğal dil işleme çalışmalarında dikkat çekici bir başarı göstermiştir. Bu algoritmalar genel olarak öngörmeli (supervised) ve öngörmesiz (unsupervised) olmak üzere ikiye ayrılır. KAB için yapılan çalışmalarda öngörmeli yaklaşımların, belirsizliği fazla olan kelimeler üzerinde öngörmesiz yaklaşımlardan daha iyi sonuçlar verdiği görülmüştür. Öngörmesiz KAB yönteminde anlam etiketli olmayan bir derlem kullanmadan belirginleştirme gerçekleştirilir. Öngörmeli makine öğrenme ile gerçekleştirilen KAB'da bir sınıflayıcı yaratılır.

Burada amaç geçmiş durumlara bakarak yeni durumları doğru olarak sınıflandıran bir sınıflayıcı oluşturmaktır. Bayes öğrenme (Bayesian learning), karar listeleri (decision lists), karar ağaçları (decision trees), k-en yakın komşuluk algoritması (k-nearest neighbor algorithm) ve yapay sinir ağları (neural networks) gibi makine öğrenmesi algoritmaları sınıflayıcı örnekleridir.

KAB işlemi için bu çalışmada uygulanabilirliğini incelenecek olan Tümevarımlı Mantık Programlama (TMP)'da öğrenmeyi gerçekleştirmek için makine öğrenmesi teknikleri kullanılmaktadır. Takip eden bölümde TMP konusu anlatılmış, KAB işleminde kullanımı çeşitli yönleriyle incelenmiştir.

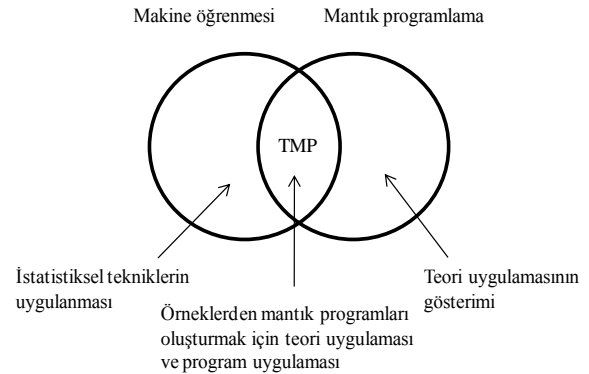
## 2. TÜMEVARIMLI MANTIK PROGRAMLAMA

Tümevarımlı öğrenme gözlemlerden, deneylerden veya bir veritabanından elde edilen örnek kümesinden genel bir kural çıkarmaya yönelik bir öğrenme tekniğidir. Elde edilen kural tüm örnek kümesini en iyi şekilde açıklayabilen bilgidir.



Şekil 1. Tümevarımlı Öğrenme Tekniği

TMP, makine öğrenmesi ve mantık programlamanın kesişimi olarak tanımlanabilir. Amacı gözlemlerden (örneklerden) hipotezler geliştiren ve artalan bilgisinden yeni bilgiler elde eden teknikler ve araçlar geliştirmektir. Bu sebeple TMP, makine öğrenmesi ve mantık programlama tekniklerinin her ikisini de kullanır [10].



Şekil 2. Makine Öğrenmesi, Mantık Programlama ve TMP

TMP arařtırmalarında amaç Horn tümcecigi<sup>1</sup> mantığında tanımlanan örnekler ve artalan bilgisi ile mantık programlamadır. Öğrenilen mantık programları genellikle Prolog sözdiziminde tanımlanır ve mantık programlarındaki bildirimsel (declarative) olma özelliği TMP'nin etkili olmasının ana nedenidir [10].

TMP, hipotezler ve gözlemler için gösterimsel mekanizma olarak hesaplamalı mantığı kullanarak klasik makine öğrenmesi tekniklerinde var olan iki önemli kısıtın üstesinden gelebilmektedir.

1. Sınırlı bilgi gösterimi formalizmini kullanmak.
2. Öğrenme sürecinde çok miktarda artalan bilgisini kullanmadaki zorluklar.

