

# Tümevarımlı Mantık Programlama İle Türkçe İçin Kelime Anlamı Belirginleştirme Uygulaması

Özlem Aydın<sup>1</sup>, Yılmaz Kılıçaslan<sup>2</sup>

<sup>1,2</sup> Bilgisayar Mühendisliği Bölümü, Trakya Üniversitesi, Edirne

[ozlema@trakya.edu.tr](mailto:ozlema@trakya.edu.tr), [yilmazk@trakya.edu.tr](mailto:yilmazk@trakya.edu.tr)

**Özet:** Verilen bir bağlamda birkaç farklı anlamı olan bir kelimenin hangi anlamı ile kullanıldığının belirlenmesi problemine Kelime Anlamı Belirginleştirme (KAB) denir. Doğal dil işleme uygulamalarında gereksinim duyulan önemli bir işlemdir. Tümevarımlı mantık programlama (TMP), makine öğrenmesi ve mantık programlamayı içeren bir yapay zekâ alanıdır. Birinci derecede tümcecik gösteriminde olan örnekler ve artalan bilgisinden birinci dereceden teorilerin oluşturulmasını amaçlar. TMP’de birinci dereceden mantığın kullanılmasının yarattığı zenginlik, KAB ve diğer doğal dil işleme uygulamalarında avantajlar kazandırmaktadır. Bu çalışmada tümevarımlı mantık programlama kullanılarak derlem-tabanlı bir kelime anlamı belirginleştirme uygulaması geliştirilmiştir.

**Anahtar Sözcükler:** Kelime Anlamı Belirginleştirme, Tümevarımlı Mantık Programlama, Makine Öğrenmesi, Mantık Programlama.

## Word Sense Disambiguation Application For Turkish With Inductive Logic Programming

**Abstract:** Word sense disambiguation (WSD) is the problem of determining the sense of a multi-sense word in a given context. It is one of the important processes needed for natural language processing applications. Inductive Logic Programming (ILP) is the area of artificial intelligence which contains machine learning and logic programming. It aims to build first-order theories from examples and background knowledge, which are also represented by first-order clauses. The richness of first order logic employed in ILP can hopefully provide advantages for NLP applications such as WSD. In this study, a corpus-based WSD application is improved by using ILP.

**Keywords:** Word Sense Disambiguation, Inductive Logic Programming, Machine Learning, Logic Programming.

### 1. Giriş

Kelime anlamına göre belirsizlik durumu doğal dil işleme uygulamalarının birçoğunda karşılaşılan ve çözülmesi gerekli olan önemli bir problemdir. Bu anlam belirsizliğinin giderilmesi için KAB işleminin uygulanması gerekir. Bahsedilen belirsizlik durumunu bir örnek cümle üzerinden açıklayalım:

“Yeşilden maviye dönüşen iri gözlerini bize çevirmişti.”

Bu cümlede çok anlamlı olan hedef kelimemiz “göz”dür. Bu kelimenin Türk Dil Kurumu Türkçe Sözlük’üne göre en sık kullanılan üç anlamı aşağıda verilmiştir [1]:

1. anlam: Görme organı.
2. anlam: Bazı deyimlerde görme, bakma.
3. anlam: Bakış, görüş.

Verilen cümleye KAB işlemi uygulandığında elde edilen sonuç, “göz” kelimesinin bu cümlede birinci anlamı ile kullanıldığı

olacaktır. Bu anlam belirleme süreci iki aşamada gerçekleşir. İlk aşamada, verilen bağlamdaki belirginleştirme yapılacak hedef kelime için var olan tüm anlamları belirlenir. Bu anlamların alınabileceği çeşitli elektronik sözlükler mevcuttur. İkinci aşamada ise kelimenin uygun anlamı belirlenir. Bunun için de kelimenin bulunduğu bağlam ve bazı bilgi kaynakları kullanılır [2].

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.

