**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# TÜRKÇE CÜMLELERDE İSİM TAMLAMALARININ BULUNMASI

**YÜKSEK LİSANS TEZİ**

**Kübra ADALI**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Bölümü**

**TEMMUZ 2014**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# TÜRKÇE CÜMLELERDE İSİM TAMLAMALARININ BULUNMASI

**YÜKSEK LİSANS TEZİ**

**Kübra ADALI**
**(504111519)**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Bölümü**

**Tez Danışmanı: Yrd. Doç. Dr. A. Cüneyd TANTUĞ**

**TEMMUZ 2014**

İTÜ, Fen Bilimleri Enstitüsü'nün 504111519 numaralı Yüksek Lisans Öğrencisi **Kübra ADALI**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"TÜRKÇE CÜMLELERDE İSİM TAMLAMALARININ BULUNMASI"** başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı :** **Yrd. Doc. Dr. A.Cüneyd TANTUĞ** .....
İstanbul Teknik Üniversitesi

**Jüri Üyeleri :** **Doç. Dr. Banu DİRİ** .....
Yıldız Teknik Üniversitesi

**Yrd. Doç. Dr. Gülşen ERYİĞİT** .....
İstanbul Teknik Üniversitesi

**Teslim Tarihi :** **11 Temmuz 2014**
**Savunma Tarihi :** **8 Temmuz 2014**

***Aileme,***

## ÖNSÖZ

Lisans eğitimimi tamamlayıp bilgisayar mühendisi ünvanını almamın ardından yüksek bilgisayar mühendisi almama son evre olan yüksek lisans tezimi vermek üzere olan bir bilgisayar mühendisi olarak mesleğimde çok daha fazla çalışmam gerektiğinin bilincindeyim.

Yürüttüğüm yüksek lisans tez çalışmam boyunca, benden yardımını, desteğini ve rehberliğini esirgemeyen, büyük bir özveri ile tezime danışmanlık yapan değerli hocam Yrd. Doc. Dr. A.Cüneyd TANTUĞ' a,

Tez çalışmam boyunca İstanbul Teknik Üniversitesi'nde beraber çalıştığım İ.T.Ü Doğal Dil İşleme Grubu'na,

Bugünlere gelmemde büyük emeği olan aileme teşekkür etmeyi bir borç bilirim.

Temmuz 2014 Kübra ADALI

**İÇİNDEKİLER**

**Sayfa**

## KISALTMALAR

**DDİ** **:** Doğal Dil İşleme
**KRA** **:** Koşullu Rastgele Alanlar
**DVM** **:** Destek Vektör Makinesi
**HTE** **:** Hafıza Tabanlı Etiketleme
**SMM** **:** Saklı Markov Modeli

## ÇİZELGE LİSTESİ

**Sayfa**

## ŞEKİL LİSTESİ

**Sayfa**

## TÜRKÇE CÜMLELERDE İSİM TAMLAMALARININ BULUNMASI

## ÖZET

Bu tezde Türkçe cümlelerde bulunan isim tamlamalarının sınırlarının tesbit edilmesi amaçlanmıştır. Türkçe cümlelerin içerisindeki isim tamlamarının bulunması varlık ismi tanıma, Türkçe cümlelerin ayrıştırılması, cümle anlam analizi, metin madenciliği, bilgi çıkarımı ve cümlenin bağlılık analizi vb. çalışmalara da destek verebilecek veya temek oluşturabilecek bir bölümü kapsamaktadır. Bu nedenle kullanım amacı kendi işlevinin yanında başka doğal dil işleme araçlarına da destek olabilecek bir çalışmadır.

Sistemin yapısı, temel olarak bir kural tabanlı sistem ve bir ardışık sınıflandırıcı tipi olan Koşullu Rastgele Alanlar kullanılmasına dayanmaktadır. Sistem, esas olarak bir makine öğrenmesi tekniği kullandığından dolayı ilk önce en iyi sonuçları verebilecek bir makine öğrenmesi modeli amaçlanmıştır.

Bu amaçla ilk önce test verisi belirlendikten sonra, geriye kalan verinin içerisinden en iyi sonuç verebilecek eğitim verisi seçilmiştir. Eğitim verisi oluşturulurken alışılagelmiş yöntemlerin aksine Türkçe-İngilize paralel bir derlemin kullanılmasıyla oluşturulmuştur. Paralel derlemin İngilizce tarafı bir doğal dil işleme aracı kullanılarak ayrıştırılmış, ve paralel derlem başka bir doğal dil işleme aracı kullanılarak eşleştirilmiştir.

Daha sonra ayrıştırılmış İngilizce tamlamalarının eşleştirme sonucunda Türkçe cümlelerdeki karşılıkları bulunarak içerisinde isim tamlamalarının sımırlarının belirlendiği bir eğitim verisi oluşturulmuştur. Bu oluşturulan eğitim verisinin içerisinden çeşitli parametreler kullanılarak en iyi sonuç veren cümleler seçilmiş, ve model optimize edilmiştir.

Sonuç olarak isim tamlamalarının çıkarılması amacının ilk olarak kural tabanlı bir sistem ve ardından ardışık bir sınıflandırıcı kullanlarak yapılmış ve bu ardışık sınıflandırıcının da eğitim verisinin otomatik olarak üretilmesi sağlanmıştır. Aldığımız sonuçlar ise kural tabanlı sistemden daha iyi sonuç vermektedir .

***Anahtar Kelimeler: Doğal Dil İşleme, İsim Tamlamaları, Cümle Ayrıştırılması, Makine Öğrenmesi, Koşullu Rastgele Alanlar, Paralel Derlem***

## NOUN PHRASE CHUNKING OF TURKISH SENTENCES

## SUMMARY

In this thesis, it is aimed that the detection of the bounds of the noun phrases that exists in Turkish sentences. The chunking of the noun phrases in Turkish sentences is a work which supports and/or can be a baseline system for the works of named entity recognition, parsing of Turkish sentences, sentiment analysis, text mining, information extraction, dependency parsing, text summirization, machine translation systems etc.

In fact, to understand the meaning of a sentence and comment about it or use it, the first way to analyze it is to parse it with a dependency parser or constituency parser. But in some cases that summarized information is needed from huge amoun of data such as information extraction etc the parsing of a sentence is can be unnecessarily big amount of effort and increases the rate of the error.

In these cases, the parsing which is done more shallowly can be more useful and easier to apply so shallow parsing which is called chunking is chosen for these natural languge processing modules. The most common type of shallow parsing is noun phrase chunking because it gives noun phrases that commonly contains the main words of the sentence.

There are two main works for Turkish in the past. The first work was a noun phrase chunker which has rule-based system. It uses dependency parser and rules that uses the dependencies and morhpological features. The second work was not an exactly noun phrase chunker. It was a work which finds the chunks that are constiuent of the sentence.

We used two different architecture of system. The first system depends on a rule-based system as a baseline system and the second system depends on Conditional Random Fields which is type of a sequence classifier that uses morphological features and dependency relations.

For the rule-based system, we built a human-annotated set. The half of the this set is used as development set which is used for formation of rules and the second half of the set is used as test set. The rule-based system has three main parts that are

preprocessing part, the dependecy parsing part and the rule set part.

In the first part there are five main parts that contains normalization, sentence division, word tokenization, morphological analysis and morphological disambiguation. In normalization part, the text is deasciified, vowelized, spell-checked, and passed through the other nomalization steps.

After the first part, the morphologically analyzes of the text is given to the second part that makes the dependency parses of the sentences of the text. We get the relation types and relation numbers from the second part.

In the last part, we apply some rules that uses morhpological features that is taken from the first part and dependency relations that is taken from the second part. We used some statistics of numbers of chunk labels which is obtained from the development set to produce rules. The rules also uses the relation types and relation numbers, and sentence chunks. After we finished the development of the rules we tested our test set with the whole rule-based system.

In our main system, we used a machine learning system, and the same test set that we used for the rule-based system. Because of the usage of a machine learning system, it is aimed optimize a machine learning model that gives the best results for noun phrase chunking.

With this aim, after the test set is isolated from the corpus, the sentences of the train data set which gives the best results is selected from the rest of the corpus. Instead of conventional manual annotation of training data, an automatic system which needs a Turkish – English parallel corpus is used for producing annotated data for training set. There are three different processes for this automatic production of chunked sentences which can be used as training data.

In the first part, the English side of parallel corpus is parsed and annotated second-level noun phrase chunks by an English parsing tool . Secondly, the parallel corpus is aligned word-by-word by also an natural language processing tool called GIZA++ . In the third part, the chunked noun phrases on the English side which are aligned to Turkish sentences are found in Turkish sentences and annotated as noun phrase chunks.

Conclusionly we used two different natural language processing tools for the automatic annotation of the train data. After we did the automatic chunking process, we tried to select the most accurate chunked sentences because the annotated sentences has the errors that is resulted from the natural language processing tools and the alignment mechanism.

For selecting the most accurate sentences, we used the normalized form of the scores

that are given by the natural language processing tools by the sentence lengths. The annotataed sentences which gives the best results are selected by using these normalized scores and some heuristic filters.The filtered sentences used as train set and our model is optimized with this automatically annotated train set.

After we selected the train set, we set 14 different features and we had done some set of experiments to select the optimum feature combination to train our model of conditional random fields. We have done the set of experiments that contains the experiments that selects and adds the best scoring feature to the combination at eahc step. We have done this set of experiments with the half of our training data because we have done so many experiments and we wanted to do the experiments in less time.

At the final step, we selected eleven different features and we used this optimum selected group of automatically annotated sentences with this optimum combination which contains eleven features. We trained our optimum model with conditional random fields tool and tested our model with our test set. The results that we get from the second system is hopeful.

As a result, the purpose of noun phrase chunking is done by using a sequence classifier additional to a rule-based system and the automatic production of annotated sentences for the train set is provided. Results that we obtain from the second system is much better than the first rule-based system.

Our second system has three marked and important property. The first one is the language independence. In the case of the parallel corpus that consists of the language which needs to be chunked and English exists, our method is applicable for the language.

The second property is that our method produces annotated data automatically and no need to manual annotation.The final important property of our system is that it is the first system that a machine learning system is used for noun phrase chunking and that conditional random fields is used for Turkish.

***Key Words: Natural Language Processing, Noun Phrases, Shallow Parsing, Machine Learning, Conditional Random Fields, Paralel Corpus.***

# 1. GİRİŞ

Doğal dil işleme, insanlar arasındaki temel iletişim aracı olan dilin makineler tarafından algılanması ve işlenmesini sağlayan bilim dalıdır. Dilin yazıya dökülmüş hali olan yazılı metinlerinde işlenmesi makineler tarafından anlamlı hale getirilmesi, doğal dil işleme sayesinde mümkün olmaktadır. Doğal dil işlemenin bir üst sınıfı olan yapay zeka günümüzde insanlar arasındaki kullanımı gün geçtikçe yaygınlaşan ve her geçen günde beklentinin arttığı ve gelecekte de insanların hayatlarını daha da kolaylaştıracağı öngörülen zeki makinelerin ortaya çıkış noktasıdır. Zeki makinelerin, insanlar arasındaki temel iletişim aracı olan dil ile direkt olarak çalışması, o dilin makine tarafından yorumlanabilmesi veya o dil ile tepki verebilmesi ancak doğal dil işleme ile mümkün olabilmektedir.

Doğal dil işleme, dil ile ilgili kullanıldığı amaca göre makine çevirisi, metin özetleme, bilgi çıkarımı, otomatik soru sorma vb. birçok alt işleve hizmet etmektedir. İnternet ortamının kullanıcıların kullanımına sunulmasıyla yaygınlaşan internet kullanımı, çok daha büyük çapta verilerin analizi veya filtrelenmesi için çok daha karmaşık işlemler gerektirmektedir. Bu gereksinimi karşılamak amacıyla makine çevirisi, özet çıkarımı, metin kategorileme vb. birden fazla doğal dil işleme alanı ortaya çıkmıştır.

Bunlardan biri olan bilgi çıkarımı, özellikle son yıllarda internetin kullanımının yaygınlaşması, internet ortamında gitgide büyüyen ve karmaşık bir hal alan verinin insanlar tarafından daha kolay ve etkin kullanabilmesi için filtrelenmesi ihtiyacının karşılanması konusuyla ilgilenmektedir. Bununla ilgili olarak geçmişte birçok çalışma yapılmıştır.

Dil ile ilgilenmesinin doğal bir sonucu olarak, doğal dil işleme, içerisinde dil ile ilgili alt katmanların olduğu, cümledeki en küçük yapıtaşından tüm cümlenin anlamına ve dilin nasıl kullanılıcağına kadar olan kısımları içerir (Eryiğit, 2006) .

Bu kısımlar;

- Dilin en küçük yapıtaşı olan fonemleri konu alan ses bilimi,
- Morfoloji, dildeki sözcüklerin yapısı üzerinde çalışan biçimbilimi,
- Dildeki sözcüklerin anlamlarını konu alam sözcük bilimi,
- Sözcüklerin nasıl bir araya gelerek bir cümle oluşturması gerektiği ile ilgilenen söz dizimi, sentaks,
- Sözcüklerin cümledeki anlam bilgisiyle ilgilenen semantiks,
- Dilin kullanım amacını inceleyen bilim dalı olan kullanım bilgisidir.

İnsanların iletişim aracı olarak kullandığı dilin içerisindeki bir veya birden fazla sözcükten oluşan en küçük ifade birimi cümledir. Cümlenin makineler tarafından anlaşılabilmesi, işlenmesi veya üretilebilmesi için analiz edilmesi gerekmektedir. Bunun için cümlenin, özne, nesne, tümleç ve yüklem gibi ana öğelerine ayrılması, her bir öğenin de cümlenin ana öğesi olan yükleme olan ilişkisi belirlenmelidir. Bu safhanın başarılı olabilmesi, cümlenin çözümlenmesi safhasından önceki bölümler olan morfoloji ve sözcük biriminin de başarımlarının yüksek olmasıyla doğru orantılıdır çünkü yukarıda belirtilen etapların her biri bir önceki etebın çıktısını girdi olarak kullanır. Cümlenin öğeleri belirlendikten sonra her bir öğenin taşıdığı anlam ve anlam açısından cümleye sağladığı katkının üzerinde durulur.