İlk sınırlama önemlidir; çünkü bazı alanlardaki incelemeler sadece birinci dereceden yüklem mantığı ile açıklanır, önermeler mantığı ile açıklanamaz. Açıkça problem şudur ki; mantık programı alanı örneklerden sentez yapar ve çoğu mantık programı sadece önermeler mantığı ile tanımlanamaz. Alan bilgisinin kullanımı önemlidir, çünkü yapay zekâdaki önemli buluşlardan biri zekâ davranışına ulaşmak için alan bilgisi kullanımının gerekli olduğudur [10]. Mantık bilginin gösterimine mükemmel bir formalizm sunar ve bundan dolayı tümevarımla bütünleşir.

TMP, çıkarımda temel mod olarak tümevarımı inceleyerek hesaplamalı mantığı uygular ve teoriyi genişletir. Her ne kadar şimdiki hesaplamalı mantık teorileri kullanıcıdan alınan mantık formülünden tümdengelimli çıkarım olarak tanımlansa da, TMP teorisi örneklerden ve artalan bilgisinden mantık programlarının tümevarımlı çıkarım yapması olarak tanımlanır.

TMP'de kullanılan mantıksal terimlerin tanımları aşağıda açıklanmıştır.

- $B$  sonlu sayıda tümcecik içeren artalan bilgisidir.
- $E$  sonlu sayıda örnek setidir =  $E^+ \cup E^-$ 
  - *Pozitif örnekler.*  $E^+ = \{e_1, e_2, \dots\}$  Boş olmayan tanımlı tümce kümesidir.
  - *Negatif örnekler.*  $E^- = \{\bar{f}_1, \bar{f}_2, \dots\}$  Horn tümcecigi kümesidir (boş olabilir).
- $B$  ve  $E$  aşağıdaki durumları sağladığında algoritmanın çıktısı  $H$ 'dir.
  - *Öncelikli kabul edilebilirlik (prior satisfiability).*  $B \cup E^- \neq \square$
  - *Sonraki kabul edilebilirlik (posterior satisfiability).*  $B \cup H \cup E^- \neq \square$
  - *Öncelikli gereklilik (prior necessity).*  $B \neq E^+$
  - *Sonraki yeterlilik (posterior sufficiency).*  $B \cup H \models e_1 \wedge e_2 \wedge \dots$

<sup>1</sup> Horn tümcecigi en fazla bir pozitif atomik önerme içeren, pozitif veya olumsuzlanmış atomik önermelerin 'veya' bağlacı ile bağlanması ile elde edilir.

Bu mantıksal terimlerin ne ifade ettiğini birinci dereceden yüklem mantığına benzer bir gösterim kullanarak bir örnek ile açıklayalım. Aile içindeki akrabalık ilişkilerinin öğrenileceği bir aile örneğini ele alalım. Bunun için verilenler şunlardır:

$$B = \begin{cases} dede(X, Y) \leftarrow baba(X, Z), ebeveyn(Z, Y) \\ baba(ali, ayşe) \leftarrow \\ anne(ayşe, ahmet) \leftarrow \\ anne(ayşe, suna) \leftarrow \end{cases}$$

Dede ve torunları arasındaki ilişkiyi gösteren pozitif örnekler aşağıda verilmiştir.

$$E^+ = \begin{cases} dede(ali, ahmet) \leftarrow \\ dede(ali, suna) \leftarrow \end{cases}$$

Ek olarak aşağıdaki negatif örneklerin de verildiğini varsayalım.

$$E^- = \begin{cases} \leftarrow dede(ahmet, ali) \\ \leftarrow dede(suna, ali) \end{cases}$$

$B$ 'nin doğrulunu kabul edip, yeni  $E^+$  ve  $E^-$  gerçekleri ile karşılaşıldığında aşağıdaki ilişki tahmin edilebilir.

$$H = ebeveyn(X, Y) \leftarrow anne(X, Y)$$

Burada  $H$ , elde edilen hipotezdir.