Bu konular içerisinde Makine Çevirisi KAB'ın en yaygın kullanıldığı alanlardan birisidir. KAB'ın önemini vurgulamak için Makine Çevirisi kapsamında bir örneği inceleyelim. Çevirisi yapılacak kaynak dili anlamada bir kelimenin birden fazla anlamı olduğu bir durum ile karşılaşıldığında KAB işlemine gerek duyulur. Makine çevirisi süreci en az iki aşamadan oluşur. İlk aşama kaynak dili anlamaya yöneliktir. İkinci aşamada hedef dilde cümleler üretilir. Çeviri işleminin doğru olarak yapılabilmesi için kaynak dilde karşılaşılabilecek kelime belirsizliği durumu çözümlenmelidir. Örneğin bir makine çevirisi sistemine içinde "kahve" kelimesinin bulunduğu bir cümleyi giriş olarak verelim. "Kahve" kelimesinin Türkçe WordNet'de [3] iki kelime anlamı karşılığı bulunmaktadır:

- 1.kahve -- Kahve, çay, ıhlamur, nargile içilen, hafif yiyecekler bulunduran, tavla, domino, bilardo, kâğıt oyunları vb. oynanan yer.
- 2.kahve -- Kavrulmuş ve çekilmiş kahve çekirdeklerinden hazırlanan içecek.

Hedef dilin İngilizce olduğunu kabul edersek "kahve" kelimesinin birinci anlamı için "cafe" kelimesi, ikinci anlamı için ise "coffee" kelimesi çeviri karşılığı olacaktır [4].

## 2. KAB'da kullanılan bilgi türleri

KAB işlemi sürecinde faydalı bazı bilgi türleri de bulunmaktadır. Bunlar kategori bilgisi (Part of speech-POS), anlamların sıklıkları (frequency of senses), eşdizimlilikler (collocations), seçimsel öncelikler (selectional preferences) ve alt-ögeleme bilgisi (subcategorization information) vb. olabilir.

- *Kategori bilgisi:* Herhangi bir kelimenin sözdizimsel sınıf bilgisidir. Bir kelimeye ait olabilecek olası anlamların sayısını azaltan faydalı bir işlemdir. Örneğin, bir kelimenin fiil olarak üç anlamı var iken, isim olarak tek anlamı olabilir.
- *Anlamların sıklıkları:* Genellikle istatistiksel yaklaşımlarda kullanılmaktadır. Bu bilgi elle etiketlenmiş derlemlerden çıkarılabilmektedir.
- *Eşdizimlilikler:* Herhangi bir kelime grubu içindeki kelimelerin arasındaki ilişkiyi verir. Şöyle ki bir kelime yanına başka bir kelime geldiğinde kelime grubu olarak başka bir anlama gelebilir. Örneğin, "kahve" kelimesi tek başına birkaç anlama geliyorken "kahve falı" kelime grubunun tek bir anlamı vardır [4].
- *Seçimsel öncelikler:* Bir kelimenin sözdizimsel ve anlamsal özelliklerini diğer bir kelime belirleyebilir. Buna

seçimsel öncelikler denir. Örneğin yemek fiili yanına canlı ya da cansız bir nesne alabilirken, öldürmek fiili sadece canlı nesne alabilir.

- *Alt-öğeleme bilgisi:* Bu bilgi kelimeler ve öbekler arasındaki belirli ilişkileri gösterir. Örneğin bir fiil yanına başka bir fiil veya isim öbeği alabilirken başka bir fiil sadece isim öbeği alabilir. Alt-öğeleme çerçeveleri tamamlayıcılar (complements) ile ilgili sözdizimsel kuralları içerir.

Verilen bu sözdizimsel ve biçimsel özelliklerin KAB çalışmalarında kullanımı daha doğru sonuçların elde edilmesi için gereklidir. Çünkü hemen hemen tüm yerel ve diğer özelliklerin KAB için faydalı olduğu bilinmektedir [5]. 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 kullanır.

### 3. KAB Yaklaşımları

KAB için günümüze kadar yapılmış olan çalışmalarda birçok yaklaşım ortaya atılmıştır. Kullanılan kaynağa göre yapılan sınıflandırmaya göre iki tür yaklaşım bulunmaktadır: bilgi-tabanlı KAB ve derlem-tabanlı KAB.