Bazen bilgi çıkarımı gibi sadece metnin içerisindeki konuyu veya içeriği baz alan işlemlerde cümlenin öğe yapısının detaylı bir şekilde çıkarılmasına ihtiyaç duyulmaz. Bunun yerine cümledeki isim ve fiil öbekleri kabaca belirlenir ve ilgili öbeklerin incelenmesi sonucu bir sonuca varılır. Buna yüzeysel çözümleme veya yüzeysel öbekleme denir. Yüzeysel öbekleme, cümledeki öğelerin ve bunun içerisindeki kelimelerin cümle içerisindeki öğe olarak görev bilgisini içermez sadece cümlede bir fiil soylu sözcüğün veya isim soylu sözcüğün, kendi etrafında, onu niteleyen ve / veya belirten diğer sözcüklerle beraber oluşturduğu sözcük öbeğini bulmaktır.

Bu yöntem hem bilgi çıkarımı, otomatik özetleme, bilgi elde etme gibi kümülatif olarak büyük hacimli veriden özet sonuçlar çıkarma mantığına dayalı olan işlemlerde daha çok verimli sonuçların alınmasını sağlar. Çünkü hem cümlenin öğeleri ve arasındaki bağlantıları belirlemek gibi bu çalışmalar için gereksiz denilebilecek analizlere çaba sarf edilmemiş olur, hem de cümle içerisindeki incelenmesi gereken

alan daraltılmış olur. Böylece hedef alınan konunun cümle veya tüm metin içerisindeki incelenmesi daha detaylı bir şekilde sağlanmış olur.

Bu ve benzeri amaçlara hizmet eden en uygun yüzeysel öbekleme türü ise ana kelimenin isim soylu bir sözcük olduğu isim öbeklerinin bulunmasıdır. İsim öbekleri, cümlede çeşitli görevlerde olabilir. Bir cümlede yüzeysel öbekleme ile tesbit edilen bir isim öbeği cümlenin öğe yapısı hakkında bir referans olamaz fakat cümlenin hatta o cümlenin bulunduğu metnin anlatmak istediği şey veya konusu hakkında önemli bir bilgi içerebilir.

### 1.1 Tezin Amacı

Bu tez çalışmasında Türkçe bir cümlede yer alan isim tamlamalarının bulunması amaçlanmıştır. İsim tamlamaları, cümleler içerisinde bir isim soylu sözcük olan ana (temel) sözcük ve ana sözcükten önce gelen ve onu niteleyen ve/veya belirten sözcükleri de içine alan bir sözcük grubudur. Amaçladığımız şey, cümle içerisinndeki isim soylu sözcüklerin oluşturduğu bu grupların başlangıç ve bitiş noktalarını yani sınırlarını tesbit etmektir.

Bu amaç doğrultusunda uyguladığımız yöntem için geçmiş çalışmalarda olduğu gibi kural tabanlı yönteme ve ardından ana model olarak istatistiksel yönteme başvurulmuştur. İlk olarak uyguladığımız yöntem, cümlelerin bağlılık ilişki bilgisinin üzerine kurallar uygulayarak isim tamlamalarını bulan yöntemdir. Bu yöntem, kural tabanlı ve dile özgü bir yöntemdir. İkinci olarak uyguladığımız yöntem bir makine öğrenmesi tekniği olan KRA'dır. KRA'ın bir makine öğrenmesi tekniği olması sebebiyle bir eğitim ve bir test kümesine hazırlamak gerekmiştir. Elimizde bulunan paralel derlemden test kümesi için cümleler seçilip işaretlenmiş daha sonrada bu cümleler söz konusu derlemden çıkarılacak, derlemin geriye kalan cümleleri ise eğitim verisi için kullanılmak üzere ayrılması planlanmıştır.

Eğitim verisi oluşturulurken el ile işaretleme yapılmayacaktır. Bunun yerine paralel derlemin avatajı kullanılarak otomatik işaretlenmiş veri elde edilecektir. Bu veriyi iyileştirme, ve kalitesini artırma işlemleri yapıldıktan sonra enson eğitim verisi üzerinden özellik seçimi optimize edilmesi ve buna ek olarak özellik kombinasyonlarının da optimize edilmesi planlanmıştır.

Son olarak optimize edilmiş veri üzerinde optimize edilmiş özellik kombinasyonu

denenerek en yüksek sonuçlar bulunması amaçlanmıştır. Yöntem, hem istatistiksel yöntem derlemi bulunan her iki dil için uygulanabilirlik açısından yani tek bir dil için değil de birden fazla dil için uygulanabilirlik açısından önemlidir. Yöntemin bir diğer önemli özelliği ise yine eğitim verisi için elle işaretleme işlemini otomatize etmiş olmasıdır. Yöntemin tek dezavantajı ise yöntemi uygulanacağı dilde İngilizce ile paralel derleme ihtiyaç duymasıdır. Elde edilen sonuçlar hem ilk kural tabanlı sistemden daha iyi bir seviyede ve umut vericidir.

### 1.2 Literatür Araştırması

Yüzeysel öbekleme ile ilgili çalışmalar çok eskilere dayanmaktadır. İngilizce ve Türkçe dahil yapılan çalışmalar daha birçok dilde mevcuttur. Church(1988), çalışmasında stokastik bir model kullanarak isim tamlaması bulan basit bir uygulama yapmıştır. Chen(1993), çalışmalarında bi-gram dil modeli kullanarak eğittikleri sistemi kullanmışlar, ve başarımları %99 luk bir doğru öbek oranı elde etmişlerdir. Bu çalışma İngilizce dili için yapılmıştır. Kermes(2002), yaptığı çalışmada Almanca için dilbilgisi kurallarına dayanan kural tabanlı bir sistem kullanmıştır. Bu sistemden elde ettiği başarı, %78,90 lık bir doğruluk oranıdır.

Singh(2002), Saklı Markov Modeli kullanarak kural tabanlı sistem yerine istatistiksel bir yöntem kullanmıştır. Hindi dili üzerinde yapılan ve birden fazla öbekleme çeşidi kullanılan bu çalışmada elde edilen başarı % 92,02 dir.

Vuckovic(2008), Hırvatça üzerine yaptığı çalışmada yine lokal dil olan Hırvatça ' ya ait dilbilgisi kurallarını kullanarak kural tabanlı bir sistem kullanmıştır. Bu sistem diğerlerinden, isim öbeklerinin yanında diğer öbekleri de bulması açısından farklıdır. Elde ettikleri başarımlar, isim öbekleri için %92,31, fiile öbekleri için %97,01 ve edat öbekleri için %83,08 dir.

Bir diğer çalışma olan Dhanalakshmi(2009), çalışmasında kural tabanlı sistemler yerine üç değişik makine yöntemi olan Destek Vektör Makineleri, Hafıza Tabanlı Öğrenme, ve KRA 'ı kullanmış, bunların içerisinden en çok başarı elde eden yöntem ise %97,49 doğruluk oranı ile KRA yöntemi en iyi sonucu vermiştir. Bu çalışma üzerinde çalışıldığı dil olan Tamil ' in Türkçe gibi sondan eklemeli dil olması ve sadece isim öbeklerinden ziyade 9 farklı öbek çeşidi üzerinde çalışması açısından önemlidir.

Bir bükünlü dil olan Lehçe üzerinde bir çalışma olan Radziszewski(2010)'ın çalışmasında karar ağaçları kullanılmıştır. Fakat bu çalışmada göze çarpan özellik, modelin eğitilmesi için verilen özellikler bir takım kalıplar kullanılmıştır. Çalışmanın başarımı %98'dir.

Almanca üzerine başka bir çalışma da Atterer(2009)'un artırımlı bir yapı kullandığı çalışmadır. Başarı oranı ise %81,5'tir.

Asahara(2003), Çince üzerine yapmış olduğu çalışmada DVM kullanmıştır. Bu çalışma normalde kelimelerden öbek değil aslında karakter öbeklerinden kelimenin sınırlarını bıulmayı amaçlamıştır. Başarı oranı % 94,4 tür.

Ramshaw ve Marcus(1995) çalışmasında DBÖ(Transformation Based Learning) ve bununla birlikte sözlük bilgisi de kullanmışlardır. Tek katmanlı isim kumesi öbeklemenin yapıldığı bu çalışmanın başarımı %92 dir.

Cardie ve Pierce(1998) sözcük tipi etiketlerinin sıralandırılmasından kaynaklanan kalıpların makine öğrenmesi ile öğrenilerek cümledeki isim öbeklerinin bulunmasını amaçlamıştır. F-Skoru %90,9'dur.

Voutilainen(1993) yaptığı çalışmada en büyük uzunluktaki isim öbeğini bulmak için bir DDİ aracını kullanıma sunmuştur. Bu çalışmada dilbilgisi ve kelime bilgisi bir arada kullanılmıştır. Bulunan doğru olanının %98,5-%100 aralığında olmasına rağmen doğruluk oranı % 95- %98 aralığındadır.

Türkçe için yapılmış çalışmalara baktığımızda, ilk olarak bir tez çalışması olan Kutlu(2010) çalışmasını görüyoruz. Bu çalışmada birebir isim öbeklerinin bulunması amaçlanmıştır. Bunun için de kural tabanlı bir sistem kullanılmıştır. Sistemde ilk önce kelimeleri etiketleme, sonra morfolojik belirsizlik giderme daha sonra morfolojik olarak analiz edilen kelimeler arasındaki cümle içerisinde ki bağlılık ayrıştırması ile çözümlendikten sonra enson olarak bu çözümlemenin üzerine çeşitli kurallar kullanarak cümledeki isim öbeklerinin çıkarılması sağlanmıştır. Elde edilen başarım % 86,2'dir.

Türkçe üzerinde bir diğer yüzeysel öbekleme çalışması ise El-Kahlout ve Akın(2013) ın yapmış oldukları çalışmadır. Bu çalışmada isim öbeklerinin bulunmasından ziyade yüklemi teşgil eden öbeğin fiil öbeği, diğer kalan cümle öğelerinin direk tek bir tür öbek olarak sınırlarının çizilmesi söz konusudur. Amacı açısından cümlenin öğelerini öbek olarak ayırmayı baz alsa da uygulamanın sonucu olarak ister istemez isim

öbeklerini de çıkaracağı için işlev olarak isim öbeklerinin bulunmasını kapsamakta fakat özel olarak isim öbeklerini ayıramamaktadır. Yöntem olarak KRA kullanılmıştır. Bu yöntemi uygulamak için ihtiyaç duydukları eğitim verisini el ile işaretleme suretiyle elde etmişlerdir. Maximum yarar sağlayan özellikleri kullanarak eğitim verisini eğiterek bir model oluşturulmuş ve elde edilen sonuçlar için tüm genel öbekler için %87,50'dir.

### 1.3 Teze Genel Bakış

Tez temel olarak altı ana bölümden oluşmaktadır:

1. Bölümde tez hakkında genel bilgi, tezin amacı ve literatür araştırması hakkında bilgi verilecek.

2. Bölümde Türkçe' nin özellikleri, isim öbekleri ve çeşitleri anlatılacak

3. Bölümde kullanılan Yöntem olan KRA 'ın nasıl kullanılacağı anlatılacak,

4. Bölümde Sistemin ana yapısı ve bileşenleri detaylı olarak anlatılacak,

5. Bölümde Yapılan testler ve sonuçları üzerinde durulacak,

6. Bölümde ise test sonuçlarından elde edilen çıkarımlar sunulacak, ve gelecekte nelerin yapılabileceği konuşulacak.

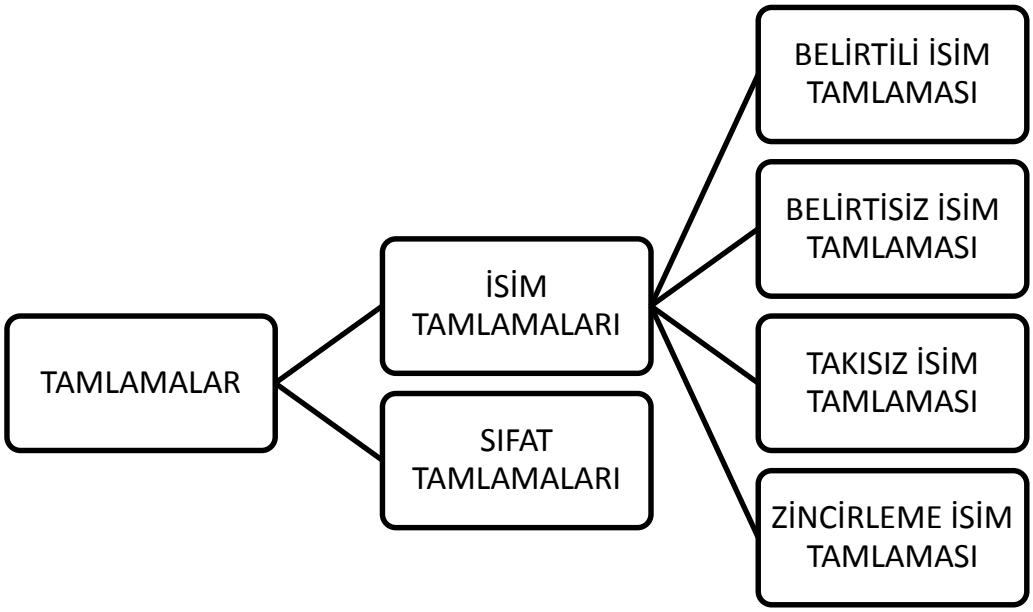
## 2. TAMLAMALAR

Türkçe‘de tamlamalar, isim öbeklerinin Türkçe dilbilgisindeki adıdır. İki veya daha fazla sözcükten oluşan birbirlerini değişik ilgilerle tamamladığı veya nitelediği söz gruplarıdır. Türkçe’ nin sondan eklemeli bir dil olması ve anlatımca zengin bir dil oluşu, isim öbeklerinin geniş yer tutmasına ve hatta systematize edilmesine ihityaç duyulmasına sebep olmuştur. Dilimizde tamlamaların hiyerarşisine bakacak olursak Türkçede iki çeşit tamlama bulunmaktadır:

Sıfat tamlaması

Ad tamlaması

Sifat tamlaması isim soylu bir sözcüğün bir veya birden fazla sözcükle nitelendirildiği, isim tamlamaları ise isim soylu bir sözcüğün bir veya birden fazla sözcükle belirtildiği tamlamalardır. İsim tamlamaları çeşitleri Şekil 2.1’deki gibidir.