Literatürde uygulaması ve testi yapılmış birçok TMP öğrenciler vardır. Bu sistemler girdi biçimine göre deneysel (empirical) ve etkileşimli (interactive) olarak veya arama yönüne göre aşağıdan-yukarıya (bottom-up) ve yukarıdan-aşağıya (top-down) olmak üzere sınıflandırılır. Deneysel sistemler giriş olarak örnek kümesini ve artalan tümcelerini alırlar ve hipotez üretmek için çıktı olarak verirler. Bu sistemlerden bazıları FOIL, MFOIL, GOLEM, ALEPH, PROGOL, LINUS, MARKUS ve MOBAL'dir. Etkileşimli sistemler bir örnek seti ile başlarlar ve bir hipotez üretirler ve bu hipotezi sistemdeki bir uzman tarafından yönlendirilen soru cevaplar ile güncelleştirirler. Etkileşimli TMP sistemlerine örnek olarak MIS, CLINT, CIGOL ve MARVIN verilebilir.

TMP'nin farklı uygulama alanları vardır. Bunlardan bazıları; yazılım mühendisliği, sistemlerin modellenmesi ve kontrolü, doğal dil işleme, veritabanından bilginin çıkarımı, veritabanı tasarımı, veri analizi, tıbbi teşhisler, çevresel izleme sistemleri ve moleküler biyolojidir [11]. Ayrıca, TMP bilimsel buluşların çeşitli aşamalarında; örneğin teori oluşturmada, deneysel tasarım ve teori doğrulamada bir araç olarak kullanılabilir [12].

Çalışmamıza uygun olarak doğal dil işleme için bu konuyu daha ayrıntılı ele alalım. Doğal dil işleme için ilişkileri, özyinelemeleri ve sınırsız yapısal gösterimleri içeren çok zengin bir gösterimsel dil gerekmektedir. TMP'de birinci dereceden mantığın

kullanılmasının yarattığı zenginlik doğal dil işleme uygulamalarında esnekliği artırarak ve yapıcı tümevarıma olan gereksinim miktarını sınırlayarak ümit verici avantajlar kazandırmıştır [13].

### 3. GELİŞTİRİLEN UYGULAMA

TMP ile yapılan çalışmalarda elde edilen deneysel sonuçlar TMP'nin artalan bilgisini kullanmada başarılı olduğunu göstermiştir. TMP'nin bu yeteneğinin kelime anlamı belirginleştirme konusunda kullanılması bu alanda önemli bir gelişme sağlayabilir [14]. Belirginleştirme için artalan bilgisi olarak KAB'da faydalı olan bilgi türleri kullanılmaktadır. Bu bilgiler belirginleştirilecek kelime ve bulunduğu bağlamdaki diğer kelimelere ait sözdizimsel ve anlambilimsel özellikleri içermektedir.

Bahsedilen özellikleri elde edilmesi için kullanılan Türkçe doğal dil işleme kaynaklarının yeterince bulunmaması nedeniyle Türkçe için yapılmış KAB çalışması çok az bulunmaktadır. Bu eksikliği gidermek amacı ile bir sözlüksel kaynak olarak Türkçe Sözlüksel Örnek Görevi (Turkish Lexical Sample Task) oluşturulmuştur. Bu çalışma Semeval-2007 sunulmuştur [15].

KAB uygulamamızı bir TMP sistemi olan Aleph (A Learning Engine for Proposing Hypotheses) ile gerçekleştirdik [16]. Aleph'de teorileri oluşturmak için 3 veri dosyası gerekmektedir:

- Artalan bilgisinin bulunduğu .b uzantılı bir dosya.
- Pozitif örneklerin bulunduğu .p uzantılı bir dosya.
- Negatif örneklerin bulunduğu .n uzantılı bir dosya.

Bu dosyaların içerik bilgisini elde etmek için kaynak olarak Türkçe Sözlüksel Örnek Görevi kullanılmıştır. Şimdi bu veri dosyalarının nasıl oluşturulduğuna bakalım.