#### 3.1 Bilgi-tabanlı KAB

Bu yaklaşımda 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. Bu yaklaşımda kullanılan elektronik sözlüklerden biri makine tarafından

okunabilir sözlüklerdir (machine-readable dictionaries). Bu sözlükler kelime anlamlarının olduğu hazır bir bilgi kaynağıdır ve ilk olarak Lesk (1986) tarafından kullanılmıştır. Lesk, belirsiz kelimenin bu sözlükteki tanımı ile bulunduğu bağlamdaki diğer kelimelerin sözlük tanımları arasındaki örtüşen kelimelerin sayısını hesaplayarak belirsiz kelimenin hangi anlamının seçileceğini bulan bir metot geliştirmiştir [6]. Lesk metodu kendisinden sonra gelen makine tarafından okunabilir sözlükleri kullanan çalışmalara temel olmuştur. Banerjee ve Pedersen, Lesk algoritmasından farklı olarak Adapted Lesk algoritması (2002) geliştirmişlerdir. Bu algorithmada bir ilişki ölçümü kullanarak bir bağlamda bir kelime belirginleştirilebilir [7]. Gömü (thesauri) de bu yaklaşımda kullanılan diğer bir kaynaktır. Gömüler kelimeler arasındaki ilişki bilgisini içeren yapılardır. Bir gömüden alınan anlamsal kategori bilgisi belirginleştirmede kullanılır. KAB'da en çok kullanılan gömü Roget's International Thesaurus'dur [8].

#### 3.2 Derlem tabanlı KAB

Derlem (bütüncü, corpus (İng.)) belli prensipler çerçevesinde özel veya genel amaçlı metin ya da konuşma parça ya da bütünlerinin, üzerinde yapılacak araştırmaya uygun işaretlemelerle beraber bir araya getirilmesinden oluşan bütündür. Günümüz derlemlerinin elektronik ortamda tutularak, erişim ve kullanım kolaylığı sağlanması yaygındır. [9]. Şimdiye kadar hazırlanmış birçok derlem bulunmaktadır. 1970'li yıllarda Brown Üniversitesinde oluşturulan "Brown Corpus" 1 milyon kelime ile en geniş kapsamlı olarak etiketlenmiş metinlerden biridir. Türkçe dil çalışmalarında kullanılmak için de ODTÜ Türkçe Derlem geliştirilmiştir [10].

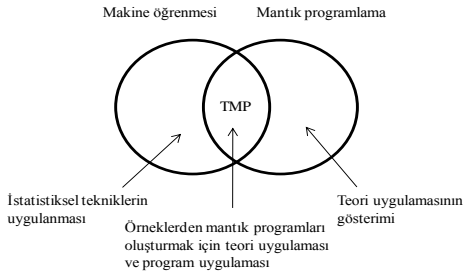
Derlem tabanlı KAB'da en başarılı yaklaşımlar derlemden sınıflayıcı veya istatistiksel modelleri öğrenmek için kullanılan istatistiksel veya makine

öğrenmesi algoritmalarıdır. Bu algoritma ve tekniklere olan ilgideki artış, makine öğrenme konusuna duyulan ilgiden kaynaklanmaktadır. Çünkü makine öğrenme algoritmaları çoğu doğal dil işleme çalışmalarında dikkat çekici bir başarı göstermiştir. Bu algoritmalar genel olarak öngörmesiz (unsupervised) ve öngörmeli (supervised) olmak üzere ikiye ayrılır. 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.

Geliştirilen uygulamada anlam belirginleştirmeyi gerçekleştirmek için makine öğrenmesi tekniği olarak tümevarımlı mantık programlama kullanılmıştır. Sonraki bölümde TMP konusu ve bu uygulamada nasıl kullanıldığı anlatılmıştır.

#### 4. Tümevarımlı Mantık Programlama

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 [11].



**Şekil 1.** Makine Öğrenmesi, Mantık Programlama ve TMP

TMP'de kullanılan mantıksal terimlerin tanımları aşağıda açıklanmıştır.