**Şekil 2.1 :** Türkçe’de tamlama çeşitleri.

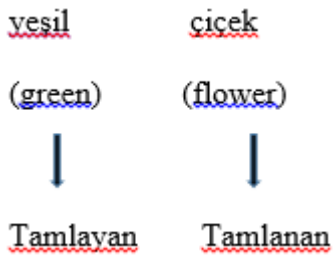
Tamlamalarda tanımındanda anlaşılacağı gibi iki ana kısım bulunur:

- Tamlanan: tamlamanın ana sözcüğü veya söz öbeğidir. İçinde bulunduğu tamlamanın asıl amacı tamlananı belirtmek veya nitelemektir.

- Tamlayan: tamlananı belirten veya niteleyen kısımdır.

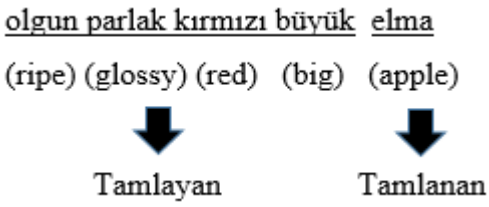
### 2.1 Sıfat tamlamaları

Sıfat tamlamaları bir veya birden fazla isim soylu sözcük grubunun oluşturduğu ifadenin bir veya birden fazla isim soylu sözcük tarafından nitelendirildiği tamlamalardır. Sıfat tamlamaları basit bir kelime sıfat, diğer kelime isim olarak iki kelimeden oluşabilir. İlgili örnek Şekil 2.2'deki gibidir:



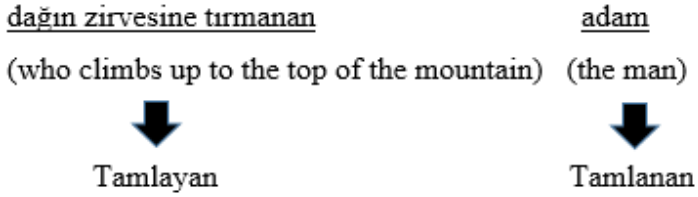

**Şekil 2.2 :** En basit yapıdaki sıfat tamlaması.

İki kelimeden daha fazla sayıda kelimeye sahip sıfat tamlamaları da çok sık kullanılmaktadır. Özellikle Türkçe' nin sıfatların isimden önce gelen dil yapısını da göz önünde bulundurursak, bir sıfat tamlamasının tamlayan kısmı sonsuz sayıda sıfat veya kelime içerebilir. Biraz daha uzun bir sıfat tamlamasına örnek Şekil 2.3'deki gibidir.



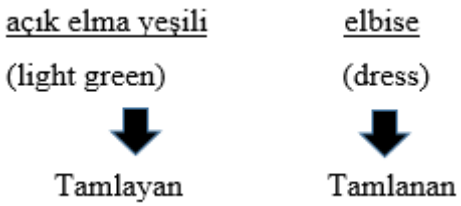

**Şekil 2.3 :** Birden fazla kelimeden oluşan tamlayanı bulunan sıfat tamlaması.

Sıfat tamlamalarının tamlayan kısmı sıfat- fiiller sayesinde ardışık birden fazla sıfat grubu yerine bir ifade ile de Şekil 2.4'deki gibi oluşturulabilir:




**Şekil 2.4 :** Birden fazla kelimeden oluşan tamlayanı sıfat-fiil olan s.t.

Bunların yanısıra bir de tamlayanı toplamda bir niteleyici olarak görev yapan ama kendi içerisinde bir çeşit isim tamlaması olan sıfat tamlamaları da Şekil 2.5 ‘teki gibi mevcuttur:




**Şekil 2.5 :** Birden fazla kelimeden, tamlayanı isim tamlaması olan sıfat tamlaması.

Yukarıdaki örneklerde de görüldüğü gibi sıfat tamlamalarındaki tamlayan ve tamlanan kısmının boyut ve şekil açısından isim soylu sözcükler olması kaydı ve tamlayanın tamlananı nitelemesi dışında bir sınırı yoktur. Örnekler bu iki şart dahilinde sonsuz sayıda genişletilebilir.

## 2.2 İsim Tamlamaları

Tamlayanın tamlananı herhangi bir şekilde belirttiği birden fazla söz öbeğine isim tamlaması denir. İsim tamlamalarında tamlayan kısımın son kelimesinin aldığı tamlayan ile tamlananın arasındaki ilişkiyi kuvvetlendiren eke tamlayan eki, tamlanan kısımda ise ana kelimenin aldığı iyelik ekine tamlanan eki denir. İsim tamlamaları tamlayan ve tamlananın aldığı eklere göre dört ana çeşide ayrılır:

Belirtili İsim Tamlaması

Belirtisiz İsim Tamlaması

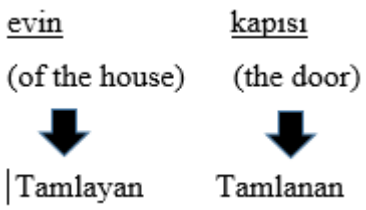
Takısız İsim Tamlaması

Zincirleme İsim Tamlaması

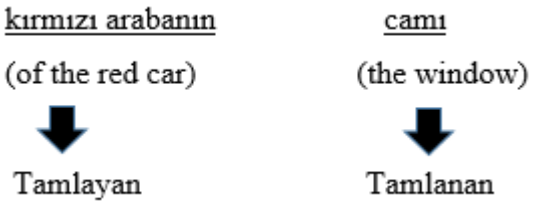
Şekil 2.1’de de Türkçe’nin tamlama hiyerarşisi görülmektedir.

İsim tamlamaları tıpkı sıfat tamlamaları gibi çok geniş bir alana sahiptir. Tamlayanın bir kelime olan tamlamaların yanısıra tamlayanın bir sıfat tamlaması olduğu, tamlayanın isim fiiller sayesinde bir ifade içerdiği, tamlayanın da kendi içerisinde bir

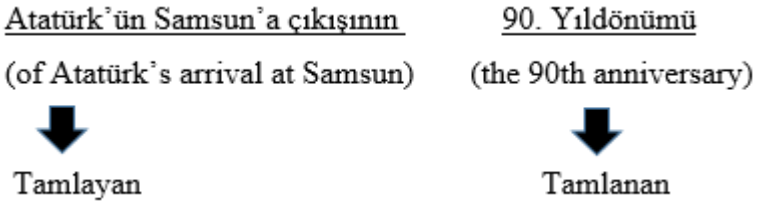
isim tamlaması olduğu vb gibi kategoriler genişletilebilir. Bu kategorilere ait örnekler Şekil 2.6, Şekil 2.7, Şekil 2.8, Şekil 2.9 'daki gibidir:



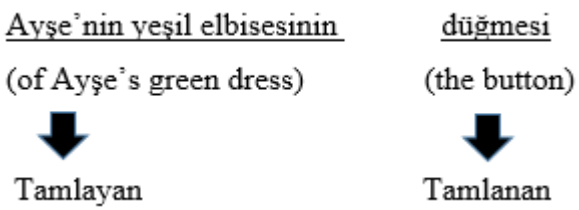

**Şekil 2.6 :** En basit yapıdaki isim tamlaması




**Şekil 2.7 :** Tamlayanı bir sıfat tamlaması olan isim tamlaması.




**Şekil 2.8 :** İsim- fiil sayesinde tamlayanın bir ifade halindeki isim tamlaması

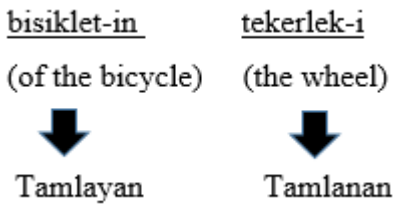



**Şekil 2.9 :** Tamlayanın da ayrı bir isim tamlaması olduğu isim tamlaması

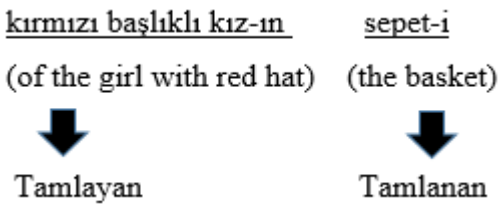
Örneklerde de görüldüğü gibi isim tamlamaları da tıpkı sıfat tamlamaları gibi tamlayanın ve tamlananın isim soylu sözcüklerden oluşması ve tamlayanın tamlananı belirtmesi koşuluyla sınırsız şekilde tamlama elde edilebilir. Bu sonuç Türkçe' nin sondan eklemeli yapısının bir sonucu olan zenginliğinin ortaya çıkardığı bir durumdur.

### 2.2.1 Belirtili isim tamlaması

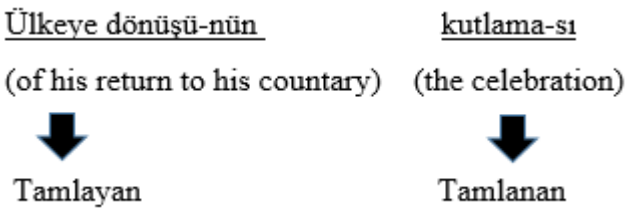
Tamlayanın ve tamlananın tamlama ekini aldığı isim tamlamalarıdır. Her iki öğeninde ek alması durumu anlam olarak tamlamayı daha net ve duru hale getirmektedir. Tamlama ekleriyle beraber çeşitli örnekler Şekil 2.10, Şekil 2.11, Şekil 2.12, ve Şekil 2.13 ‘ teki gibidir:



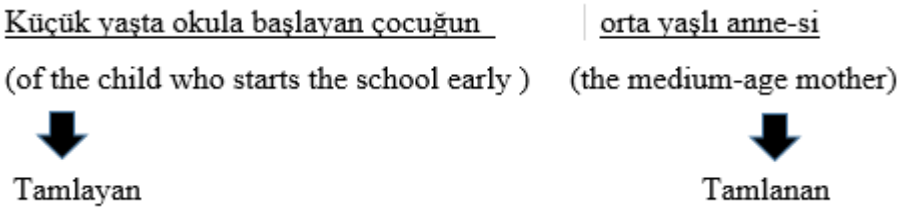

**Şekil 2.10 :** En basit yapıdaki belirtili isim tamlaması




**Şekil 2.11 :** Tamlayanı sıfat tamlaması olan belirtili isim tamlaması




**Şekil 2.12 :** Tamlayanın isim-fiil olduğu bir belirtili isim tamlaması

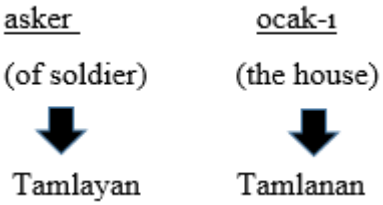



**Şekil 2.13 :** Tamlayanın ve tamlananın sıfat tamlaması olduğu isim tamlaması

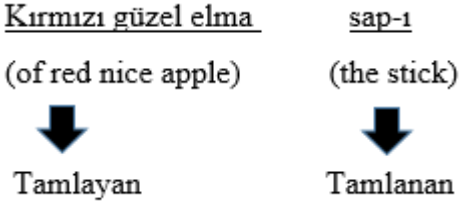
### 2.2.2 Belirtisiz isim tamlaması

Belirtisiz isim tamlamalarında, tamlayan, tamlayan eki almaz, sadece tamlanan,

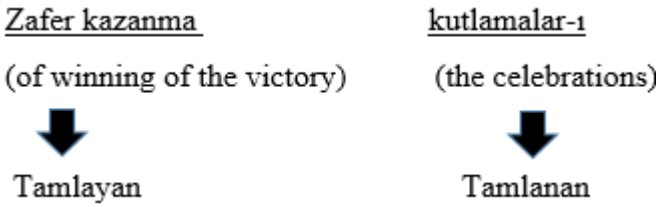
tamlanan eki alır. Anlamda tamlayana aitlik vugusu belirtili isim tamlamasına göre daha azdır. Tamlayan ek almadığı için çok kullanılan belirtisiz isim tamlamalarının zamanla bileşik isme dönüştüğünün örnekleri fazla sayıda mevcuttur. Belirtisiz isim tamlamalarına ait bir takım örnekler Şekil 2.14, Şekil 2.15, Şekil 2.16, ve Şekil 2.17 deki gibidir:



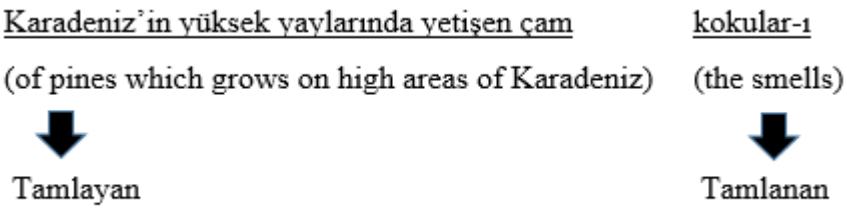

**Şekil 2.14 :** En basit belirtisiz isim tamlaması




**Şekil 2.15 :** Tamlayanı sıfat tamlaması olan belirtisiz isim tamlaması




**Şekil 2.16 :** Tamlayanı isim-fiil olan belirtisiz isim tamlaması



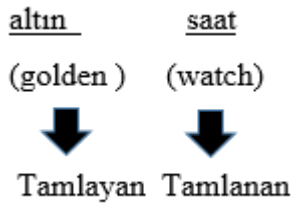

**Şekil 2.17 :** Tamlayanı sıfat tamlaması olan belirtisiz isim tamlaması

### 2.2.3 Takısız isim tamlaması

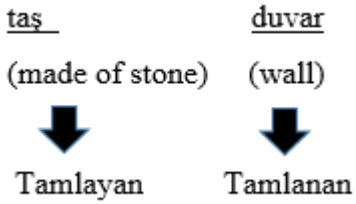
Bu isim tamlaması türünde tamlayan ve tamlanan tamlama eki almaz. En basit ve sade isim tamlaması türüdür. Çoğu zaman sıfat tamlamalarıyla ayırt edilemez ve bu konuda belirsizlik taşırlar

Bu belirsizlik eğer tamlayan tamlananın neden yapıldığını belirtiyorsa takısız isim tamlaması belirtmiyorsa sıfat tamlaması olarak kabul edilir. Veya –den eki tamlayana getirildiğinde anlam bozulmuyorsa veya tamlayan ile tamlanan arasına "gibi" edatı getirildiğinde anlam bozulmuyorsa bu tamlama takısız isim tamlaması olarak kabul edilir. Ek almadığı için diğer isim tamlamalarında olduğu gibi yaygın bir kullanım alanı yoktur, sadece bir şeyin neyden yapıldığını anlatmak için kullanılır.