#### Artalan Bilgisi

Artalan bilgisi için kullandığımız özellikleri aşağıda vermiş olduğumuz bir örnek tümce üzerinden açıklayalım:

**Tümce:** “İnce uzun parmaklı elleri masanın üstündeydi.”

**Hedef kelime:** el

1. Anlamı belirginleştirilecek hedef kelimenin sağında ve solunda bulunan kelimeler. Uygulamada hedef kelimenin yanındaki ilk kelimeler alınmıştır. Yapılmış bazı çalışmalarda hedef kelimenin yanından alınan kelime sayısı beşe kadar çıkmaktadır. Ancak Türkçe Sözlüksel Örnek Görevi içinde hedef kelimenin yanındaki ilk kelimeler verildiği için uygulamamızı tek kelime sınırladık. Bu özelliğin programdaki gösterimi aşağıdaki gibidir:

*has\_bag(tümce\_no, kelime).*  
*has\_bag(snt1, parmak).*

*has\_bag(snt1, üst).*

2. Hedef kelimenin sağında ve solunda bulunan kelimeler ve bu kelimelerin hedef kelimeye göre konum bilgisi.

*has\_narrow(tümce\_no, kelimenin\_konumu, kelime).*  
*has\_narrow(snt1, first\_content\_word\_left, parmak).*  
*has\_narrow(snt1, first\_content\_word\_right, üst).*

3. Hedef kelimenin sağında ve solunda bulunan kelimelerin kategori bilgisi ve hedef kelimeye göre konum bilgileri.

*has\_pos(tümce\_no, kelimenin\_konumu, kelimenin\_kategori\_bilgisi).*  
*has\_pos(snt1, first\_content\_word\_left, noun).*  
*has\_pos(snt1, first\_content\_word\_right, verb).*

4. Hedef kelimenin sağında ve solunda bulunan kelimelerin ontolojik düzey bilgisi. Ontoloji düzeyi 3 olarak sınırlandırılmıştır.

*has\_ont(tümce\_no, kelimenin\_konumu, ontoloji\_düzeni, ontoloji).*  
*has\_ont(snt1, first\_content\_word\_left, 1, physical\_entity).*  
*has\_ont(snt1, first\_content\_word\_left, 2, organism).*  
*has\_ont(snt1, first\_content\_word\_left, 3, body).*  
*has\_ont(snt1, first\_content\_word\_right, 1, physical\_entity).*  
*has\_ont(snt1, first\_content\_word\_right, 2, location).*  
*has\_ont(snt1, first\_content\_word\_right, 3, region).*

5. Hedef kelimenin sağında ve solunda bulunan kelimelerin hal bilgileri ve hedef kelimeye göre konum bilgileri.

*has\_case(tümce\_no, kelimenin\_konumu, kelimenin\_hali).*  
*has\_case(tümce1, first\_content\_word\_left, \_).*  
*has\_case(tümce1, first\_content\_word\_right, \_).*

6. Hedef kelimenin sağında ve solunda bulunan kelimeler ile olan ilişki türü ve bu kelimelerin hedef kelimeye göre konum bilgileri.

*has\_rel(tümce, kelimenin\_konumu, ilişki\_türü).*  
*has\_rel(snt1, first\_content\_word\_left, modifier).*  
*has\_rel(snt1, first\_content\_word\_right, sentence).*

Artalan bilgisi dosyasında eğitim kümesi ve test kümesi örneklerinin özellikleri bulunmaktadır.

#### Pozitif Örnekler

Hedef kelimenin doğru anlamını gösteren bilgidir.  
*sense(tümce, hedef\_kelimenin\_anlamı).*  
*sense(snt1, sense1).*

## Negatif Örnekler

Hedef kelimenin doğru anlamını sağlamayacak şekilde diğer anlam etiketleri kullanılarak elde edilen bilgidir.

*sense(tümce, hedef\_kelimenin\_hatalı\_anlamı).*  
sense(snt1,sense3).

Geliştirilen uygulamada hedef kelime olarak belirlenen “el” kelimesi için Türkçe Sözlüksel Örnek Görevi’nde eğitim kümesinde 113 tümce, test kümesinde 38 tümce bulunmaktadır. Öncelikle eğitim kümesi tümcelerine ait artalan bilgisi, pozitif örnekler ve negatif örnekler alınarak Aleph sistemine verilecek olan veri dosyaları oluşturuldu. Sonrasında sisteme verildi ve bir model elde edildi. Bu modelde bulunan kurallardan (hipotezlerden) biri aşağıdaki gibidir:

*sense(A, B) :- has\_rel(A, first\_content\_word\_left, modifier), B=1.*

Bu kural bize verilen bir tümcede *has\_rel(A, first\_content\_word\_left, modifier)* özelliği olması durumunda “el” kelimesi için anlam karşılığının birinci anlamı (kolun bilekten parmak uçlarına kadar olan, tutmaya ve iş yapmaya yarayan bölümü) olacağını söylemektedir.