- $B$  sonlu sayıda cümlecik içeren artalan bilgisidir.
- $E$  sonlu sayıda örnek setidir =  $E^+ \cup E^-$ 
  - *Pozitif örnekler.*  $E^+ = \{e_1, e_2, \dots\}$  Boş olmayan tanımlı cümle kümesidir.
  - *Negatif örnekler.*  $E^- = \{\bar{f}_1, \bar{f}_2, \dots\}$  Horn cümlecigi kümesidir (boş olabilir).

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:

$$B = \left\{ \begin{array}{l} dede(X, Y) \leftarrow baba(X, Z), ebeveyn(Z, Y) \\ baba(ali, ayşe) \leftarrow \\ anne(ayşe, ahmet) \leftarrow \\ anne(ayşe, suna) \leftarrow \end{array} \right.$$

Dede ve torunları arasındaki ilişkiyi gösteren pozitif örnekler aşağıda verilmiştir.

$$E^+ = \left\{ \begin{array}{l} dede(ali, ahmet) \leftarrow \\ dede(ali, suna) \leftarrow \end{array} \right.$$

Ek olarak aşağıdaki negatif örneklerin de verildiğini varsayalım.

$$E^- = \left\{ \begin{array}{l} \leftarrow dede(ahmet, ali) \\ \leftarrow dede(suna, ali) \end{array} \right.$$

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

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

Burada  $H$ , elde edilen hipotezdir.

## 5. Geliştirilen Uygulama

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'de kabul edilmiştir [12].

Geliştirilen uygulamada Türkçe sözlüksel örnek görevinden alınan 10 isim, 10 fiil ile 6 sıfat ve zarf üzerinde KAB gerçekleştirilmiştir. Bu kelimelere ait anlam sayısı, eğitim ve test örnek setinde bulunan cümle sayısı ve toplam örnek cümle sayısı Tablo 1'de verilmiştir.

Kelimeler	Anlam sayısı	Eğitim boyutu	Test boyutu	Örnek boyutu
<b>İsimler</b>				
ara	7	192	63	255
baş	5	68	22	90
el	3	113	38	151
göz	3	92	27	119
kız	2	96	21	117
ön	5	72	23	95
sıra	7	85	28	113
üst	7	69	23	92
yan	5	65	31	96
yol	6	68	29	97
<b>Fiiller</b>				
al	24	963	125	1088
bak	4	207	85	292
çalış	4	103	61	164
çık	6	138	87	225

geç	11	164	90	254
gel	20	346	215	561
gir	6	163	84	247
git	13	214	120	334
gör	5	206	68	274
konuş	6	129	63	192
<b>Zarflar &amp; Sıfatlar</b>				
büyük	6	97	26	123
doğru	6	81	38	119
küçük	4	45	14	59
öyle	4	51	23	74
son	2	86	18	104
tek	2	40	10	50

**Tablo 1.** Kelimeler, anlam sayıları ve örnek boyutları

KAB uygulamamızı bir TMP sistemi olan ALEPH (A Learning Engine for Proposing Hypotheses) ile gerçekleştirdik [13]. 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 bilgisinin bulunduğu dosya

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

*Cümle:* “Yeşilden maviye dönüşen iri gözlerini bize çevirmişti.”

*Hedef kelime: göz*

**1) has\_bag() özelliği:** Anlamı belirginleştirilecek hedef kelimenin sağında ve solunda bulunan kelimeler. Bu özelliğin programdaki gösterimi aşağıdaki gibidir:

*has\_bag(cümle\_no,kelime).*

*has\_bag(snt1, iri).*

*has\_bag(snt1, çevir).*

**2) has\_narrow() özelliği:** Hedef kelimenin sağında ve solunda bulunan kelimeler ve bu kelimelerin hedef kelimeye göre konum bilgisidir.

*has\_narrow(cümle\_no, kelimenin\_konumu, kelime).*

*has\_narrow(snt1, first\_content\_word\_left, iri).*

*has\_narrow(snt1, first\_content\_word\_right, çevir).*

**3) has\_pos() özelliği:** Hedef kelimenin kategori bilgisi ile hedef kelimenin sağında ve solunda bulunan kelimelerin kategori bilgisi ve hedef kelimeye göre konum bilgileridir.