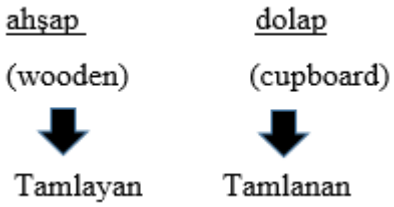
Takısız isim tamlamarıyla alakalı olarak örnekler Şekil 2.18, Şekil 2.19, Şekil 2.20, ve Şekil 2.21'deki gibidir:



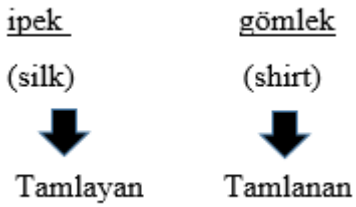

**Şekil 2.18 :** Basit bir takısız isim tamlaması




**Şekil 2.19 :** Bir diğer takısız isim tamlaması




**Şekil 2.20 :** Bir diğer takısız isim tamlaması

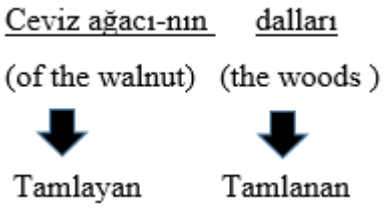



**Şekil 2.21 :** Bir diğer takısız isim tamlaması

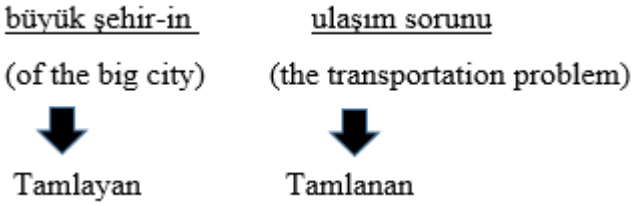
### 2.2.4 Zincirleme isim tamlaması

Bu isim tamlaması çeşidi ise isim tamlamalarının içerisinde en karmaşık ve büyük çaplı olanıdır. En az üç kelimeden oluşur. Bunun sonucu olarak da iki isim tamlamasının içiçe geçmiş hali zincirleme isim tamlaması olarak kabul edilir. İçerisinde sıfat bulunan herhangi bir belirtili veya belirtisiz isim tamlaması, zincirleme isim tamlaması olarak kâbul edilmez.

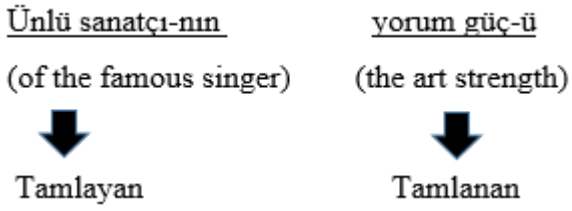
Zincirleme isim tamlaması örnekleri Şekil 2.22, Şekil .2.23, Şekil 2.24 ve Şekil 2.25‘ deki gibidir:



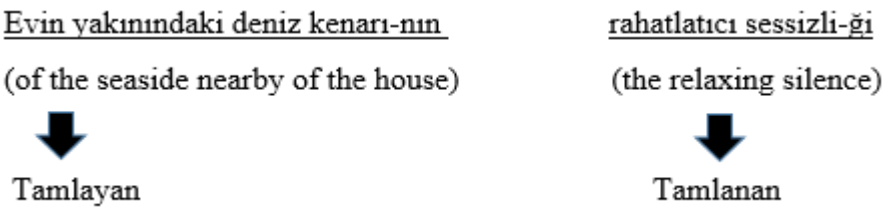

**Şekil 2.22 :** Basit bir zincirleme isim tamlaması




**Şekil 2.23 :** Bir sıfat tamlaması + bir isim tamlaması olan isim tamlaması




**Şekil 2.24 :** İki tamlamadan oluşan bir diğer örnek.




**Şekil 2.25 :** Daha karmaşık bir diğer örnek

Zincirleme isim tamlamaları ilk bakışta, aralara sıfat girmiş bir belirtili veya belirtisiz isim tamlamasına benzeyebilir fakat bu ikisini ayırmanın en önemli ve tek bir yolu belirtisiz veya belirtili isim tamlamalarında tamlayan veya tamlanan hiçbir zaman bir isim tamlaması oluşturamaz fakat zincirleme isim tamlamalarında tamlayan veya tamlanan dan en az biri belirtili, belirtisiz veya takısız isim tamlamasıdır.

## 3. KOŞULLU RASTGELE ALANLAR

KRA, Lafferty ve arkadaşları (Lafferty**,** 2001**)** tarafından önerilen istatistiksel dizilim sınıflandırmasına dayanan bir makine öğrenmesi yöntemidir (Kazkılınç, 2013). Dizilim içerisindeki her birim bir etikete sahiptir. Ve her bir birime atanmaya çalışılan aday etiketlerin olasılıkları hesaplanır. Aday etiketler arasında olasılık dağılımı hesaplandıktan sonra en çok olasılık değerine sahip olan etiket birime verilir.

Bu mantıkla, KRA modelini P ( y*| x*) olasılık hesabı ve bu hesabın maximize edilmesi üzerine kurulu olan bir model olarak kabul edersek, X* = $x_1$, $x_2$, $x_3$, …, $x_n$

sisteme girdi olarak verilen dizilimi, Y* = $y_1$, $y_2$, $y_3$, $y_4$, $y_5$,…, $y_n$ sistemden alınan her bir birime ait çıktı etiketlerini temsil eder.

Buna göre model bağıntı 1 ile formulize edilebilir:

$$p_\theta\,(y|x) = \frac{1}{Z_\theta(x)} \exp\left\{\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k\, f_k\,(y_{t-1}, y_t, x_t)\right\} \qquad (1)$$

Burada ZΘ (x) tüm olası etiket dizileri için normalleştirme faktörüdür (Kazkılınç 2013). Bu faktör bağıntı 2'deki gibi formulize edilebilir:

$$Z_\theta\ (x) = \sum_{y \epsilon Y^T} \exp\left\{\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t\right\} \qquad (2)$$

Yukarıdaki formülde görüleceği üzere $y_{t-1}$, $y_t$ ve $x_t$ nitelik fonksiyonunun değerini değiştiren parametrelerdir. $y_{t-1}$ bir önceki girdinin etiketi, $y_t$ sıradaki girdinin etiketi ve $x_t$ ise sıradaki girdinin kendisidir. Nitelik fonksiyonları, makine öğrenmesinde kullanılmak istenen nitelikleri belirleyen fonksiyonlardır.

KRA, bir dizilim sınıflandırıcı olarak kullanıldığı için doğal dil işleme ile ilgili uygulamalarda diğer yöntemlere nazaran çok daha iyi sonuçlar vermektedir (Dhanalakshmi, 2009). Ayrıca KRA, MEMMs in de sağladığı avantajları bünyesinde taşır ve etiketleme de meydana gelen sistematik yanılgı problemini ortadan kaldıran bir yöntemdir. Bunun yanı sıra Saklı Markov Modeli ve stokastik gramerlerin dezavantajlarınında üstesinden gelmiş bir yöntemdir (Dhanalakshmi, 2009).

### 3.1 KRA Aracının Kullanımı

KRA, CRF++ isminde, açık kaynak kodlu, kullanımı basit ve çok çeşitli uygulamalara uyarlanabilir bir araç aracılığıyla kullanılmıştır. Ardışıklık arz eden veriler üzerinde bölümleme veya etiketleme işlemi amacıyla kullanılan bu araç, bilgi çıkarımı, öbekleme ve varlık ismi tanıma gibi birçok doğal dil işleme uygulamalarında kullanılmıştır.

CRF++ aracının yazılım paketi (Kudo,2013) her makine öğrenmesi tekniğinde olduğu gibi iki kısımdan oluşmaktadır:

CRFLearn: Hazır işaretlenmiş eğitim verisini alarak modelin eğitilme işlemini gerçekleştirir.

CRFTest: Modeli ve işaretlemeye hazır olan veriyi alarak işaretlenmiş halini çıktı olarak üretir.

Bu iki kısımdan birinci kısım, veriyi her bir satırda bir eleman, ve her bir kolonda bir nitelik ve enson kolonda ise sınıf etiketi olması şartıyla alarak modeli eğitir. Her iki cümlenin arasına istenirse bir boş eleman konulabilir. Şekil 3.1'de eğitime hazır veri kümesinden bir kesit verilmiştir:

```
Sincaplar       sincap  Noun    false   false   B
çabuk   çabuk   Adj     false   false   O
hareket hareket Noun    false   false   O
ederler et      Verb    false   false   O
.               .       Punc    false   false   O
```

**Şekil 3.1 :** Araç tarafından eğitilmeye hazır bir eğitim verisi

Eğitilmek için uygun formata getirilen eğitim verisinin yanısıra CRF++ aracının bize

sağladığı avantajlardan biri olan ve bize eğitim verisindeki kolonlarda bulunan özelliklerin değişik kombinasyonlarını kullanma fırsatını veren nitelik şablon dosyasını kullanır. Bu dosyada sırasıyla kombinasyonlar alt alta yazılır. Her bir kombinasyonun içinde hangi özelliklerin olacağı [pozisyon numarası,kolon numarası] formatında belirtilir. Pozisyon önceki, sonraki veya mevcut elemanın pozisyonunu, kolon numarası ise eğitim verisindeki hangi özelliği seçtiğini belirler. Örneğin şablonda [0,3] yazması demek mevcut elemanın 3. Kolonundaki özelliğinin alınması demektir.[-1,3] ise bir önceki elemanın 3. Kolonundaki özelliğinin alınması demektir. İki veya daha fazla özelliğin bir kombinasyon şeklinde belirtilmesi ise ,

```
[pozisyon numarası, kolon numarası] /%x[pozisyon numarası, kolon numarası]
```

biçiminde belirlenir. Şekil 3.2'de nitelik kombinasyon şablonu dosyasından bir kesit verilmektedir:

```
U18:%x[-1,2]
U19:%x[-1,3]
U20:%x[-1,4]
U21:%x[-2,0]
U22:%x[-2,1]
U23:%x[-2,2]
U24:%x[-2,3]
U25:%x[-2,4]

# feature combinations for the current position
U26:%x[0,0]/%x[0,1]/%x[0,2]/%x[0,3]/%x[0,4]
U27:%x[1,3]/%x[0,3]/%x[-1,3]
U28:%x[1,4]/%x[0,4]/%x[-1,4]
U29:%x[1,1]/%x[0,1]/%x[-1,1]
U30:%x[1,2]/%x[0,2]/%x[-1,2]
U31:%x[2,2]/%x[1,2]/%x[0,2]/%x[-1,2]/%x[-2,2]
U32:%x[2,3]/%x[1,3]/%x[0,3]/%x[-1,3]/%x[-2,3]
U33:%x[2,4]%x[1,4]/%x[0,4]/%x[-1,4]/%x[-2,4]
U34:%x[2,3]/%x[1,4]/%x[0,4]
U35:%x[1,3]/%x[0,4]/%x[-1,4]
U36:%x[1,3]/%x[0,3]/%x[-1,4]
U37:%x[2,3]/%x[1,3]/%x[0,4]
```

**Şekil 3.2 :** Aracın kullanacağı nitelik sablonu dosyasından bir kesit.

Şekil 3.2'deki kesittende görüldüğü gibi her bir satırda bir nitelik kombinasyonu bulunmaktadır. "U" ile başlayan kısım nitelik kombinasyon numarası diğer

kısımlarda yukarıda da anlatıldığı gibi [pos,kolon] formatında kodlanan niteliklerin uni-gram, bi-gram, tri-gram vb. şeklinde belirtilen kombinasyonlarıdır. Sonuç olarak CRF++'nin eğitim kısmı yukarıda belirtilen uygun formata getirilmiş eğitim verisini ve nitelik kombinasyon şablon dosyasını kullanarak veriyi eğitir ve bir model üretir.

CRFTest ise CRF++'ın sonuç üretme kısmıdır. Bu kısım için üç tane bileşen gereklidir.