## 4. SONUÇ

Çoğu KAB tekniği, belirginleştirme için farklı sözlüksel kaynaklardan alınan bilgilerin aralarındaki ilişkiyi tespit edememektedir. Ancak TMP ilişkisel bilginin gösteriminde başarılıdır ve çeşitli kaynaklardan alınan verilerle farklı bir yapı oluşturabilmektedir. Birinci dereceden yüklem mantığını kullanarak bu veriler arasındaki bağlamsal ilişkileri kapsayacak gücü kazanır.

Sonuç olarak Türkçe için, Türkçe Sözlüksel Örnek görevini kaynak olarak kullanan ve bir TMP sisteminden faydalanan bir KAB uygulaması geliştirilebilir.

## 5. KAYNAKLAR

- [1] Weaver, W., Translation. Mimeographed, 12 pp., Reprinted in Locke, William N. and Booth, A. Donald, 1955 (Eds.), “Machine Translation of Languages”. John Wiley & Sons, New York, 1949.
- [2] Masterman, M., “The thesaurus in syntax and semantics.” Mechanical Translation, 4, 1-2, 1957.
- [3] Quillian, M. R., “A design for an understanding machine.” Communication presented at the colloquium Semantic problems in natural language. September 1961. King’s College, Cambridge University, Cambridge, United Kingdom, 1961.
- [4] Masterman, M., “Semantic message detection for machine translation, using an interlingua.” 1961 International Conference on Machine Translation

- of Languages and Applied Language Analysis, Her Majesty’s Stationery Office, London, 1962, 437- 475, 1961.
- [5] Wilks, Y. A., “On-line semantic analysis of English texts.” Mechanical Translation, 11(3-4), 59-72, 1968.
- [6] Lesk, M., “Automated SenseDisambiguation Using Machine-readable Dictionaries”, Proceedings of the SIGDOC Conference, 1986.
- [7] Wilks, Yrick, Dan Fass, Cheng-Ming Guo, James E. MacDonald, Tony Plate & Brian A. Slator., “Providing Machine Tractable Dictionary Tools”, Semantics and the Lexicon, ed. by James Pustejovsky, 341–401. Dordrecht: Kluwer Academic Publishers, 1990.
- [8] Veronis, J., Ide, N., “WSD with Very Large Neural Networks Extracted from MRDs”, Int. Conf. on Computational Linguistics, 1990.
- [9] Ide, N. and Veronis, J., “Word Sense Disambiguation: The State of the Art”, Computational Linguistics, 1998.
- [10] Muggleton, S. and De Raedt, L. “Inductive logic programming: Theory and methods.”, J. Logic Program, 1994.
- [11] Dzeroski, S., Bratko, I., “Applications of inductive logic programming”, In: De Raedt, L. (ed.). Advances in ILP. Amsterdam: IOS Press : Ohmsha, pp.65-81, 1995.
- [12] De Raedt, L., “Inductive logic programming and scientific discovery”, Technical report, Katholieke Universiteit Leuven, Leuven, Belgium, 1993.
- [13] Mooney, R.J.: “Inductive Logic Programming for Natural Language Processing”, In: Inductive Logic Programming. LNCS, vol. 1314, pp. 3–24. Springer, Heidelberg, 1997.
- [14] Specia, L., Nunes, M.G.V., Srinivasan, A., Ramakrishnan, G., “Word Sense Disambiguation using Inductive Logic Programming”, Proceedings of the 16th International Conference on ILP, Springer-Verlag, 2007.
- [15] Orhan, Z., Çelik, E., Demirgüç, N., “SemEval-2007 Task 12: Turkish Lexical Sample Task”, 2007.
- [16] Srinivasan, A.: The Aleph Manual Available at <http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/>, 1999.