*has\_pos(cümle\_no,kelimenin\_konumu,kelimenin\_kategori\_bilgisi).*

*has\_pos(snt1, target\_word,noun).*

*has\_pos(snt1, first\_content\_word\_left,adj).*

*has\_pos(snt1, first\_content\_word\_right, verb).*

**4) has\_ont() özelliği:** Hedef kelimenin sağında ve solunda bulunan kelimelerin ontolojik düzey bilgisidir. Ontoloji düzeyi 3 olarak sınırlıdır.

*has\_ont(cümle\_no,kelimenin\_konumu,ontoloji\_düzeyi, ontoloji).*

*has\_ont(snt1,first\_content\_word\_left,1,abstraction).*

*has\_ont(snt1,first\_content\_word\_left,2,quality).*

*has\_ont(snt1,first\_content\_word\_left,3,state).*

*has\_ont(snt1,first\_content\_word\_right,1,physical\_entity).*

*has\_ont(snt1,first\_content\_word\_right,2,motion).*

*has\_ont(snt1,first\_content\_word\_right,3,motion).*

**5) has\_case() özelliği:** Hedef kelimenin hal bilgisi ile hedef kelimenin sağında ve solunda bulunan kelimelerin hal bilgileri ve hedef kelimeye göre konum bilgileri.

*has\_case(cümle\_no, kelimenin\_konumu, kelimenin\_hali).*

*has\_case(cümle1, target\_word, acc).*

*has\_case(cümle1, first\_content\_word\_left, \_).*

*has\_case(cümle1,first\_content\_word\_right, \_).*

**6) has\_rel() özelliği:** Hedef kelimenin sağında ve solunda bulunan kelimeler ile olan ilişki türü ve bu kelimelerin hedef kelimeye göre konum bilgileri.

*has\_rel(cümle,kelimenin\_konumu,ilişki\_türü).*

*has\_rel(snt1, target\_word, object).*

*has\_rel(snt1,first\_content\_word\_left, modifier).*

*has\_rel(snt1,first\_content\_word\_right, sentence).*

**7) has\_ possessor() özelliği:** Hedef kelime ile hedef kelimenin sağında ve solunda bulunan kelimelere ait sahipyici(possessor) bilgisi.

*has\_ possessor (snt1,target\_word,tr).*

*has\_ possessor(snt1,first\_content\_word\_left,fl).*

*has\_ possessor(snt1,first\_content\_word\_right, fl).*

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

### Pozitif örnek dosyasının oluşturulması

Hedef kelimenin doğru anlamını gösteren bilgi burada verilir.

*sense(cümle, hedef\_kelimenin\_anlamı).*

*sense(snt1,sense1).*

### Negatif örnek dosyasının oluşturulması

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

*sense(cümle,hedef\_kelimenin\_hatalı\_anlamı).*

*sense(snt1,sense3).*

Uygulamada her kelimenin eğitim kümesi cümlelerine ait artalan bilgisi, pozitif örnekler ve negatif örnekler alınarak ALEPH sistemine verilmiş ve bir model oluşturulmuştur. Bu modelin değerlendirilmesi aşamasında test kümesine ait artalan bilgisinin, pozitif örneklerin ve negatif örneklerin bulunduğu dosyalar oluşturulmuştur. Son olarak da bu dosyalar sisteme verilerek test işlemi yapılmıştır. Elde edilen sonuçlar aşağıdaki tabloda verilmiştir.

Kelimeler	Precision	Recall
Fiiller	0.635	0.805
İsimler	0.699	0.717
Zarflar ve Sıfatlar	0.821	0.656

**Tablo 2.** TMP ile elde edilen başarı oranları

Öğrenme sonucu elde edilen hipotezlerden biri aşağıdaki gibidir:

*sense(A,B):- true, B=1.*

Bu hipotez her ne kadar kural gibi görünüyorsa da aslında aşağıdaki gerçeğe denktir.