Birincisi, test verisinin uygun formata getirilmiş halidir. Bu format mutlaka eğitim verisi için hazırladığımız formatın sadece en son kolonu olan sınıf etiketlerinin bulunduğu kolonun çıkarılmış halidir. Şekil 3.3'de yukarıda eğitim verisi için olan formatın test verisi için hazırlanması gereken format versiyonu verilmiştir:

```
Sincaplar       sincap  Noun    false   false
çabuk   çabuk   Adj     false   false
hareket hareket Noun    false   false
ederler et      Verb    false   false
.               .       Punc    false   false
```

**Şekil 3.3 :** Araç tarafından kullanılacak olan test verisi.

Şekil 3.3'de görüldüğü üzere eğitim verisinden sadece bir kolonu, yani sınıf etiketlerinin bulunduğu eksiktir.

Bileşenlerin ikincisi ise, aynı modelin eğitiminde kullanıldığı nitelik kombinasyon şablonu dosyasıdır. Modelin eğitimi esnasında kullanılan birebir aynısı olmak zorundadır.

Bileşenlerin üçüncüsü ve en önemlisi model dosyasıdır. Eğitim sonucunda oluşan model dosyası kullanılır. Üretilen sonuç, yine eğitim verisi formatında en son kolonda sınıf etiketleri bulunmak koşuluyla çıktı olarak verilir.

Bu aracın en iyi n tane sonucu üretme opsiyonu bulunmaktadır. Eğer üretilen sonuç sayısı parametresi (n) birden daha fazla verilirse araç her bir eleman için en iyi olasılık değerine sahip n tane sonuç üretir.

```
# 0 0.107240
Borissov  Borissov  Prop      false     false     B
,         ,         Puncfalse     false     O
bunun     bu        Pron false     true      O
ülke      ülke      Noun   false     false     O
üzerinde  üzer      Noun   true      false     O
yıkıcı    yık       Adj    false     false     O
bir       bir       Det    false     false     O
etki      etki      Noun   false     false     O
yapacağını  yap     Noun   true      false     O
söyledi   söyle     Verb   false     false     O
.         .         Puncfalse     false     O

# 1 0.085476
Borissov  Borissov  Prop   false     false     B
,         ,         Punc   false     false     O
bunun     bu        Pron   false     true      O
ülke      ülke      Noun   false     false     B
üzerinde  üzer      Noun   true      false     I
yıkıcı    yık       Adj    false     false     O
bir       bir       Det    false     false     O
etki      etki      Noun   false     false     O
yapacağını  yap     Noun   true      false     O
söyledi   söyle     Verb   false     false     O
.         .         Punc   false     false     O
```

**Şekil 3.4 :** Araç tarafından üretilen çıktı verisi.

Şekil 3.4'te verilen çıktıdan bir kesit bulunmaktadır: Bu kesitte n 2'dir En üst kısımda birer satır halinde görülen her grubun başında kaçıncı iyi sonuç olduğu ve

olasılık değeri verilir.

# 4. CÜMLE YAPISI VE BAĞLILIK ANALİZİ

Cümle, dili ifade eden en küçük yapı birimidir. Cümlenin makineler tarafından yorumlanabilmesi için çözümlenmesi ve analiz edilmesi gerekmektedir. Cümle Türkçe dilbilgisinin tanımlamasına göre Türkçe bir cümle, dört ana öğeden oluşabilmektedir:

- Yüklem
- Özne
- Tümleç
- Nesne

Yüklem, cümlede iş, oluş veya hareket bildiren öğedir.Cümlenin ana öğesidir. Yüklemsiz bir cümle düşünülemez.

Özne, cümlede yüklem vasıtasıyla belirtilen oluş veya eylemin sahibidir.

Tümleç, yüklemi niteleyen veya belirten öğelerdir.

Nesne, cümlede yüklemin belirttiği eylemden etkileneni belirten kısımdır.

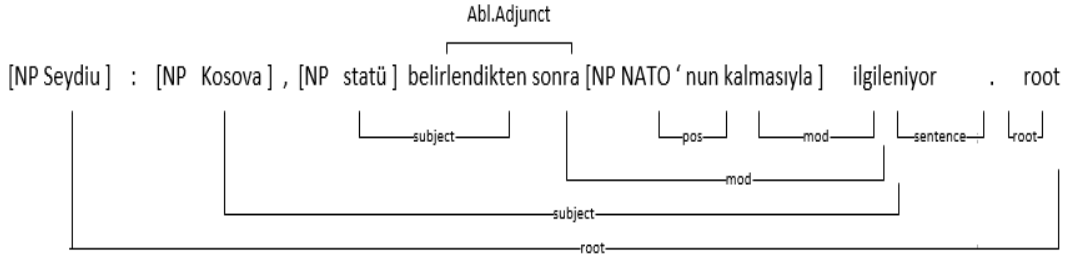
“Ali, annesinin verdiği parayı, marketten ekmek almak için kullandı.” Cümlesinde “Ali” özne, “annesinin verdiği parayı” nesne, “marketten almak için” tümleç, “kullandı ” ise yüklemdir.

Cümlenin büyüklüğü arttıkça, cümle içerisinde olan öğe yapısı ve aralarındaki bağlantılar daha da karmaşık hale gelmektedir.

## 4.1 Cümlelerin Bağlılık Analizi

Cümlede bağlılık analizi, cümlede yüklemden başlamak üzere, önce yüklem haricinde cümlenin ana öğelerinin yükleme bağlanması, ve öğelerin içerisindeki kelimelerin ise öğe içerisinde hiyeraşik olarak bağlantılarla birbirlerine bağlanmasıyla ortaya çıkan bir analizdir.

Bir örnek cümlenin bağlılık analizi Şekil 4.1’deki gibidir:




**Şekil 4.1 :** Bir cümlenin bağlılık analizi

Şekil 4.1'de de görüldüğü gibi en üst hiyerarşi düzeninde her bir cümle öğesi yükleme bağlanıyor. Bir alt seviyede öğeler kendi içerisinde hiyerarşik olarak birbirlerine bağlıdır. Bu bağlantıların tipi ve hangi kelimeden hangi kelimeye olduğu bilgisi cümlelerin sistematik bir şekilde incelenmesine imkan verir.

# 5. SİSTEM YAPISI

Sistemimizde kullanılan yapılar hakkında yapılan bilgilendirmeden sonra bu bölümde tez amacını gerçekleyen sistemin yapısını, bileşenlerini ve bileşenlerin işleyiş sırası hakkında detaylı bilgi bu bölümde verilecektir.

Sistem için iki ayrı yöntem kullanılmış ve bu iki yöntem için system yapısı ayrı ayrı incelenecektir.

- Kural Tabanlı Yöntem
- KRA Tabanlı Yöntem

Her iki yöntemde de metnin sistemin ana mekanizması tarafından işleme alınmadan önce gördüğü ön işlemler ve analiz işlemleri birebir aynıdır. Bu nedenle metni hazırlama modülü iki ayrı sistemde ayrı ayrı incelemek yerine bir başlık altında incelenmiştir. İlgili yöntem başlıkları altında metnin hazırlık işlemlerinin yeri açıklanmıştır. Metnin hazırlık işlemleri tamamlandıktan sonra işlenmiş metne ilgili yöntem uygulanmıştır.

## 5.1 Metnin Hazırlanması

Bu bölümde ham metin olarak sisteme giren veri işlenerek her iki yöntem için gerekli olan formata dönüştürülür ve metinle alakalı gereken bilgiler elde edilip her iki mekanizma için ilgili veri formatında ilgili sisteme gönderilir. Beş ana kısımdan meydana gelir:

- Metnin Normalizasyon Hatalarının Düzeltilmesi
- Metnin Cümlelere Ayrılması
- Metnin Kelimelere Ayrılması
- Metnin Biçimbilimsel Analizinin Yapılması
- Metnin Biçimbirimsel Belirsizliğinin Giderilmesi

### 5.1.1 Metnin normalizasyonu

Üzerinde çalıştığımız verilerin büyük çoğunluğunun web verisi olmasından dolayı metin normalizasyonuna ihtiyaç duyulmuştur. Bunun için (Eryiğit,2013) İ.T.Ü. doğal dil işleme grubuna ait normalizasyon aracı kullanılmıştır. Normalizasyon işlemi, sesli harf düzeltme, sesli harf tamamlama, hece kontrolü, Türkçe karakterleri düzeltme vb. işlemleri kapsamaktadır.

### 5.1.2 Metnin cümlelere ayrılması

Normalize edilmiş metnin cümlelere ayrılma işlemi Türkçe'de cümleleri ayıran noktalama işaretlerinin herhangi biri görüldüğünde metni o noktadan itibaren bölme koşulunu çalıştırarak bölen bir mekanizma ile yapılmıştır. Dışarıdan referans herhangi bir araç kullanılmamıştır.

### 5.1.3 Metnin kelimelere ayrılması

Cümlelere ayrılmış metnin her bir cümlenin içerisindeki kelimelerin arasında bir veya birden fazla boşluk olduğunu kabul ederek sistemin kendi içerisinde verilen cümleleri kelimelere ayırabilecek geliştirme yapılmıştır. Bu yüzden bu işlem için de hariçten bir araç kullanılmamıştır.

### 5.1.4 Metnin biçimbilimsel analizinin yapılması

Biçimbilimsel çözümlemeye, birinci yöntem için cümle bağlılık analizi için gerekli olan bilgileri vermek ve ikinci yöntem için ise sınıflandırıcı için gerekli olan bilgileri vermek için gerek duyulmuştur. Bunun için Oflazer'in (Oflazer, 2013), (Beesly, 2013) Xerox sonlu durum makineleri üzerinde derlediği iki seviyeli çözümleyici kullanılmıştır.

Analiz sonucu çıkan veriye örnek verecek olursak;

***"Kesinlikle Azkaban'a gönderilmek gibi bir niyetim yok."***

cümlesinin analiz sonucu Şekil 5.1'deki gibidir:

Şekil 5.1'de de görüldüğü gibi bazı kelimelerin biçimbirimsel analizi birden fazla bulunmaktadır. Örnek cümlede ki "kesinlikle" kelimesinin birden fazla analiz sonucu ürettiği şekilde görüldüğü gibidir.

```
Kesinlikle:
kesinlikle+Adverb
kesinlik+Noun+A3sg+Pnon+Ins

Azkaban'a:
Azkaban'a+?

gönderilmek:
gönderil+Verb+Pos^DB+Noun+Inf1+A3sg+Pnon+Nom

gibi:
gibi+Adverb
gibi+Postp+PCGen
gibi+Postp+PCNom

bir:
bir+Adverb
bir+Adj
bir+Num+Card

niyetim:
niyet+Noun+A3sg+Pnon+Nom^DB+Verb+Zero+Pres+A1sg
niyet+Noun+A3sg+P1sg+Nom

yok:
yok+Noun+A3sg+Pnon+Nom
yok+Adj
yok+Conj
yok+Adverb
yok+Interj

.:
.+Punc
```

**Şekil 5.1 :** Örnek cümlenin morfolojik analiz sonucu

Şekil 5.2'dekş gibi bir başka örnek daha verecek olursak;

```
Burada:
burada+Adverb

ne:
ne+Adj
ne+Adverb
ne+Conj
ne+Pron+Ques+A3sg+Pnon+Nom

kadar:
kadar+Postp+PCGen
kadar+Postp+PCNom
kadar+Postp+PCDat

süre:
sür+Verb+Pos+Opt+A3sg
süre+Noun+A3sg+Pnon+Nom

kalacaksınız:
kal+Verb+Pos+Fut+A2pl

?:
?+Punc
```

**Şekil 5.2 :** Diğer örnek cümlenin morfolojik analiz sonucu

"Burada ne kadar süre kalacaksınız?" cümlesinin analizinden ne kadar çok ihtimal çıktığı Şekil 5.2'de görülmektedir. Bu nedenle analizini elde ettiğimiz verinin analiz verisini verimli ve etkin bir şekilde kullanmak için bu belirsizliklerin giderilmesi ve her bir kelimeye ait analiz sayısının bir olması gerekmektedir.

### 5.1.5 Metnin biçimbilimsel belirsizliğinin giderilmesi

Biçimbirimsel belirsizliğin giderilmesi için Sak 'ın uygulaması (Sak,2007) kullanılmıştır.