*sense(A,I).*

Bu kural bize verilen bir cümlede hiçbir özelliğe bakılmaksızın hedef kelime için anlam karşılığının birinci anlamı olacağını söylemektedir. Örneğin “göz” kelimesi test edildiğinde aşağıdaki tablo elde edilmektedir.

		doğru sonuç		
		+	-	
elde edilen sonuç	+	21	7	28
	-	6	20	26
		27	27	54

**Tablo 3.** “göz” kelimesi için test performansı Tabloya bakıldığında test dosyasında bulunan 27 örnek cümle için, pozitif dosyada *sense(A,I)* hipotezini gerçekleyen 21 örnek var iken diğer anlamları için 6 örnek olduğu görülmektedir. Benzer şekilde aynı hipotez negatif dosyada 7 örnek için birinci anlam değerini alırken 20 örnek için de diğer anlamlarını sağlamaktadır.

Sonuç olarak öğrenmeden sonra kelimelerin özelliklerine bağımlı olmayan bir prolog gerçeği ile ifade edilmiş bir genellemeye ulaşılmıştır. Bunun nedeni de elimizdeki kelimelerin birinci anlamının diğer anlamlara oranla daha fazla sayıda bulunmasıdır. Bu dengesizlik ALEPH sisteminin özelliklerden bağımsız olarak her kelime için “her kelime birinci anlamış taşır.” biçiminde aşırı bir genelleme yapmasıdır. Bu aşırıktan kaçınmak için daha dengeli örnek kümeleri ile çalışılmalıdır.

## 6. 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 [14].

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 [15].

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

## 7. Kaynaklar

- [1] <http://tdk.gov.tr>
- [2] Ide, N. and Veronis, J., "Word Sense Disambiguation: The State of the Art", Computational Linguistics, (1998).
- [3] Bilgin, O. Çetinoğlu, Ö., and Oflazer, K., "Morphosemantic Relations In and Across Wordnets: A Study Based on Turkish", Proceedings of the Second Global WordNet Conference (GWC 2004), January, Brno, Czech Republic, (2004).
- [4] Aydın, Ö., Tüysüz, M.A.A., Kılıçaslan Y., "Türkçe için bir Kelime Anlamı Belirginleştirme Uygulaması," XII. Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Ulusal Kongresi, Eskişehir Osmangazi Üniversitesi, Eskişehir, (2007).
- [5] Agirre, E. ve Edmonds, P., "Word Sense Disambiguation: Algorithms and Applications". Kluwer Academic Publishers, (2004).
- [6] Lesk, M., "Automated Sense Disambiguation Using Machine-readable Dictionaries", Proceedings of the SIGDOC Conference, (1986).
- [7] Banerjee, S., Pedersen T., "An adapted Lesk algorithm for word sense disambiguation using WordNet". In: Proceedings of the third international conference on intelligent text processing and computational linguistics, Mexico City, February 17–23, pp 136–145, (2002).
- [8] Chapman, R., "Roget's International Thesaurus (Fourth Edition)", Harper and Row, New York, (1977).
- [9] Kennedy, G., "An Introduction to Corpus Linguistics", Longman, (1998).
- [10] Oflazer, K., Say, B., Tur, D. Z. H., Tur, G., "Building A Turkish Treebank, Invited Chapter In Building and Exploiting Syntactically-Annotated Corpora", Anne Abeille Editor, Kluwer Academic Publishers, (2003).
- [11] Muggleton, S. and De Raedt, L. "Inductive logic programming: Theory and methods.", J. Logic Program, (1994).
- [12] Orhan, Z., Çelik, E., Demirgüç, N., "SemEval-2007 Task 12: Turkish Lexical Sample Task", (2007).
- [13] Srinivasan, A.: The Aleph Manual Available at <http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/>, 1999.
- [14] Aydın, Ö., Kılıçaslan, Y., "Tümevarımlı Mantık Programlamanın Kelime Anlamı Belirginleştirmeye Uygulanabilirliğinin İncelenmesi", IV. İletişim Teknolojileri Ulusal Sempozyumu Bildiriler Kitabı, 135-140, (2009).
- [15] 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).