Yukarıdaki örneklerin biçimbilimsel belirsizliğinin giderilmiş hali Şekil 5.3 ve Şekil 5.4'deki gibidir:

```
<S> <S>+BSTag
Kesinlikle kesinlik+Noun+A3sg+Pnon+Ins
Azkaban'a Azkaban'a+?
gönderilmek gönderil+Verb+Pos^DB+Noun+Inf1+A3sg+Pnon+Nom
gibi gibi+Postp+PCGen
bir bir+Adj
niyetim niyet+Noun+A3sg+P1sg+Nom
yok yok+Adverb
. .+Punc
<S> <S>+BSTag
```

**Şekil 5.3 :** Örnek cümlenin morfolojik belirsizliğinin giderilmiş enson hali.

```
<S> <S>+BSTag
Burada burada+Adverb
ne ne+Adverb
kadar kadar+Postp+PCNom
süre süre+Noun+A3sg+Pnon+Nom
kalacaksınız kal+Verb+Pos+Fut+A2pl
? ?+Punc
<S> <S>+BSTag
```

**Şekil 5.4 :** Diğer cümlenin morfolojik belirsizliğinin giderilmiş enson hali.

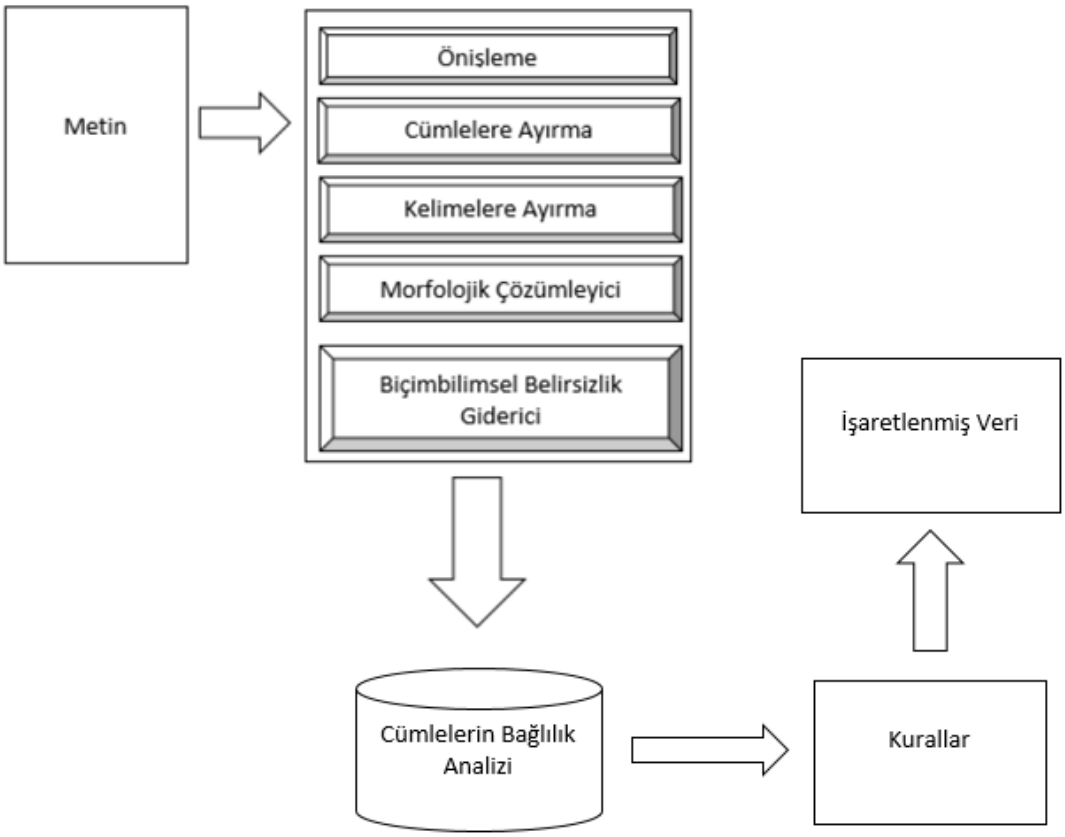
Elde edilen metin enson biçimbirimsel belirsizlik gidericiden de geçtikten sonra daha önceden bahsettiğimiz sınama verisi formatına sokulur. Ve o şekilde sınıflandırıcıya gönderilir.

## 5.2 Kural Tabanlı Yöntem

Bu yöntem ilk olarak kullanmış olduğumuz yöntemdir. Üç ana kısımdan oluşur:

- Sistem için metnin önişlemeden geçirilerek uygun formata dönüştürülmesi ve biçimsel çözümlemesinin yapılması.
- Biçimbilimsel çözümlemesinin yapıldığı verinin bağlılık analizinin çıkarılması.
- Çıkarılan bağlılık analiz bilgisini kullanarak daha önceden oluşturulmuş kuralların uygulanarak metnin üzerine tamlama bilgisinin atanması.

Sistemin yapısı Şekil 5.5 ‘te gösterilmiştir:




**Şekil 5.5 :** Sistemin Genel İşleyiş

**“**Dış ticaret açığı bir önceki aya göreyse yüzde 20, 4 arttı . “ cümlesinin birinci modülden geçirilmiş hali Şekil 5.6’daki gibidir:

```
<S> <S>+BSTag
Dış dış+Noun+A3sg+Pnon+Nom
ticaret ticaret+Noun+A3sg+Pnon+Nom
açığı açık+Noun+A3sg+P3sg+Nom
bir bir+Num+Card
önceki önce+Noun+A3sg+Pnon+Nom^DB+Adj+Rel
aya ay+Noun+A3sg+Pnon+Dat
göreyse gör+Verb+Pos+Opt+Cond+A3sg
yüzde yüz+Num+Card^DB+Noun+Zero+A3sg+Pnon+Loc
20, 20,+?
4 4+Num+Card
arttı art+Verb+Pos+Past+A3sg
. .+Punc
```

**Şekil 5.6 :** Örnek cümlenin morfolojik belirsizliğinin giderilmiş en son hali.

Birinci modülden morfolojik olarak analiz edilmiş cümlenin cümle bağlılık analizi Şekil 5.7'deki gibidir:

```
1   Dış dış Noun    Noun    A3sg|Pnon|Nom   2   MWE
2   ticaret ticaret Noun    Noun    A3sg|Pnon|Nom   3   MWE
3   açığı   açık    Noun    Noun    A3sg|P3sg|Nom   13  SUBJECT
4   bir bir Num Card    _   6   MODIFIER
5   _   önce    Noun    Noun    A3sg|Pnon|Nom   6   DERIV
6   önceki  _   Adj Rel _   7   MODIFIER
7   aya ay  Noun    Noun    A3sg|Pnon|Dat   8   DATIVE.ADJUNCT
8   göreyse gör Verb    Verb    Pos|Opt|Cond|A3sg   13  MODIFIER
9   _   yüz Num Card    _   10  DERIV
10  yüzde   _   Noun    Zero    A3sg|Pnon|Loc   13  MODIFIER
11  20, 20, ?   ?   _   13  MODIFIER
12  4   4   Num Card    _   13  MODIFIER
13  arttı   art Verb    Verb    Pos|Past|A3sg   14  SENTENCE
14  .   .   Punc    Punc    _   0   ROOT
```

**Şekil 5.7 :** Örnek cümlenin bağlılık analizi

Bu safhadan sonra oluşturduğumuz kurallar, bağlılık analiz bilgileri kullanılarak uygulanır.

### 5.2.1 Kural yapıları

Uygulanacak kuralları çıkarırken 250 cümleden oluşan bir geliştirme seti kullanılmıştır. İlk önce genel durumu görmek amacıyla geliştirme setindeki cümlelerin bağlılık analizinden çıkan bağıntıların isim tamlaması durumunu belirten etiketlere göre dağılımları çıkartıldı. Dağılımların detayı Çizelge 5.1'deki gibidir:

**Çizelge 5.1 :** Nitelik Komb. Göre Sınama Grubu Sonuçları

| BAĞLILIK TİPİ | BB | BI | BH | BO | IB | II | IH | IO | HB | HI | HH | HO | OB | OI | OH | OO | DİĞER | TOTAL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ABLATIVE.ADJUNC. | 0 | 0 | 2 | 5 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 17 | 0 | 0 | 1 | 6 | 0 | 35 |
| APPOSITION | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CLASSIFIER | 0 | 17 | 13 | 0 | 0 | 26 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 84 |
| COLLOCATION | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| COORDINATION | 0 | 0 | 0 | 0 | 0 | 25 | 11 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 10 | 14 | 0 | 65 |
| DATIVE.ADJUNCT | 0 | 0 | 1 | 8 | 0 | 8 | 0 | 8 | 1 | 1 | 3 | 40 | 0 | 0 | 1 | 21 | 0 | 92 |
| DETERMINER | 0 | 10 | 17 | 0 | 0 | 4 | 5 | 0 | 3 | 1 | 1 | 1 | 1 | 0 | 0 | 8 | 0 | 51 |
| EQU.ADJUNCT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ETOL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FOCUS.PARTICLE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| INSTRUMENTAL.ADJUNCT | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 1 | 0 | 6 |
| INTENSIFIER | 0 | 0 | 0 | 0 | 0 | 9 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 10 | 10 | 0 | 32 |
| LOC.ADJUNCT | 0 | 0 | 0 | 7 | 0 | 4 | 1 | 6 | 0 | 2 | 0 | 29 | 0 | 0 | 0 | 6 | 0 | 55 |
| MODIFIER | 0 | 108 | 109 | 24 | 2 | 168 | 116 | 70 | 3 | 0 | 2 | 59 | 2 | 7 | 13 | 135 | 0 | 818 |
| MWE | 0 | 70 | 28 | 0 | 1 | 79 | 67 | 0 | 14 | 0 | 1 | 20 | 1 | 0 | 1 | 540 | 0 | 322 |
| NEG.PARTICLE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 3 |
| NOTCONNECTED | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 1 | 101 | 5 | 37 | 7 | 0 | 169 |
| OBJECT | 0 | 15 | 5 | 3 | 0 | 31 | 27 | 7 | 0 | 1 | 2 | 142 | 0 | 0 | 2 | 64 | 0 | 299 |
| POSSESSOR | 0 | 9 | 17 | 1 | 0 | 10 | 42 | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 83 |
| QUES.PARTICLE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| RELATIVIZER | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ROOT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 742 | 742 |
| S.MODIFIER | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 6 |
| SENTENCE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 223 | 0 | 242 |
| SUBJECT | 0 | 10 | 3 | 38 | 0 | 10 | 5 | 118 | 0 | 2 | 6 | 183 | 0 | 0 | 1 | 230 | 0 | 396 |
| VOCATIVE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 3 |

Her bir kelimenin tamlama konumu ile ilgili olarak kural tabanlı sistem için 4 ayrı etiket kullanıldı (Ramshaw ve Marcus, 1995):

- B: Tamlamanın ilk kelimesi
- I: Tamlamanın ara kelimesi
- H : Tamlamanın ana kelimesi
- O: Tamlama dışı kelime

Bu tablodan çıkarılan ilk sonuç, B'ye giden bağlantı tipleri (BB, IB, OB, HB ) yok

denecek kadar az olması bir cümlede kendisine hiç bir bağlantı olmayan bir kelimenin potansiyel bir “B” olması anlamının çıkartılmasını sağlamıştır. Kural tabanlı sistem, bu sonuçtan başlayarak her bir cümlede her bir kendisine bağlantı olmayan kelime olan potansiyel ”B” olan kelimeden başlayarak bağlılıklar boyunca eklenerek devam eden aday tamlamalar aşağıda belirtilen üç koşuldan biri halinde sonlandırılması esasına dayandırılan bir sistem olarak tasarlandı. Tamlamaların sonlandırma koşulları aşağıdaki gibidir:

- Yükleme bağlanıyorsa
- Object veya subject olarak herhangi bir isim – fiil ,bağ-fiile, edat, zarf veya fiile bağlanıyorsa,
- İsim Cümlelerinde yükleme bağlanıyorsa.

Ayrıca aday tamlamalarının her birinin ana kelimesinin isim veya isim soylu sözcük olması sağlanmıştır.

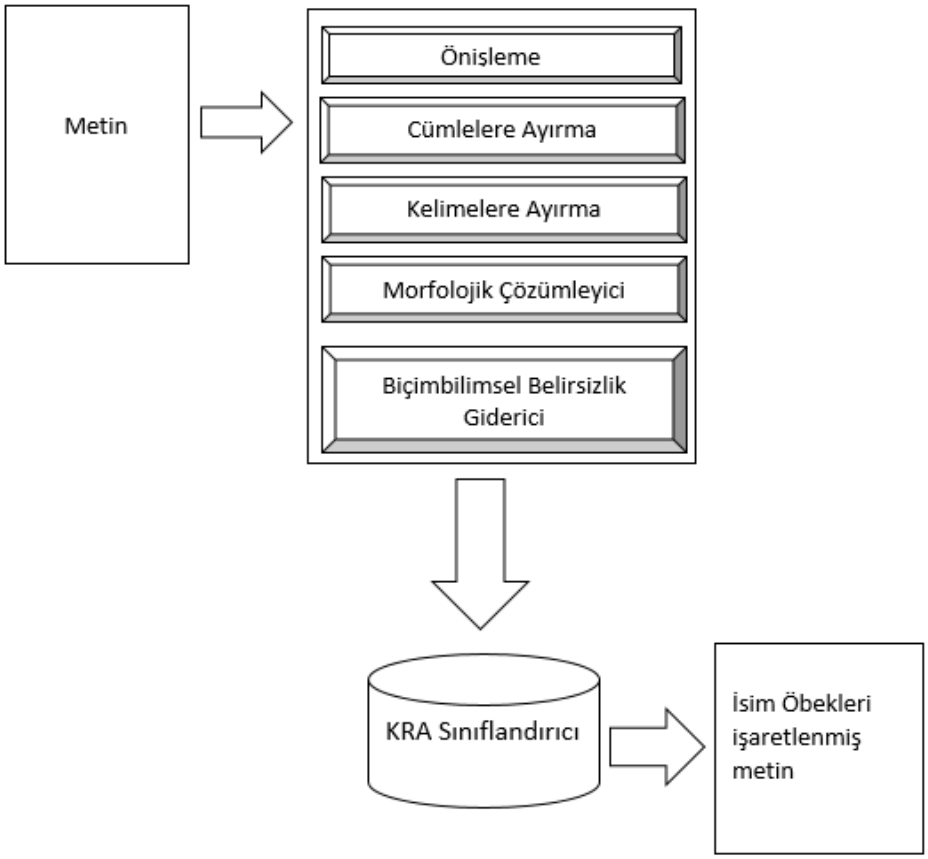
### 5.3 KRA Tabanlı Yöntem

İkinci ve ana yöntemimizdir. Sistem temel olarak dört ana bölümden oluşmaktadır:

- Sistem için metnin önişlemeden geçirilerek uygun formata dönüştürülmesi ve biçimsel çözümlemesinin yapılması.
- Daha önceden parametreleri optimize edilmiş model kullanılarak CRF++ aracı ile sınama işlemi yapılması
- Elde edilen etiketli verinin tekrardan işaretli veriye dönüştürülmesi

Daha önceden eğitilmiş ve optimize edilmiş model ile birlikte formatı hazırlanmış metni KRA sınıflandırıcıya verdiğimizde çıktı olarak isim tamlamalarının sınırları belli olan veriyi alabiliriz. Sadece yapmamız gereken ek işlem sınıflandırıcıdan gelen sınır etiketlerine göre cümlelerdeki ilgili yerlerdeki isim tamlamalarını işaretlemektir

Sistemin genel yapısının özeti Şekil 5.8’ de gösterilmiştir.




**Şekil 5.8 :** Sistemin Genel İşleyişi

# 6. SINAMA

Bu bölümde ana yapısının bir önceki bölüm.de verilen sistemin optimizasyonu ve en iyi sonuca ulaşması için gerçekleştirilen testler ve sonuçları verilecektir. Bunun için ilk önce eğitim ve sınama veri setleri, daha sonra grup testlerinde kullanılan metriklerden bahsedilecektir. Daha sonra sınama grupları ve elde edilen test sonuçları en basitinden en karmaşığına doğru ilerleyen bir şekilde verilecektir.

## 6.1 Kullanılan Veri Kümeleri

Kural tabanlı sistemimiz için 500 cümlelik işaretlenmiş test verisini 250 cümlesi geliştirme kümesi 250 cümlelik kısmı ise sınama kümesi olmak üzere iki eşit parça halinde kullanılmıştır. Kuralları 250 cümlelik geliştirme kümesi üzerinde geliştirmiş, sınamalarını ise diğer 250 cümlelik sınama veri seti kullanılarak alınmıştır. Bu 500 cümlenin içerisinde el ile işaretlenmiş 805 tane isim tamlaması bulunmaktadır.

Tezin amacını gerçekleştirmek için tasarladığımız ikinci sistem bir makine öğrenmesi tekniği kullanması sebebiyle eğitim ve sınama veri kümeleri olmak üzere iki çeşit veri kümesine ihtiyaç duyulmuştur.

Birinci sınama grubunda ise 600 K lık bir web derleminden seçilmiş ve (yukarıda kural tabanlı sistemde de kullanılan) 500 cümlelik bir test verisi işaretlenmiş, ve geri kalan derlemden ise bazı parametreler kullanılarak 221534 cümlelik bir eğitim kümesi elde edilmiştir. Bu eğitim kümesinin üzerindeki çalışmalar ise yine eğitim kümesine seçilen cümlelerle alakalı olduğu için eğitim kümesinin verileri sonuçlarla beraber verilecektir.

İkinci sınama grubunda ise yine 500 cümlelik aynı test seti kullanılmıştır. Yine 221534 cümlelik veri seti üzerinde seçme işlemleri yapılmıştır.

Üçüncü sınama grubunda, bu kez 1M sayıda alınan bir derlemden 100K ‘lık ve 200K’lık cümle (Yıldız ve Tantuğ, 2012) seçilmiş ve bu cümle sayıları eğitim verisi olarak kullanılmıştır. Bu derlemlerin testinde yine 500 cümlelik test kümesi kullanılmıştır.

## 6.2 Eğitim Verisinin Otomatik Olarak Elde Edilmesi

Tezimizin amacı olan isim öbeklerinin bulunması için kullanılan yöntem bir makine öğrenmesi tekniğidir. Bunun için hazır olarak işaretlenmiş veri kümesi gerekmektedir. Bu gereksinimi karşılamak için el ile işaretleme yapmak yerine aşağıdaki adımların izlenmesiyle eğitim verisinin otomatik olarak elde edilmesi sağlanmıştır. Bu yöntemin uygulanabilirliği için parallel derleme ihtiyaç duyulmaktadır.

- Türkçe cümlelerin karşılığı olan ingilizce cümleler Stanford Parser (Socher, 2013) aracı ile ingilizce cümlelerdeki birinci dereceden isim tamlamalarının işaretlenmesi
- GIZA++ (GIZA++,2013) ile Türkçe cümlelerdeki kelimelerin İngilizce parallel cümlelerdeki kelimelere eşleştirilmesi. Bunun için Türkçe sondan eklemeli bir dil olması sebebiyle önce Türkçe cümlelerdeki kelimelerin gövdeleri alınmıştır böylece daha tutarlı eşleşmeler sağlanmıştır.
- Enson olarak İngilizce cümlelerdeki tamlamaları Türkçe' deki eşleşen kelimelere yönelterek Türkçe cümlelerdeki tamlamaların işaretlenmesi sağlanmıştır.

Bu durumda eğitim verisi için en iyi işaretlenmiş cümleleri seçmek gerekmektedir.

## 6.3 Kullanılan Değerlendirme Ölçütleri

Kullanılan değerlendirme ölçütleri 3 çeşittir:

- Kesinlik (P): Olması gereken öbeklerin yüzde kaçının bulunduğunu gösterir.
- Gerigetirim (R): Bulunan öbeklerin yüzde kaçının doğru olduğunu gösterir.
- F Skoru: P ve R nin harmonik ortalamasıdır.

Bu üç çeşit skorun bir de ikiye dallanmış halleri bulunmaktadır:

- Tam öbek eşleşmesi
- Kısmi öbek eşleşmesi : Her bir öbek tam eşleşmediği zaman eşleştiği oranda puan alır, cümlenin puanı toplam öbeklerden aldığı puanın o cümledeki toplam öbek sayısına bölümüdür.

Bu iki çeşidi bir örnek ile açıklayacak olursak;

"Ali'nin kırmızı arabası" : bulunması gereken tamlama

"kırmızı arabası" : bulunan tamlama,

Böyle bir sonuca tam öbek eşleşmesi 0 puan verirken kısmi öbek eşleşmesi ise 2/3=%66,7 puan verir.

Bu iki faktörü de göz önünde bulundurduğumuzda, toplamda 6 tane ölçüt değerimiz bulunmaktadır.

### 6.4 Sınama Grupları ve Sonuçlar

Kural tabanlı yöntem için elde ettiğimiz optimum sonuç, Çizelge 6.1'deki gibidir:

**Çizelge 6.1 :** Kural tabanlı sistem sonuçları

| Sistem | Tam Eşleşme(F1) | Yarı Eşleşme(F1) |
|---|---|---|
| Kural Tabanlı Sistem | 40.32 | 57.75 |

Çizelge 6.1'de de görüldüğü gibi kural tabanlı sistemin başarımı tam eşleşme de %40.32, kısmi eşleşmede %57.75 ' dir.

KRA tabanlı sistem için modelimizi optimize etmek amacıyla üç farklı grup sınama işlemi gerçekleştirilmiştir. İlk grupta, 500 cümlelik test verisi üzerinde, 600 K lık veriden 35 kelimelik cümle boyutu kıstıyla ve Türkçe – İngilizce eşleşmelerinde bir Türkçe kelimeye 4 ten fazla İngilizce kelime gelmeme kısıtının sonucunda elimizde 221534 cümlelik eğitim verisi kalmıştır. Bu eğitim verisinden eşleşme skoruna göre sıralanmış ve ilk 200K cümle dört eşit parçaya bölünüp eğitim ve sınama safhalarından geçirilmiştir. Bu sınama grubu eşleşme skorunun KRA modelinin başarımında nasıl etkili olduğunu görmek amacıyla yapılmıştır.

**Çizelge 6.2 :** Sadece eşleşme skoruna göre ve verinin sabit tutulduğu sonuçlar

| *Parça No* | *1* | *2* | *3* | *4* |
|---|---|---|---|---|
| **Sınama Verisi** | 500 | 500 | 500 | 500 |
| **Ort. Eşleşme Skoru** | 0,20425 | 0,117437 | 0,083674 | 0,057742 |
| **Tam Eşleşme ( R )** | 44.13 | 43.85 | 43.06 | 41.27 |
| **Kısmi Eşleşme(R)** | 68.69 | 69.44 | 70.04 | 70.1 |
| **Tam Eşleşme ( P )** | 48.89 | 47.08 | 46.02 | 43.22 |
| **Kısmi Eşleşme(P)** | 76.09 | 74.56 | 74.86 | 73.41 |
| **Tam Eşleşme ( F )** | 46.39 | 45.41 | 44.49 | 42.23 |
| **Kısmi Eşleşme(F)** | 72.20 | 71.9 | 72.37 | 71.71 |
| **Eğitim Verisi** | 50000 | 50000 | 50000 | 50000 |

Çizelge 6.2'deki sonuçlardan da anlaşılacağı gibi eşleşme skorunun başarıya çoğu anlamda etki ettiği fakat tek başına yeterli olmadığı görülebilmektedir.

Bu yüzden cümlelerin işaretlenme kalitesini tam olarak ölçebilmek için içerisinde eşleşme skorunun ve öbekleme skorunun bir arada olduğu yeni bir total skora göre söz konusu testler bir daha yapıldı.

$$\boldsymbol{Eş.Skoru = LOG(eşleşme\ skoru) \div \left(\frac{Türkçe\ cümle\ boyutu + ingilice\ cümle\ boyutu}{2}\right)} \quad (2)$$

$$\boldsymbol{Öbekleme\ Skoru = \frac{ÖBEKLEME\ SKORU}{ingilizce\ cümle\ boyutu}} \quad (3)$$

$$\mathbf{Total\ Skor = Normalizasyon(Eş.\ Skoru) + Norm.(Öbekleme\ Skoru)} \quad (4)$$

Yukarıdaki verilen üç ayrı eşitlik total skorun hesaplanmasını anlatmaktadır. Total skora göre sonuçlar Çizelge 6.3'teki gibidir.

Tablodan da anlaşılacağı gibi tüm ölçütlere ait değerler birinci parçaya yani total skoru en yüksek olan parçaya aittir. Bu da total skorun tek başına otomatik işaretleme kalitesini yansıtabildiğini gösteriyor.

**Çizelge 6.3 :** Total skora göre ve veri boyutunun sabit tutulduğu sonuçlar

| *Parça No* | *1* | *2* | *3* | *4* |
|---|---|---|---|---|
| **Sınama Verisi** | **500** | 500 | 500 | 500 |
| **Ort. Eşleşme Skoru** | **0,20425** | 0,117437 | 0,083674 | 0,057742 |
| **Tam Eşleşme ( R )** | **47.28** | 44.06 | 41.92 | 40.70 |
| **Kısmi Eşleşme(R)** | **71.91** | 70.31 | 68.73 | 68.56 |
| **Tam Eşleşme ( P )** | **49.77** | 47.72 | 45.64 | 43.80 |
| **Kısmi Eşleşme(P)** | **75.70** | 76.14 | 74.84 | 73.79 |
| **Tam Eşleşme ( F )** | **48.50** | 45.82 | 43.70 | 42.2 |
| **Kısmi Eşleşme(F)** | **73.75** | 73.11 | 71.65 | 71.08 |
| **Eğitim Verisi** | **50000** | 50000 | 50000 | 50000 |

Yaptığımız deneyler KRA aracının en fazla 200K cümle ile bir model eğitebildiğini söylüyor. Bu sonuçtan hareketle KRA modelimizi bir miktar daha optimize edebilmek için 1000M lik daha büyük bir derlemden total skora göre 200Klık bir veri kümesi seçip kullanabileceğimiz maximum veri ile model eğitme işlemi yapıldı. İlk önce 1000M lik veri total skora göre sıralanıp daha sonra 100Klık 10 eşit parçaya bölündü. Ve model, bu 10 eşit parça ile teker teker eğitilip test edildi. Sonuçlar Çizelge 6.4'teki gibidir:

**Çizelge 6.4 :** Total skora göre ve verinin sabit tutulduğu ve 100 Klık Sonuçlar

| Veri Grubu (100K) | Cümle Sayısı | Tam Eşleşme(F1) | Kısmi Eşleşme(F1) |
|---|---|---|---|
| 1 | 100K | 42.66 | 67.69 |
| 2 | 100K | 46.18 | 70.23 |
| 3 | 100K | 46.03 | 71.58 |
| 4 | 100K | 44.45 | 71.28 |
| 5 | 100K | 43.22 | 70.76 |
| 6 | 100K | 40.58 | 69.41 |
| 7 | 100K | 40.48 | 69.62 |
| 8 | 100K | 38.68 | 68.39 |
| 9 | 100K | 40.58 | 69.71 |
| 10 | 100K | 37.78 | 60.01 |

Çizelge 6.4'teki tabloya göre optimum modelimizi eğitmek için kullanacağımız 200K lık veri, 2. ve 3. 100 K lık parçaların birleşimidir. Elimizde bulundurduğumuz 1000 Klık veriden maximum verim alacağımız 200 Klık veriye karar verdikten sonra nitelik seçimi ile ilgili daha detaylı bir çalışma yapıldı. Üzerinde çalışma yapılan toplam 14 tane nitelik aşağıdaki gibidir:

- Kelime gövdesi
- Kök
- Tip Etiketi
- İyelik Eki
- Tamlama Eki
- Özel İsim Olup Olmadığı
- İsmin Hal Ekleri ( Her biri bir nitelik olarak kullanılmak üzere )
- Çoğul Eki
- Son 4 karakter
- Sıfat – fiil olup olmama özelliği (gel- den gelen vb.)

Bu nitelikler sadece gövde özelliği taban değeri olarak alınmış, diğer nitelikler en yüksek sonuç verme sırasına göre teker teker eklenerek başarımları ölçülmüş, en yüksek başarımlı nitelik grubu elde edilmiştir. Nitelik test sonuçları Çizelge 6.5'teki gibidir:

Tabloya baktığımızda en iyi kombinasyonun ilk onbir niteliğin bulunduğu koyu renk ile belirlenmiş kombinasyondur. Bu sonuçlar, yapılması kolay ve çabuk sürmesi açısından 100 Klık bir veri ile yapılmıştır. Bulmuş olduğumuz optimum nitelik kombinasyonunu, daha önceden bulunan optimum 200K lık veri ile eğitilerek optimum sonuç bulunmuştur. Sonuçlar Çizelge 6.6'daki gibidir:

**Çizelge 6.5 :** Optimum nitelik kombinasyonunu için yapılmış test sonuçları

| Sıra No | Eklenen Nitelik | Tam Eşleşme(F1) | Yarı Eşleşme(F1) |
|---|---|---|---|
| 0 | Gövde | 41.71 | 68.48 |
| 1 | Kök | 43.09 | 69.52 |
| 2 | Tip Etiketi | 43.44 | 69.49 |
| 3 | Son 4 Kar. | 44.21 | 69.61 |
| 4 | İyelik Eki | 44.35 | 70.05 |
| 5 | Tamlama Eki | 44.81 | 69.91 |
| 6 | Yalın Hali | 46.48 | 70.91 |
| 7 | De Hali | 46.86 | 71.54 |
| 8 | i Hali | 46.77 | 71.27 |
| 9 | Çoğul olma | 46.81 | 71.19 |
| 10 | e Hali | 47.05 | 71.00 |
| **11** | **den Hali** | **47.08** | **70.72** |
| 12 | Sıfat Fiil Eki | 46.77 | 70.35 |
| 13 | Kelime Pozisyonu | 44.85 | 69.33 |
| 14 | Özel İsim Olma | 44.65 | 69.3 |

**Çizelge 6.6 :** Optimum Sonuçlar

| Veri Sayısı | Nitelik Kombinasyonu | Tam Eşleşme (F1) | Yarı Eşleşme (F1) |
|---|---|---|---|
| 100K | Optimum Komb. | 47.08 | 70.72 |
| 200K | Optimum Komb. | 52.28 | 76.79 |

KRA sisteminden elde edilen optimum sonuç birebir eşleşmede %52,28, ve kısmi eşleşmede % 76,79'dur.

Her iki sistemin karşılaştırılmış sonuçları ise Çizelge 6.7'de verilmiştir:

**Çizelge 6.7 :** İki Sistemin Karşılaştırması

| Eğitim Veri Sayısı | Nitelik Kombinasyonu | Tam Eşleşme(F1) | Yarı Eşleşme(F1) |
|---|---|---|---|
| 200K | KRA Tabanlı Sistem | 52.28 | 76.79 |
| - | Kural Tabanlı Sistem | 40.32 | 57.75 |

Tablodan da anlaşıldığı gibi KRA tabanlı sistem, kural tabanlı sistemden daha yüksek başarımlı bir sistemdir. Başarıma ek olarak uygulanabilirlik ve dilden bağımsızlık açısından da KRA tabanlı sistem çok daha avantajlı bir duruma sahiptir.

# 7. SONUÇ VE ÖNERİLER

Yapay zekanın bir alt birimi olan doğal dil işleme, insanoğlunun kendisinde bulunan en güçlü iletişim aracını yani dili makineler alemine tanıtmayı, anlatmayı ve de dilin yapabildiklerini makine dünyasına taşımayı amaçlar. Dile en küçük ifade birimi olan cümleyi eğer dilin makineler alemine taşınmasını istiyorsak çözümleme zorunluluğu kaçınılmaz bir gerçektir. Cümle, bilgisayar ortamnda hem yapısal olarak hem de anlamsal olarak çözümlendiği zaman dil makineler dünyasına aktarılabildiği söylenebilir.

Cümlenin yapısal kısmı bir kenara bırakıldığında anlamsal kısmının anlaşılabilmesi için içerisindeki öbeklerin bir başka deyişle parçacıkların ilk önce tesbit edilmesi gerekmektedir. Bu tesbit işlemi cümlenin dilbilgisi yapısının derinlemesine yapmadan da yüzeysel öbekleme sayesinde cümlenin içerisinde bulunan ana bölümlerin belirlenip anlamlarının çıkarılmasıyla gerçekleşebilir.

Bu amaç doğrultusunda bu çalışma kapsamında gerçekleştirilen çalışma, bir Türkçe cümlenin içerisindeki isim öbeklerini bir başka deyişle isim ve sıfat tamlamalarını bulmaktadır. Yöntem olarak iki ayrı yöntem uygulanmıştır. İlk yöntem olarak cümlelerin bağlılık analizi sonuçlarını basit kurallara tabi tutmaya dayanan kural tabanlı bir yöntem uygulanmıştır. Yöntem, beş ayrı safhadan oluşan önişleme kısmı, bağlılık analizlerinin çıkarılması, ve kuralların uygulandığı kısım olmak üzere üç ayrı bölümden oluşmaktadır. İkinci yöntem olarak, KRA tabanlı yöntem, beş ayrı safhadan oluşan önişleme kısmı ve optimize edilen bir makine öğrenmesi modeli sayesinde cümledeki tamlamaların sınırlarını çizmektedır.

Kural tabanlı sistem için geliştirme kümesi üzerinden çıkarılan bir takım istatistiksel verilerden yola çıkarak kurallar geliştirilmiştir. Bu kuralların kullandığı parametreler kelimelerin morfolojik özelliklerinden ve bağlılık analizinin vermiş olduğu bilgiden oluşmaktadır. Yöntemi geliştirmek diğer yöntemden daha kolay olmasına rağmen hem dilin dilbilgisi kurallarına bağımlı olması açısından hem de çıkarılan kuralların zamanla değişme ihtimali olacağından kullanışlı bir yöntem değildir.

KRA tabanlı yöntem, bir makine öğrenmesi yöntemi olması sebebiyle elimizde hazır veri ile eğitilen ve en iyi sonucu almak için optimize edilen bir model olması gerekmektedir. Model ilk önce nitelikleri ve nitelik kombinasyonları açısından optimize edilmiş daha sonrada otomatik olarak üretilen hazır işaretlemmiş veri kalitesi açısından optimize edilmiştir. Veri kalitesini doğrudan belirleyen parametre olarak total skor adının verildiği değer olduğu görülmüştür. Ayrıca yöntem otomatik eğitim verisi üretmesi açısından son derece ayırt edici bir özelliğe sahiptir. İstendiği takdirde ve parallel derlem olduğu takdirde dilden bağımsız olarak da uygulanabilir.

Otomatik eğitim verisinin üretilmesi, karşı dildeki cümlenin öbekleme işleminin yapılması, karşı dildeki kelimelerin kaynak dilde cümledeki kelimelere eşleştirilmesi, ve en son olarak da karşı dildeki belirlenmiş öbeklerin kaynak dildeki eşleştirildiği kelimeler aracılığıyla kaynak dilde de yerlerinin tesbitinin sağlanması ile gerçekleşebilmektedir. Bu çalışma, otomatik eğitim verisi üreten ilk çalışma olma ünvanını taşımaktadır.

En son optimize edilmiş modelimizin başarımı birebir eşleşme için %52,28 (birebir eşleşme) ve %76,79 (kısmi eşleşme) dir. Manuel olarak hiçbir şey yapılmadan elde ettiğimiz bu sonuçlar umut vericidir. Fakat gelecekte yapmamız gereken çok şey mevcuttur. Bunlardan ilki daha büyük çapta parallel bir derlemden aynı kaliteye sahip daha çok sayıda cümle elde etmek ve bunlarla modeli tekrardan eğitmek suretiyle başarımı artırma şansı yakalamaktır.

Ayrıca işaretlenimş verinin kalitesini artırmak amaçlı ilk olarak geliştirdiğimiz kural tabanlı sistemin bir miktar daha geliştiriilip işaretlenmiş eğitim verisi üretmede kullanmak suretiyle yeni bir model eğitip başarımının ölçümü yapılabilir.

Bir diğeri ise nitelik sayısını ve dolayısıyla nitelik kombinasyon aralığını genişletmektir. Bu yolla da başarım artabilir. Üçüncüsü ise bu sistemin çıktısının üzerine dil modeli uygulamaktır. Yine bu yöntemde başarımı artırabilir çünkü dil modelleri makine öğrenmesi modellerinden çok daha fazla ve kapsamlı veri üzerinden çıkarılabilir.Son olarak optümum modelimizin diğer sistemler üzerinde etkisi ölçülebilir, çünkü zaten Türkçe cümlelerin isim tamlamalarının bulunması daha önce de belirtildiği gibi tek başına kullanılmasından ziyade metin madenciliği, cümlenin bağlılık analizinin çıkarılması, bilgi çıkarımı , varlık ismi tanıma vb. işlemlerde büyük ölçüde yapıcı ve yardımcı bir konumda bulunabilmektedir.

## KAYNAKLAR


[1] **Eryiğit, G., Adalı, E. ve Oflazer.K .** (2006)**.** Türkçe Cümlelerin Kural Tabanlı Bağlılık Analizi. In Proceedings of the 15th Turkish Symposium on Artificial Intelligence and Neural Networks, 17-24.

[2] **Eryiğit, G.,** (2010). Yapay Zeka ve Dil Teknolojileri, Bilişim Dergisi. Sf.82.

[3] **Church, K. W.** (1988)**.** A stochastic parts program and noun phrase parser for unrestricted. In Proceedings of the Second Conference on Applied Natural Language Processing. Austin, Texas

[4**] L.A. Ramshaw ve M.P. Marcus.** (1995). Text chunking using transformation based learning. In Proceedings of the Third Workshop on Veri Large Corpora. ACL.

[5] **Lafferty, J. D., McCallum, A. ve Pereira, F. C. N., (**2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML, Sf. 282–289.

[6] **Taku Kudo**, 2003. CRF++:Yet Another CRFToolkit http://chasen.org/~taku/software/CRF++/.

[7] **Cardie, C., ve Pierce, D.** (1998). Error-driven pruning of treebank grammars for base noun phrase identification. In Proceedings of COLING-ACL'98. Association for Computational Linguistics.

[8] **Voutilainen, A.** (1993). NPTool, a detector of English noun phrases. In Proceedings of the Workshop on Veri Large Corpora (pp. 48-57). Association for Computational Linguistics.

[9] **Kutlu, M.** (2010). Noun Phrase Chunker for Turkish Using Dependency Parser. Master Thesis, Bilken University, Ankara.

[10] **Coşkun, N.** (2013). Türkçe Tümecelerin Öğelerinin Bulunması. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, İstanbul.

[11] **Oflazer, K.** (1994). Two Level Descriptin of Turkish Morphology. Literary and Linguistic Computing, 9(2):137-148.

[12] **Gülşen Eryiğit**, 2013.ITU NLP Pipeline. http://tools.itu.edu.tr/.

[13] **Beesly, K. R., ve Karttunen, L.** (2003). Finite State Morphology. Version 1.5.2-incubating, CSLI Publications, Stanford, California, 2003 (CSLI Studies in Computational Linguistics, xviii+510 pp and CD-ROM, ISBN 1-57586-434-7) Center for the Study of Language and Information, Leland Stanford Junior University

[14**] Sak, Haşim, Güngör, Tunga ve Saraçlar, Murat.** (2007) . Morphological Disambiguation of of Turkish text with perceptron algorithm. ICICLing 2007, volume LNCS 4394, 107-118.

[15] **Richard Socher, John Bauer, Christopher D. Manning ve Andrew Y. Ng.** (2013). Parsing With Compositional Vector Grammars. Proceedings of ACL 2013

[16 ] **GIZA++ Toolkit** .2013. http://www.statmt.org/moses/giza/GIZA++.html

[17] **Kazkılınç, S. ve Adalı, E.**(2013) Koşullu Rastgele Alanlar ile Türkçe Haber Metinlerinin Etiketlenmesi

[18] **Hengirmen, M.**(2002) Türkçe Dilbilgisi, Sf : 118-142.

[19] **El-Kahlout, I.D., ve Akın A.A.**(2013) Turkish Constituent Chunking with Morphological and Contextual Features, 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part I

[20] **Dhanalakshmi V, Padmavathy P, ve Kumar M,**(2009) Chunker For Tamil. 2009 International Conference on Advances in Recent Technologies in Communication and Computing

[21] **Kermes, H. ve Evert, S.,**(2002) YAC – A Recursive Chunker for Unrestricted German Text. In Proceedings of the Third International Conference on Language Resources and Evaluation, Las

[22] **Vuckovic, V., Tadic, M. ve Dovedan, Z.,**(2008) Rule Based Chunker for Croatian, In Proceedings of International Conference on Language Resources and Evaluation.

[23] **Singh, A., Bendre, S. ve Rangal, R.**,(2002) HMM Based Chunker for Hindi In Proceedings of ACL 2002

[24] **Radziszewski, A. ve Piasecki, M.,** (2010) A Preliminary Noun Phrase Chunker for Polish. In Proceedings of Intelligent Information Systems. Sf:169-180

[25] **Atterer, M. ve Schlangen, D.,** (2009) RUBISC - a Robust Unification-Based Incremental Semantic Chunker. In Proceedings of ACL 2009

[26] **Asahara, M., Goh, C. L., Wang, X. ve Matsumoto, Y.,**(2003)Combining Segmenter and Chunker for Chinese Word Segmentation. In Proceedings of ACL 2003.

[27] **Chen, K., ve Chen, H.,** (1993) A Probabilistic Chunker. In: Proceedings of ROCLING VI

[28] **Torunoğlu, D., ve Eryiğit, G.,** (2014) A Cascaded Approach for Social Media Text Normalization of Turkish**.** In 5th Workshop on Language Analysis for Social Media (LASM) at EACL, Gothenburg, Sweden, April 2014

## ÖZGEÇMİŞ

**Ad Soyad: Kübra ADALI**

**Doğum Yeri ve Tarihi: ESKİŞEHİR / 26.01.1986**

**Adres:** **Merter/İSTANBUL**

**E-Posta:** **kubraadali@itu.edu.tr**

**Lisans:** **Gazi Üniversitesi / Bilgisayar Mühendisliği**

**Yüksek Lisans (Varsa):**

**Mesleki Deneyim ve Ödüller:**

**Yayın ve Patent Listesi:** Kübra Adalı and Gülşen Eryiğit. Vowel and Diacritic Restoration for Social Media Texts. accepted for publication In 5th Workshop on Language Analysis for Social Media (LASM) at EACL, Gothenburg, Sweden, April 2014.

## TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR