CUMHURİYET ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Ana Bilim Dalı

OTOMATİK DOKÜMAN ÖZETLEME YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Yüksek Lisans Tezi

Yunus Emre IŞIK

Sivas
Ocak 2018

CUMHURİYET ÜNİVERSİTESİ

Sosyal Bilimler Enstitüsü
Yönetim Bilişim Sistemleri Ana Bilim Dalı

OTOMATİK DOKÜMAN ÖZETLEME YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Yüksek Lisans Tezi

Yunus Emre IŞIK

Tez Danışmanı:

Doç. Dr. Oğuz KAYNAR

Sivas
Ocak 2018

KABUL VE ONAY

Üniversite:	**: Cumhuriyet Üniversitesi**
Enstitü	**: Sosyal Bilimler Enstitüsü**
Ana Bilim Dalı	: Yönetim Bilişim Sistemleri
Bilim Dalı	:
Tezin Başlığı	**: Otomatik Doküman Özetleme Yöntemlerinin Karşılaştırılması**
Savunma Tarihi	**: 03.01.2018**
Danışmanı	**: Doç. Dr. Oğuz KAYNAR**

	Unvanı - Adı Soyadı	**İmza**
Jüri Başkanı	**: Prof. Dr. Yılmaz GÖKŞEN**	
Üye	**: Doç. Dr. Oğuz KAYNAR**	
Üye	**: Yrd. Doç. Dr. Ahmet Gürkan YÜKSEK**	

Oy Birliği ☑

Oy Çokluğu ☐

Yunus Emre IŞIK **tarafından hazırlanan** Otomatik Doküman Özetleme Yöntemlerinin Karşılaştırılması **başlıklı tez, kabul edilmiştir. …./…./…….**

Prof. Dr. Ahmet ŞENGÖNÜL
Enstitü Müdürü

ETİK İLKELERE UYGUNLUK BEYANI

Cumhuriyet Üniversitesi Sosyal Bilimler Enstitüsü bünyesinde hazırladığım bu Yüksek Lisans/Doktora/Sanatta Yeterlik tezinin bizzat tarafımdan ve kendi sözcüklerimle yazılmış orijinal bir çalışma olduğunu ve bu tezde;

1- Çeşitli yazarların çalışmalarından faydalandığımda bu çalışmaların ilgili bölümlerini doğru ve net biçimde göstererek yazarlara açık biçimde atıfta bulunduğumu;

2- Yazdığım metinlerin tamamı ya da sadece bir kısmı, daha önce herhangi bir yerde yayımlanmışsa bunu da açıkça ifade ederek gösterdiğimi;

3- Başkalarına ait alıntılanan tüm verileri (tablo, grafik, şekil vb. de dahil olmak üzere) atıflarla belirttiğimi;

4- Başka yazarların kendi kelimeleriyle alıntıladığım metinlerini, tırnak içerisinde veya farklı dizerek verdiğim yine başka yazarlara ait olup fakat kendi sözcüklerimle ifade ettiğim hususları da istisnasız olarak kaynak göstererek belirttiğimi,

beyan ve bu etik ilkeleri ihlal etmiş olmam halinde bütün sonuçlarına katlanacağımı kabul ederim.

Yunus Emre IŞIK

ÖNSÖZ

Yüksek Lisans eğitimim boyunca fikirleri ve desteğiyle beni doğru yönlendiren kıymetli danışman hocam Doç. Dr. Oğuz KAYNAR'a; hayatımın her aşamasında yanımda olan değerli aileme; takıldığım her noktada bana zamanlarını ayırarak yardımcı olan çalışma arkadaşlarım Yasin GÖRMEZ, Mehmet Ali DEVECİ, Mustafa TEMİZ, Ferhan DEMİRKOPARAN ve Murat Fatih TUNA'ya ve son olarak hep yanımda olan çok sevdiğim eşime teşekkürlerimi bir borç bilirim.

Yunus Emre IŞIK

İÇİNDEKİLER

KISALTMALAR

ODÖ	: Otomatik Doküman Özetleme
YSA	: Yapay Sinir Ağları
GA	: Genetik Algoritma
TR	: TextRank
PR	: PageRank
LR	: LexRank
DUC	: Document Understanding Conferences
TDA	: Tekil Değer Ayrışımı
GAA	: Gizli Anlam Analizi
YSA	: Yapay Sinir Ağları
MLP	: Multilayer Perceptron
NB	: Naive Bayes
ROUGE	: Recall Oriented Understudy for Gisting Evaluation
TF	: Terim Frekansı
IDF	: Inverse Document Frequency
XML	: Extended Markup Language

TABLO LİSTESİ

ŞEKİL LİSTESİ

ÖZET

Günümüzde internetin gelişmesiyle beraber makale, haber, web sayfaları gibi bilgi içeren dokümanların tamamı dijital ortamda üretilip saklanmaktadır. Buna ek olarak kullanıcıların yeni içerik girdiği web 2.0 ortamlarındaki artışla birlikte internetteki bilgi içeren doküman miktarı inanılmaz seviyelere yükselmiştir. Çok büyük miktardaki dokümanlar arasından istenilen bilgiye ulaşım zaman alıcı olmasının yanı sıra aranan bilginin gözden kaçırılmasına da neden olabilmektedir. Bu probleme, dokümanın boyutunu düşürürken içerdiği konu ve fikir hakkında bilgiyi aktarabilecek özetleme sistemleri çözüm olabilir.

Otomatik doküman özetleme, verilen bir dokümanın bilgisayar ve algoritmalar vasıtasıyla özetinin hızlı ve objektif şekilde oluşturulmasıdır. Bu işlem yorumlayıcı ve çıkarıcı olmak üzere 2 başlığa ayrılır. Yorumlayıcı özetleme dokümanın ana fikir ve konularının belirlenmesi, bu fikirler çerçevesinde özetin yeni cümlelerle ifade edilmesi sürecidir. Çıkarıcı özetleme ise mevcut doküman içerisinden konuyu iyi şekilde yansıtan cümlelerin belirlenerek özet olarak sunulmasıdır.

Bu yüksek lisans tez çalışmasında literatürde kabul görmüş farklı çıkarıcı otomatik özetleme yaklaşımları detaylı şekilde ele alınmış ve karşılaştırmalı olarak değerlendirilmiştir. Veri seti olarak sadece bir dile bağımlı kalınmayıp dilin özetleme açısından etkileri incelenmiştir. Ayrıca gereksiz kelime temizliğinin ve kelimelerin köklerinin alınmasının özetleme başarısına etkileri farklı dillerde ortaya koyulmuştur. Yapılan uygulamalarda özet başarısını ölçmek için kabul gören Rouge değerlendirme paketi kullanılmıştır. Elde edilen deneysel sonuçlara göre beklenin aksine gereksiz kelimelerin temizlemenin tüm yaklaşımlarda özetleme başarısını olumsuz etkilediği belirlenmiştir. Ayrıca klasik yöntem diğer tüm yöntemlerden daha başarılı özetler ortaya çıkartmıştır. Türkçe ve diğer dillerdeki dokümanlar 3 farklı tipte ele alınarak, araştırmacılar için ilginç olabilecek istatistiksel sonuçlar ortaya çıkarılmıştır. Bu tez çalışmasının özellikle Türkçe dilindeki gelecek otomatik doküman özetleme çalışmalarına katkıda bulunması amaç ve dileğimizdir.

Anahtar Kelimeler: Otomatik doküman özetleme, özetleme yaklaşımları, çıkarıcı özetleme

ABSTRACT

Today, with the development of the internet, all the documents containing information such as articles, news, web pages are produced and saved in digital environment. In addition, along with the increase in web 2.0 environments in which users post new content, the number of documents containing information on the Internet has increased to incredible levels. Accessing the desired information from this huge amount of documents can be time consuming and moreover resulting in more information being missed. Summarizing systems which can provide information about subject and idea of document besides reducing dimension of it would be the solution of this problem.

Automatic document summarization is a quick and objective way of summarizing a given document through computers and algorithms. This process is basically divided into two headers, the interpreter and the extractor. Interpreter summarization is to determine the main ideas and topics of the document and to express the summary in new terms with the scope of these ideas. Extractor summarization is to dertermine and present sentences that reflect the topic best in the document.

According to obtained experimental results, it was seen that cleaning the stop-words affect adversely to all dataset in different language. Besides that, conventional method that calculates the score of sentence by summing attribute scores acquired higher Rouge scores than others. Turkish and others documents were handled 3 different types and statistical results might be interesting were found out for researchers. We hope this thesis can contribute to next automatic document summarizing works specially in Turkish.

Keywords: Automatic document summarization, summarization approaches, extractive summarization

1. GİRİŞ

Tarih boyunca insanoğlunun en çok değer verdiği varlıklardan birisi de bilgidir. Eski çağlarda çivi yazısı ile duvarlara bırakılan bilgiler, yazının icadından sonra metin belgelerinde saklanmaya başlanmıştır. Bunun sonucunda oluşan kütüphaneler, herhangi bir konuda araştırma yapılacağı zaman başvurulan ilk kaynaklardır. Ancak özellikle teknoloji ve internetin gelişmesiyle beraber yeni bilgi içeren her şey elektronik ortamda üretilip saklanmaktadır. Bunun yanı sıra eskiden var olan ve bilgi içeren ses, video, metin gibi farklı ortamlar da dijitalleştirilerek internetten erişilebilir hale getirilmiştir. Ayrıca web 2.0 teknolojisiyle beraber kullanıcıların yeni içerik girebildiği blog, twitter gibi sosyal medyaların artması internet ortamında metin, doküman, web siteleri gibi bilgi içeren kaynakların devasa boyutlarına ulaşmasına neden olmuştur.

Bilgi içeren kaynakların artması yenilik ve çeşitlilik açısından büyük fırsatlar yaratmaktadır. Fakat bu durum verilerin saklanma maliyetinin yükselmesi ve istenilen bilgiye hızlı bir şekilde ulaşılamaması sorununu da beraberinde getirmektedir. Özellikle internet üzerinden herhangi bir konuda araştırma yapıldığı zaman binlerce farklı site karşımıza çıkmaktadır. Bu sitelerden hangisinde tam olarak istenilen bilginin bulunduğunu belirlemek için tamamının okunup incelenmesi gerekebilir. Bu iş, zorluğu ve çok ciddi zaman alması nedeniyle bir problem haline dönüşmektedir. Karşılaşılan bu problem, ilgili dokümanın içerdiği bilgiyi daha kısa şekilde gösteren bir özet yardımıyla çözülebilir. Ancak internetteki metin belgelerinin özetleri genellikle oluşturulmamıştır. Bu özeteler yazar ya da uzman kişiler tarafından tekrardan yazılabilse de milyonlarca yazının incelenip özetlerinin çıkartılması hem maliyet hem de süre açısından etkin bir yöntem değildir. Ortaya çıkabilecek bu olumsuzlukların tamamı kaynakta istenilen bilginin bulunup bulunmadığını hızlı ve verimli bir şekilde ortaya koyabilecek bilgisayar destekli otomatik özetleyici sistemler ile rahatlıkla çözülebilir.

Otomatik doküman özetleme (ODÖ) kısaca bir veya daha fazla dokümanın önemli içeriğini tutarken boyutunu düşürme işlemidir (Lee vd. 2009) . Buradaki doküman bir makale, haber, bilimsel yazı olabilir. Amaç kullanıcıya bu dokümanların

tamamını okumadan araştırdığı bilginin doküman içerisinde olup olmadığını gösterebilecek bir özet sunmaktır. Literatür incelendiğinde, bu işlemi gerçekleştirmek için zaman içinde çok farklı yöntemler denenmiş olsa da genel olarak özetleme yöntemlerinin özellik tabanlı, makine öğrenmesi, kümeleme, gizli anlam ve çizge tabanlı yaklaşımlar şeklinde gruplara ayrıldığı görülmektedir (Saziyabegum ve Sajja 2016).

ODÖ çalışmaları 1960'lı yıllardan günümüze kadar ilginin artarak devam ettiği bir alandır. Bu konuda öncü olarak kabul edilen ilk çalışma Luhn tarafından yapılmıştır. Luhn (1958), özette yer alacak cümlelerin önemini içerdiği kelimelerin frekanslarına göre belirlemiştir. Cümlede çok sık ve çok az geçen kelimeler önemsiz olarak kabul edilip temizlenmiştir. Geriye kalanlar ise anahtar kelimeler olarak belirlenmiştir. Her cümle en az bir anahtar kelime içerecek ve dörtten fazla önemsiz kelime içermeyecek şekilde parçalara ayrılmıştır. Bu parçaların içerdiği anahtar kelime sayısının karesi, parçanın toplam kelime sayısına bölünerek bir skor elde edilmiştir. Bu skor aynı zamanda cümlenin skoru olarak kabul edilmiştir. Skorlarına göre sıralanan cümlelerden en iyilerin belirli bir oranı birleştirilerek özet metin oluşturulmuştur.

Doküman ve cümlelerin öznitelikler ile temsil edilebilmesi, araştırmacıları bilgi niteliği olabilecek farklı öznitelikler aramaya yönlendirmiştir. Baxendale (1958), cümlenin paragraf içerisindeki konumunun önemli olup olmadığını incelemiştir. Yaptığı testlerde önemli cümlelerin %85 oranında ilk, %15 oranında son paragrafta yer aldığını saptayarak literatüre cümle konumu özniteliğini kazandırmıştır. Benzer şekilde Edmundson ve Wyllys (1961) özel kelimeler, başlık kelimeleri, ipucu kelimeleri ve konum gibi öznitelikleri cümle önemini belirlemede kullanarak özetleme alanında yapılacak çalışmalara katkı sağlamıştır. Bununla beraber ilk defa sistem özeti ile orijinal özeti karşılaştırarak değerlendirdiği için referans değerlendirme yöntemi olarak Edmunssonun (1969) çalışması kabul edilir. Sonraki dönemlerde öznitelik çıkarmaya ilgi artarak devam etmiş ve tematik kelime, numerik veri (Lin 1999) , tf-idf skorları (Brandow, Mitze, Rau 1995), cümle uzunluğu (Teufel 1997), negatif/pozitif kelime (Altan 2004) gibi çok farklı niteleyici özniteliklerle cümlelerin skorları hesaplanmıştır. Bunun sonucunda öznitelikleri kullanarak cümleleri seçmeye dayanan özellik tabanlı özetleme yaklaşımı ortaya çıkmıştır.

Zamanla cümleler yapısal ve içerik bazlı birçok farklı tipte öznitelik ile temsil edilebilir hale gelmiştir. Cümleler için çeşitli öznitelik değerleri toplanarak cümlelerin önem seviyesini gösteren skor değerleri hesaplanır ve buna göre özette yer alacaklar belirlenir. Klasik özetleme yöntemlerinde bu özniteliklerin her birisi eşit öneme sahip kabul edildiği için belirli bir seviyeden sonra özet başarısında artış görülmemiştir. Bu nedenle, her bir özniteliğe eşit ağırlık vermek yerine farklı ağırlıklar verme yaklaşımları ortaya atılmıştır (Aristoteles, Ridha, Adisantoso 2012). Bu ağırlık değerleri uzman görüşü doğrultusunda elle belirlenebileceği gibi çeşitli, sezgisel algoritmalar yardımıyla da otomatik olarak belirlenebilmektedir.

Özniteliklerin önemini belirlemek için kullanılan sezgisel yaklaşımlara örnek olarak genetik, sürü parçacık, yapay arı koloni, karınca koloni, ısıl işlem, yapay bağışıklık algoritmaları verilebilir. Klasik yöntemde her bir özniteliğin ağırlığı 1 kabul edilirken, sezgisel algoritmalar özet başarısını artıracak şekilde özniteliklerin ağırlığını belirler. Bunlardan birisi olan genetik algoritmayı Meena ve Gopalini (2015), cümle çıkarımı için kullanmışlardır. Çalışmada cümleleri nitelemek için 21 farklı öznitelik kullanılmıştır. Veri setinin bir kısmı 100 iterasyonlu genetik algoritma modeliyle eğitilerek özniteliklere ait optimum ağırlıklar hesaplanmıştır. Elde edilen ağırlıklara göre test verisindeki cümlelerin skorları çıkarılmış ve en yüksek skora sahip cümleler özet için seçilmiştir. Bir diğer sezgisel algoritma olan parçacık sürü optimizasyonu, Binwahlan ve Salim (Binwahlan, Salim, Suanmali 2009) tarafından özetleme üzerine uygulanmıştır. Cümleleri temsil etmek için 5 öznitelik kullanılan çalışmada 100 iterasyonlu bir model oluşturulmuş ve 100 haber ile eğitilerek optimum öznitelik ağırlığı belirlenmiştir. Model teste tabi tutulduğunda, MsWord Summarizer modülünden daha başarılı sonuçlar elde etmiştir.

Özetlemede kullanılan diğer bir yaklaşım ise makine öğrenmesine dayalı yöntemlerdir. Makine öğrenmesi yöntemlerinin kullanıldığı özetleme çalışmalarında, genelde cümlelerin özette geçip geçmediğini gösteren bir sınıflayıcı model cümlelere ait öznitelikler yardımıyla eğitilmektedir. Danışmalı öğrenme yöntemine göre çalışan bu sınıflandırıcılar; girdi olarak cümlelere ilişkin öznitelik değerlerini kullanmakta, çıktı olarak da ilgili cümlenin özette geçip geçmediği bilgisinden yaralanmaktadır. Eğitim veri setinde yer alan dokümanlarla bir sınıflandırıcı model eğitilerek sonrasında bu model test veri setindeki dokümanların özetini çıkarmak için kullanılmaktadır.

Makine Öğrenmesi ile özetleme çalışması ilk olarak Kupiec vd. (1995), tarafından yapılmıştır. Bu çalışmada cümlelere ait 5 farklı öznitelik Naive Bayes algoritmasıyla eğitilerek bir model oluşturulmuştur. Bu model yeni gelen bir dokümandaki hangi cümlelerin özette yer alıp almayacağını sınıflamaktadır. 188 doküman üzerinde yapılan testte modelin %84 oranında doğru cümleyi seçtiği gözlemlenmiştir. SUMMARIST, C4.5 algoritmasını kullanarak otomatik özetleme yapan makine öğrenmesi tabanlı bir diğer sistemdir. Bu sistemde, bilinenlerin yanı sıra isim, tarih, alıntı ve zamir gibi farklı öznitelikler de sisteme dâhil edilmiştir. Tüm öznitelikler bireysel ve kombinasyonlar olacak şekilde teste tabi tutulmuştur. Sonuçlar incelendiğinde tüm özniteliklerin birlikte kullanıldığı model en yüksek başarıyı elde etmiştir (Hovy, Lin 1998). Diğer bir çalışmada da Svore vd. (2007), cümleleri sıralamak amacıyla yapay sinir ağlarını kullanmışlardır. İkili çiftler şeklinde öğrenen RankNet algoritması ile veri setindeki bütün ihtimaller yapay sinir ağıyla (YSA) eğitilmiştir. Ortaya atılan yöntem CNN haber sitesinden alınan veriler ile test edildiğinde, YSA ile özetleme modelinin varsayılan özetleyiciden (dokümanın ilk 3 cümlesini seçerek oluşturan) daha yüksek skorları elde ettiği saptanmıştır. Bunların dışında destek vektör makineleri (Hirao vd. 2002), MLP, K en yakın komşu (Silva vd. 2015), rastgele orman (John, Wilscy 2013), eş-eğitim (Wong, Wu, Li 2008) gibi pek çok sınıflandırma algoritması da otomatik özetleme sistemi oluşturmak için başarıyla uygulanmıştır.

Kümeleme, özetelme sistemlerinde hangi cümlenin seçileceği konusunda faydalanılan bir diğer yaklaşımdır. Hernandez vd. (2008), klasik çıkarıma dayalı özetleme sistemlerinde benzer anlamdaki cümlelerin özette bulunabileceğini öne sürmüştür. Buna çözüm olarak dokümandaki cümlelerin K-Means algoritması ile kümelenmesini ve her kümeden sadece 1 cümle alarak farklı anlamdaki cümlelerin seçilmesini önermiştir. Böylelikle dokümanın iyi bir şekilde temsil edileceği öne sürülmüştür. Kümelemenin kullanıldığı bir diğer çalışma ise Arapça diline uygulanan iki aşamalı özetlemedir. İlk aşamada dokümanlar K-Means ile kümelenerek benzer olanlar belirlenmiştir. İkinci aşamada ise küme içerisindeki cümlelerin isim tamlamaları çıkarılmıştır. Bu isim tamlamalarının diğerleriyle benzerlik skorlarına göre cümlelere puan verilmiştir. Belirli eşiğin üstündeki cümleler özet olarak kabul edilmiştir (Fejer, Omar 2014).

Literatürde sıkça kullanılan bir diğer yaklaşım, kelimeleri birlikte geçme sıklıklarına göre gruplayan, böylelikle alt anlamı ya da gizli anlamı tespit etmeye çalışan lineer cebir yöntemleridir. Bu cebir yöntemleri ile terim-cümle matrisi şeklinde temsil edilen doküman, matris ayrıştırılmasına maruz bırakılır. Ortaya çıkan matrisler hem cümleler arası hem de terimler arası ilişki hakkında bilgi verir. Bu cebirsel yöntemlerden ilki gizli anlam analizidir. Bu analiz giriş matrisini tekil değer ayrışımı yöntemiyle 3 farklı matrise ayrıştırmaktadır. Oraya çıkan matrislerden ilki terim-kavram, ikincisi tekil değer, sonuncusu ise kavram-cümle ilişkisini ortaya koymaktadır. Kavramların dokümandaki altbilgileri veya gizli anlam ilişkilerini tuttuğu kabul edilmektedir. Tekil değerler ise bu kavramların önem katsayılarıdır. Gong ve Liu (2001), gizli anlam analizi ile dokümanı ayrıştırmış ve önemli cümleleri seçmek için ortaya çıkan matrislerden yararlanmışlardır. Dokümandaki gizli ilişkileri gösteren kavramlar önemine göre sıralı bir şekilde gelmektedir. Bu kavramlar ile en fazla ilişkisi olan cümleler aynı zamanda dokümanı en iyi temsil eden cümleler olarak kabul edilmiştir. Kavram-cümle arasındaki ilişki ise ayrıştırma sonucunda ortaya çıkan üçüncü matrisin hücre değerleriyle belirlenmiştir. Aynı yöntemi Steinberger ile Jezek (2004) kavramın önemini gösteren tekil değerleri hesaba katarak; Muray vd. (2005) ilgili tekil değerin oranını kullanarak; Özsoy vd. (2011) ise kavram ile ilgisiz cümleleri dışarıda tutarak geliştirmişlerdir. Bir diğer matris ayrıştırma yöntemi olan NNMF (Negatif olmayan matris ayrıştırma, Non-Negative Matrix Factorization)'da ise ayrıştırma sonucu açıklayıcı değişkenleri ve aktivasyon katsayısı gösteren 2 matris ortaya çıkmaktadır. Gizli anlam analizinden farklı olarak bu matrislerde negatif bir değer bulunmaz. Ayrıca bu matrisler daha seyrek oldukları için gizli kavramlar daha kolay bir şekilde saptanabilmektedir. Bu yöntem, hem Türkçe hem de İngilizce veri setlerine uygulanmış ve gizli anlam analizinden daha yüksek skorlar elde edilmiştir (Lee vd. 2009; Güran, Bayazit, Bekar 2011).

Yakın tarihte metin içeriğindeki bilgilerin çizge modelleriyle temsil edildiği ve sıralandığı birçok çalışma yapılmıştır. Bunların en ünlülerinden olan PageRank (PR) algoritması (Page vd. 1998), siteye giren ve çıkan bağlantıları kullanarak internet sitelerinin önemini belirleyen bir sıralama algoritmasıdır. Bu algoritmanın popülerlik kazanması, araştırmacıları çizge teorisini farklı doğal dil işleme konularına uygulamaya yönlendirmiştir. Özetleme de bu konulara dâhil olmuştur. TextRank

algoritması, PR algoritmasından esinlenilerek oluşturulan ve özetlemede kullanılan algoritmalardan birisidir. Düğümlere internet siteleri yerine cümleler yerleştirilerek önemine göre sıralanır. Düğümler arasındaki bağlantıların skoru ise cümlelerin birbirlerine benzerliklerine göre belirlenir. Çizge modeli bu parametrelere göre oluşturulduktan sonra özyinelemeli şekilde PR algoritması ile koşturularak her bir cümlenin nihai skoru elde edilir (Mihalcea, Tarau 2004). Çizge yapısını kullanarak özet oluşturulan bir diğer önemli algoritma LexRank'tır. Bu algoritma, cümleleri sıralamak için merkezilik skorunu kullanır. Merkezilik skoru, cümlelerin diğer cümlelerle olan benzerlik sayısına göre belirlenen ve cümlenin önem sırasını belirten bir değerdir. Bu skor hesaplanırken önceden belirlenen bir eşik değeri altında kalan matris değerleri 0'a eşitlenerek önemsiz olanlar temizlenir. Kalan değer sayısı toplanarak ilgili cümlenin merkezilik derecesi elde edilir. Merkezilik derecesi en yüksek cümle, dokümanı en iyi yansıtan cümleler olarak kabul edilir. (Erkan, Radev 2004). Bu iki çalışma çizge ile özetlemede en bilinen çalışmalardır. Bunlar dışında, cümlelerin benzeşmezlik skorlarına göre önemine karar veren (Patil, Brazdil 2007) ya da bir kenarın çok sayıdaki düğüme bağlı olabildiği hipergraf kullanılarak (Bellaachia, Al-Dhelaan 2014) yapılan çizge tabanlı özetleme çalışmaları da literatürde yer almıştır.

Otomatik özetleme literatürü genel olarak bu gruplarda çalışmış olsa da farklı yöntemlerin özet başarısına etkisi de incelenmiştir. Conroy ve O'Leary (2001), sıralı veriler üzerindeki olasılıksal dağılımı göstermek amacıyla kullanılan Saklı Markov modelini, cümlelerin özette geçme ihtimalini hesaplamak için uygulamıştır. Pardo vd. (2003) ise TF-IDF ve anahtar kelime yöntemleri ile cümle skorlarını belirledikten sonra iki aşamalı kural sonucu önemli cümleyi belirten bir sistem oluşturmuştur. Bu sistem uzmanlar tarafından %90 oranında başarılı bulunmuştur. Metin üzerine uygulanabilen her yöntem otomatik özetleme konusuna da uygulanabileceği için Mrmr (minimum redundancy and maximum relevance) (Oufaida, Nouali, Blache 2014), retorik yapı (Azmi, Al-Thanyyan 2012), WordNet (Bellare vd. 2004; Dang, Luo, Zhang 2008), MCMR (maximum coverage and minimum redundant) (Alguliev vd. 2011) gibi yöntemlerde farklı dillerdeki dokümanlara başarıyla uygulanmıştır.

Bu tezde, önemli kabul edilen otomatik özetleme yaklaşımları incelenerek karşılaştırılmış, her bir yöntemin avantaj ve dezavantajları belirlenmeye çalışılmıştır.

Karşılaştırma sürecinde Türkçe ve İngilizce dillerinin yanı sıra birçok farklı dilde veri seti kullanılarak özetleme sistemlerinin dil üzerindeki etkisi de ayrıca incelenmiştir.

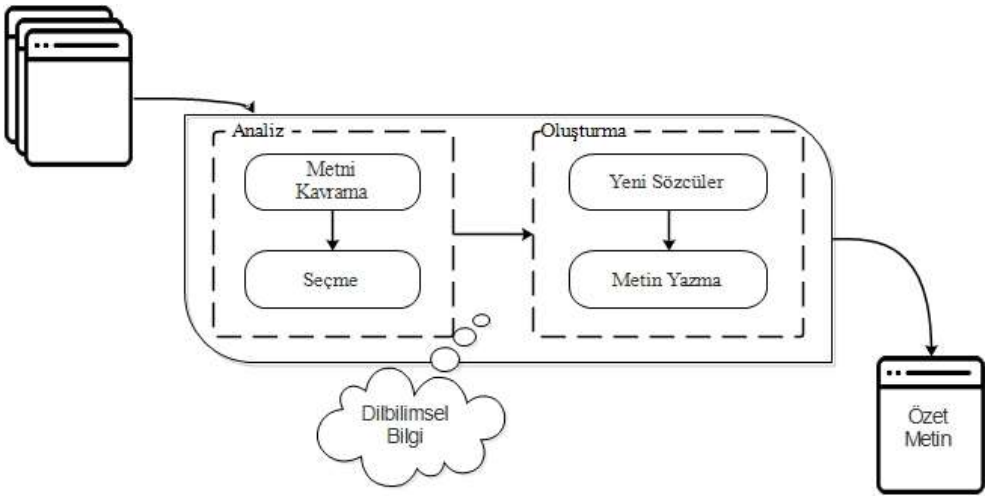
Çalışmanın bir sonraki bölümü olan otomatik doküman özetleme kısmında özet ve özetleme türleri, popüler olan otomatik özetleme algoritmaları, cümle benzerlik yöntemleri geniş bir şekilde ele alınarak örneklerle açıklanmıştır. 3.Bölüm olan uygulama da kullanılan veri setleri, dokümanlara uygulanan ön işlemler, özetleyici sistemlerin değerlendirme ölçütleri ve tez kapsamında açıklanan algoritmaların veri setlerinden elde ettiği sonuçlara yer verilmiştir. Son kısım olan sonuç ve önerilerde, bulgular irdelenmiş, çıkan sonuçlar yorumlanmış ve gelecek çalışmalara yönelik bilgi ve öneriler paylaşılmıştır.

2. OTOMATİK DOKÜMAN ÖZETLEME

2.1. Metin Özetleme

Tanımsal olarak özet, bir dokümanın içeriğini doğru bir şekilde ifade edecek, tercihen yazarları tarafından oluşturulan kısaltılmış metindir (*American National standard for writing abstracts*. 1977). Oluşturulan bu kısa metnin ana işlevi dokümanın içeriği, yapısı, konusu, fikri hakkında bilgi vermektir. Ancak bir özet ne kadar iyi olursa olsun, dokümanda kısaltma işlemi yapıldığı için bilgi kaybı kaçınılmazdır. Burada dikkat edilmesi gereken, bilgi kaybının dokümanın önemli kısımlarından değil önemsiz kısımlarından oluşmasıdır. Bu şekilde kullanıcıya dokümanın temel olarak ne ile ilgili olduğu ve okunmaya değer olup olmadığı hakkında yol gösterir.

Metin özetleme ise girdinin metin belgesi (doküman, haber, internet sitesi, makale vs.) çıktının ise kısaltılmış özet olduğu işlem sürecidir. Bir insan, özet oluşturacağı zaman öncelikle metni okuyarak aklında analiz eder. Edindiği fikirlere göre metnin önemli kısımlarını seçer. Sonrasında yeni veya orijinal dokümandaki sözcükler ile metni tekrar oluşturarak özetleme işlemini tamamlar. Bu aşamalar şekil 1'de gösterilmiştir.



Şekil 1. İnsan Özet Oluşturma Süreci

Özetleme işlemleri kullanıcıların taleplerine göre günümüze kadar şekillenerek farklı tiplerle ele alınmış ve değişik ihtiyaçların çözümüne olanak sağlamıştır. Bu sebeple fonksiyonlarına, giriş doküman sayılarına, içeriğine, dokümanın tipine ve özet yazarına gibi çeşitli kriterlere göre kategorize edilmiş özetleme türleri ortaya çıkmıştır.

Özetleme türleri fonksiyonlarına göre;

- Indicative (belirtici) özet, giriş dokümanın sadece ana konu konuları hakkında bilgi veren özetleme yaklaşımıdır. Bu açıdan içindekiler kısmına benzemektedir. Tam bilgiyi içermeyebilir ancak kullanıcıya okumaya değer olup olmadığı hakkında yardım edebilir.
- Informative (bilgilendirici) özet, dokümanın içeriğini yansıtan, işlenen konuyu açıklayan özetlerdir. Bu özetler orijinal dokümanın kısaltılmış hali olarak tanımlanabilir. Belirtici özetlere göre nispeten daha uzundur.

Doküman sayısına göre;

- Tekil doküman özeti, sadece bir metni giriş verisi olarak alıp o metnin özetini çıkarma işlemidir.
- Çoklu doküman özeti, aynı konu hakkındaki birden fazla sayıda metnin giriş verisi olarak alınması ve konuyu yansıtan bir özet oluşturulmasını amaçlayan özetleme tipidir. Birden farklı kaynaktan elde edilmiş fakat aynı konudaki haberlerin özetlenmesi bu tipe örnek olabilir.

Özet içeriğine göre;

- Genel özetleme, kullanıcının araştırdığı bilgiyi yok sayarak dokümanın genel özetinin çıkarılmasıdır. Bu tipte oluşturulan özet orijinal doküman hakkında olabildiğince bilgiyi içermektedir. Ancak bilgilendirici özetleme tipi kadar uzun olmasına gerek yoktur. Bu özetleme tipi en yaygın kullanılanıdır.
- Sorgu bazlı özetleme, kullanıcı tarafından belirlenen bir sorguya göre oluşturulan özettir. Belirlenen sorgu kelime veya cümle olabilir. Bu özetleme tipi, özet cümlelerini verilen sorguyla ilişkisine göre oluşturur. Böylelikle araştırılan bir konuyla ilgili doküman içerisinde yer alan bilgiler daha kolay bulunabilir.

Dokümanın tipine göre;

- Haberlerin özetlenmesi,
- Belirli bir alanla ilgili dokümanların özetlenmesi (bilim, teknoloji, kimya, hukuk vs. gibi),
- Hikâye, roman gibi edebi metinlerin özetlenmesi,
- Blog, twitter, facebook gibi sosyal medyadaki yazıların özetlenmesi,
- Wikipedia gibi ansiklopedik kaynaklardan elde edilen metinlerin özetlenmesi şeklinde elde edilen kaynak ve kaynağın türüne göre özetlere de ayrılmaktadır.

Özeti yazanın kim olduğuna göre;

- Yazar özeti, orijinal dokümanı yazan kimse tarafından oluşturulan özettir. Bu özeti yazar kendi bakış açısını, hissiyatını kullanarak oluşturmuştur.
- Uzman özeti, yazar dışında dokümanın konusu hakkında bilgisi olan ancak özet yazma konusunda uzman olmayan kimseler tarafından oluşturulur.
- Profesyonel özet ise muhtemelen konu hakkında uzman olmayan ancak özet yazma tekniklerine hâkim, norm ve standartları konusunda bilgi sahibi kimseler tarafından yazılan özetlerdir.

şeklinde ayrılabilir.

Metinlerin özetlerinin çıkartılması birçok açıdan faydalı olabilir. Günümüzde verilerin çokluğu sebebiyle ortaya çıkan depolama problemi, bilginin her yerde paylaşılması sebebiyle hangisinin değerli olduğunun belirlenememesi, araştırma çalışmalarının yoğun zaman alması gibi farklı konularda özetlerden yararlanılmıştır. Ancak her dokümanın özeti bulunmadığı gibi var olan özetlerin konu hakkında uzman kişiler tarafından yazıldığından emin olunamaması, bu özetleri kullanışsız hale getirmiştir. Ortaya çıkan durumda özetleri gerektiğinde hızlı bir şekilde çıkartabilecek, öznel bakış açısından bağımsız ve maliyeti olmayan sistemlere ihtiyaç duyulmuştur. Otomatik doküman özetleme sistemleri tam da bu iş için geliştirilmiştir.

2.2. Otomatik Doküman Özetleme Yaklaşımları

Otomatik doküman özetleme, kaynak doküman hakkında olabildiğince çok ve tutarlı bilgi içeren, sıkıştırma oranının kaynak dokümanın üçte birinden az olması gereken, bilgisayar yazılımları ile kısa metin oluşturma işlemidir (Monreo, Manuel 2014:6). Bu yazılımlar çeşitli algoritmalar ile dokümanı analiz ederek bilgi çıkarımında bulunurlar. Çıkarılan bilgilerin ışığında dokümanın hangi kısmının özet olacağı seçilir veya bilgiler kullanılarak özet oluşturma işlemi sağlanır. Karşılaşılan temel problem, doküman içerisinde hangi kısımların bilgi tuttuğunu belirlemektir. Ayrıca özette anlamsal kopukluklar olmaması için bilgilerden çıkarılacak kısımların birbirleriyle tutarlı olması da önemlidir. Bahsi geçen sıkıştırma oranı ise özetin orijinal dokümanına olan uzunluğuyla ilgili orandır. İhtiyaçlar doğrultusunda belirlenebilecek bu oran genellikle %15 ila %25 arasında tutulmaktadır (Lloret, Palomar 2012). Bilgisayar destekli otomatik özetleme sistemleri genel olarak 2 temel başlık altında ele alınır. Bunlar sırasıyla yorumlayıcı ve çıkarıcı özetleme sistemleridir.

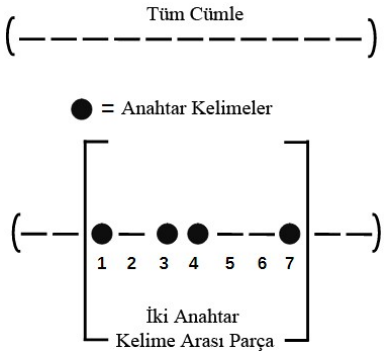
Yorumlayıcı özetleme, dokümanı detaylı şekilde analiz eder; içerdiği bilgiyi, kavramları, konusunu belirler ve bunları kullanarak özeti oluşturur. Yorumlayıcı yaklaşımın ilk aşaması olan bilgi çıkarımında doküman gelişmiş istatistiksel ve dilbilimsel yöntemler ile analiz edilerek dokümana ait alt kavramlar ortaya çıkarılır. WordNet (Fellbaum 1998), VerbNet (Kipper, Dang, Palmer 2000) gibi kelime sözlükleri; kelimeler arasındaki sözdizimsel bağlılığı gösteren bağımlı gramer ve ağaç (Tree) yapıları alt kavramları çıkarmak için kullanılan bazı tekniklerdir. İkinci aşama ise bu alt kavramlar çerçevesinde özetin yeni cümlelerle ifade edilmesidir. Bu süreçte mevcut dokümandan farklı kelime ve cümleler kullanılabilir. Oluşturulan cümlede özne-yüklem uyumuna ve cümle öğelerine (özne, yüklem, bağlaç, zamir vs.) dikkat edilerek anlamsız olması engellenir.

Çıkarıcı özetleme ise doküman içerisindeki parçaların (kelime öbeği, cümle, paragraf vs.) çeşitli istatistiksel yöntemlerle işlenerek önemine göre sıralanmasına dayanmaktadır. Bu yöntemlerle dokümandaki uzunluk, kelime frekansı, ipucu kelime içermesi gibi temel özniteliklere göre veya gizli anlam analizi ve çizge üzerinden elde edilen bilgilerle metin parçaları sıralanabilir. İstenilen özet uzunluğuna göre sıralanan parçalar seçilir, birleştirilir ve özet elde edilir. Çıkarıcı özetlemenin gelişmiş

dilbilimsel yöntemlere ihtiyaç duymaması, karmaşık olan yeniden oluşturma süreci içermemesi gibi nedenler araştırmacıların bu türe yorumlayıcıdan daha fazla ilgi duymasına neden olmuştur. Bu sebeple literatürde çıkarıcı özetleme başarısını arttırmaya yönelik daha çok çalışma bulunmaktadır. Çıkarıcı özetlemeyi 5 farklı grupta toplayabiliriz. Bunlar sırasıyla özellik tabanlı, makine öğrenmesi, kümeleme, çizge tabanlı ve gizli anlam analizine dayalı yöntemlerdir.

2.2.1. Özellik Tabanlı Özetleme Sistemleri

Çıkarıcı özetleme sisteminde, özeti oluşturulacak parçalar (cümle, paragraf, kelime öbekleri) belirli yöntemlerle puanlanır ve bu puanlara göre sıralanır. Literatürdeki ilk çalışma bir tek özniteliğe bağlı kalınarak cümleleri sıralamıştır. Bu çalışmada şekil 2'deki gibi cümlenin içerdiği anahtar kelimelere göre puanlaması esas alınmıştır. Anahtar kelimeler doküman içerisindeki terim frekanslarına göre belirlenmiştir. En sık geçen kelimeler günlük kelimeler olduğu için, en az geçen kelimeler de önemsiz sayılarak temizlenmiş, geriye kalanlar anahtar kelime olarak kabul edilmiştir.



Şekil 2. Luhn'un Özetleme Yaklaşımı

Cümlelerin puanlanmasında en uzun iki anahtar kelimeyi içeren bir pencere kullanılır. Bu pencere içerisindeki anahtar kelime sayısının karesi, kelime sayısına bölünerek cümlenin skoru hesaplanır. En yüksek skoru alan cümleler özette yer alır. Ancak bu algoritma sadece bir tane öznitelik ile cümlelerin önemine karar vermiştir.

Dokümanda anahtar kelime içermesine rağmen dokümanı tam temsil etmeyen pek çok cümle bulunabilir. Ortaya çıkan bu durum, tek bir öznitelik ile değerlendirmenin doğru olmadığını gösterir. Bu probleme çözüm olarak farklı

özniteliklerin cümle seçimine etkisinin kanıtlanması, bu özniteliklerin cümleleri puanlamak için kullanılmasını beraberinde getirmiştir. Özellik tabanlı yaklaşım, bu özniteliklerin beraber kullanılmasından doğan ve cümlelerin sahip oldukları öznitelikler ile temsil edilmesini amaçlayan bir yöntemdir. Birden fazla özniteliğin kullanılması, farklı niteleyiciler ile cümle öneminin belirlenmesini sağlar. Bu öznitelikler çeşitli gruplar altında ele alınabilir. Yapısal olan öznitelikler cümlelerin dış yapısıyla veya görünüşleriyle alakalı olanlardır. Bunlar:

- *Cümle Uzunluğu:* Cümlelerin uzunluğu genellikle cümlenin önemi açısından bilgi taşıyabilir. Uzun cümlelerin içerisinde daha fazla bilgi barındırdığı kabul edilmektedir. Ayrıca bu değerin filtrelenmesiyle tarih belirten satırlar veya yazar isimleri gibi kısımların önüne geçilebilir.

$$Özellik_{Cümle\ Uzunluğu}\ (S_i) = \frac{S_i\ cümlesindeki\ Kelimelerin\ Toplam\ Sayısı}{En\ uzun\ Cümledeki\ Kelimelerin\ Toplam\ Sayısı} \quad (1)$$

- *Cümle / Paragraf Konumu:* Cümlelerin doküman içerisindeki bulundukları konumlar cümlelerin önemlerini belirtebilir. Örneğin, dokümanın ilk/son cümlesi olması veya cümlenin doküman/paragrafta konumu ya da sırası o cümlenin önemli olup olmadığının göstergesidir. Aynı şekilde cümlenin bulunduğu paragrafın konumu da öznitelik olarak kullanılmaktadır. Örneğin, dokümanda 20 cümle olduğunu farz edersek ilk cümle konumu için 20/20, 5.cümle konumu için 15/20 olarak hesaplanabilir.

$$Özellik_{Cümle\ Konumu}\ (S_i) = \frac{Toplam\ Cümle\ Sayısı - S_i\ Cümle\ Konumu}{Toplam\ Cümle\ Sayısı} \quad (2)$$

- *Farklı fontlu / Büyük Harfli Kelimeler:* Doküman içerisinde farklı fontlu (kalın, italik, altı çizili) veya büyük harflerle yazılmış kelimeler genellikle önemli bir konuda bilgi taşımaktadır. Bu kelimelerin bulunması çalışmalarda öznitelik olarak kullanılmıştır.

$$Özellik_{FarklıFont}\ (S_i) = \frac{S_i\ Cümlesinde\ Farklı\ Fontlu\ /\ Büyük\ Harfli\ Kelime\ Sayısı}{S_i\ Cümlesinde\ Toplam\ Kelime\ Sayısı} \quad (3)$$

İkinci grup cümlenin içerdiği kelimelerin öznitelikleridir. Literatürde cümleleri niteleyecek pozitif/negatif kelime, birlikte geçme sıklığı, anlamdaş kelime geçme sayısı gibi farklı öznitelikler bulunsa bile en sık kullanılanlar şöyledir:

- *Terim Ağırlığı(TF/ISF, Term Frequency/Inverse Sentence Frequency):* TF/ISF'in hesaplanma amacı bir kelimenin cümlede ne kadar önemli olduğunu göstermektedir. Bir cümledeki terimlerin var olup olmadığı o cümlenin önemini hesaplamada kullanılan yaygın yöntemlerden birsidir. İlgili cümledeki Terimlerin TF/ISF değerleri toplanarak cümlenin önem derecesi hesaplanabilir.

$$\text{Özellik}_{Terim\ Ağırlığı}\ (S_i) = \frac{S_i\ \text{cümlesinde} \frac{TF}{ISF}\ toplamı}{Maximum\ (\frac{TF}{ISF} toplamı)} \quad (4)$$

- *Başlık Kelimeleri:* Dokümanda yer alan cümlelerdeki kelimeler ile başlıktaki kelimelerin eşleşmesiyle hesaplanan ve cümlenin konu hakkında daha ilgili olup olmadığını gösterebilecek özniteliktir.

$$\text{Özellik}_{Başlık\ Kelimeleri}\ (S_i) = \frac{S_i\ \text{cümlesinde geçen Başlık Kelimeleri Toplamı}}{En\ çok\ geçen\ Başlık\ Kelimeleri\ Toplamı} \quad (5)$$

- *Özel İsim:* En fazla miktarda özel isim içeren cümleler genellikle önemli cümleler olarak görülmektedir. Bir cümlede fazla sayıda özel isim bulunması, o cümlenin özette geçme şansını yükseltir.

$$\text{Özellik}_{\text{Özel İsim}}\ (S_i) = \frac{S_i\ \text{özel isim Toplamı}}{S_i\ \text{cümle uzunluğu}} \quad (6)$$

- *İşaret – İpucu Kelimeler(Cue Word) :* Cümle içerisindeki işaret/ipucu kelimenin olup olmadığıyla hesaplanan özniteliktir. "Sonuç olarak", "Özetlersek" , "Kısaca" gibi içerisinde konu ile ilgili özetleyici bilgiler taşıyan kelimelerin cümlede bulunması, bu cümlelerin önemli bilgiler içerdiğine işarettir.

$$\text{Özellik}_{\text{İpucu Kelime}}\ (S_i) = \frac{S_i\ \text{cümlesindeki işaret kelimelerin Toplam Sayısı}}{S_i\ \text{cümlesindeki Toplam Kelime Sayısı}} \quad (7)$$

- *Tematik Kelimeler*: Tematik (konu ile ilişkili) kelimeler gereksiz kelimeler olan edat, bağlaç dışında dokümanda en fazla geçen kelimeler olarak nitelendirilir. Bu kelimelerin konu ile daha fazla ilişkisi olduğu düşünülür. Bu sebeple ilgili cümlede geçip geçmediği bakılarak cümlenin özette olup olmamasına karar verilir.

$$\text{Özellik}_{Tematik\ Kelime}\ (S_i) = \frac{S_i\ \text{cümlesindeki Toplam Tematik Kelime Sayısı}}{S_i\ \text{cümlesindeki Toplam Kelime Sayısı}} \quad (8)$$

- *Numerik Veri*: Cümle içerisinde bulunan sayılar istatistiksel bir veri veya deneysel bir sonuç hakkında bilgi verebilir. Bu nedenle içerisinde numerik verilerin bulunduğu cümlelerin seçilmesi, özetin konu temsili açısından önem teşkil edebilir.

$$\text{Özellik}_{\text{Nümerik Veri}}\ (S_i) = \frac{S_i\ \text{cümlesindeki Toplam Nümerik Veri}}{S_i\ \text{cümlesindeki Toplam Kelime Sayısı}} \quad (9)$$

Son grup ise cümlenin benzerlik skorlarıyla ilgilidir. Genellikle bir cümlenin diğer cümlelere veya dokümanın diğer parçalarına benzerliği, o cümleyi önemli kılmaktadır:

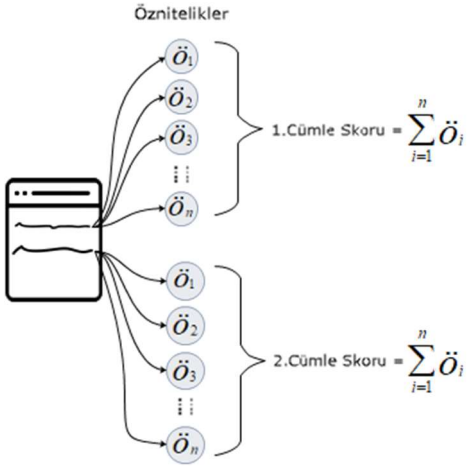
- *Cümle Benzerliği*: N cümleden oluşan bir dokümandaki her bir S_i cümlesinin bir diğeriyle benzerliği hesaplanır ve N x N'lik bir matris oluşturulur. Oluşturulan bu matrisin köşegenlerinde yer alan veriler her cümlenin kendisiyle benzerlik değeri olacağı için köşegenlerdeki değerlere 0 atanır. En son olarak ise her bir cümlenin diğer cümlelerle olan benzerlikleri toplanarak maksimum olan benzerliğe bölünerek cümle benzerliği özelliği hesaplanmış olur.

$$\text{Özellik}_{\text{Cümle Benzerliği}}\ (S_i) = \frac{Toplam\ [(S_i,S_j)]}{Maksimum\ [(S_i,S_j)]}\ i = 1,2,3,\dots N\ ;\ j = 1,2,3,\dots N \quad (10)$$

- *Gizli Anlam Skoru*: Gizli anlam analizi matris ayrıştırmasıyla cümlenin diğer cümlelerle ilişkisini kavramsal uzayda gösterilmesidir. Bu uzayda elde edilen bilgiler ile cümlenin önem skoru çıkartılır. Bu skor birçok çalışmada öznitelik olarak da kullanılmıştır.
- *Çizge Skoru*: Metinlerden bir başka bilgi edinme tekniği, metnin çizge üzerinde temsil edilmesiyle sağlanmaktadır. Cümleler arasındaki bağlantılar kenarlar ile ifade edilerek cümlelerin önemi ile ilgi skor hesaplanabilir. Bu şekilde hesaplanan skorlar (TextRank, LexRank vs. skorlar) öznitelik olarak kullanılabilmektedir.

Çalışmalarda daha farklı özniteliklere de yer verilmiştir. Fakat bunlar mevcuttan türetilmiş ve farklı olmayan özniteliklerdir. Örneğin, cümlenin başlık ile benzerliği gibi bazı çalışmalarda ilk/son cümle benzerliğine bakılmaktadır. Bu öznitelik zaten cümleler arası benzerlik özniteliği ile ortaya çıkarılabileceği için tez

kapsamında gerek görülmemiştir. Öznitelikler belirlendikten sonra cümlenin öneminin hesaplanması aşamasına geçilir.



Şekil 3. Özellik Tabanlı Özetleme Sistemi Çalışma Mekanizması

Özellik tabanlı özetleme sistemlerinde dikkat edilmesi gereken bir diğer husus hangi metin parçalarının kullanılarak özetin oluşturulacağıdır. Literatürde en sık "cümle" kullanılsa da bazı çalışmalarda paragraf ve kelime öbekleri de kullanılmıştır. Gereksiz kelimelerden temizlenmiş ve kökleri alınmış cümleler işlenmeye hazır hale gelir. Bu cümleler öznitelik formülleri kullanılarak analiz edilir ve cümlelerin öznitelik skorları çıkarılır. Her cümlenin numerik öznitelik değeri aynı zamanda cümleye bir puan vermiş olur.

$$Skor(C_i) = ö_{i,1} + ö_{i,2} + ö_{i,3} \dots ö_{i,n} = \sum_{i=1}^{n} ö_{i,n} \qquad (11)$$

Klasik yöntemde cümleye ait öznitelik değerleri belirli bir fonksiyondan geçirilerek, kesin skor elde edilir. Skor fonksiyonu için genellikle eşitlik 11'deki öznitelik değerlerinin toplamı kullanılmaktadır. Her bir cümlenin skoru aynı zamanda dokümanı temsil etme oranı olarak görülebilir. Yani skoru en yüksek olan cümle dokümanın içeriği, konusu ve vermeye çalıştığı fikir hakkında en çok bilgiye sahip olan cümledir. Özetlemenin son aşamasında skoru en yüksek cümleler sıkıştırma oranına göre seçilir. Seçilen cümleler orijinal dokümandaki sırasına göre birleştirilerek özet metin oluşturulur. Örneğin elimizde tablo 1'deki gibi gereksiz kelimelerden temzilenmiş ve kökleri alınmış 7 farklı cümle bulunsun.

Tablo 1. 7 Cümleden Oluşan Örnek 1

C1:	*"petrol altın yüksel"*
C2:	*"uluslararası piyasa petrol altın fiyat yüksel"*
C3:	*"abd tip hafif ham petrol mayıs teslim fiyat cuma gün dolar yüksel sonra bugün art varil dolar işlem gör"*
C4:	*"londra tip brent ham petrol sent art varil dolar sat"*
C5:	*"petrol fiyat yüksel geçen hafta abd istihdam yönelik açıkla veri amerikan merkez banka faiz indir git beklenti art etki"*
C6:	*"petrol ihraç ülke örgüt sekreter abdullah badri yap açıklama piyasa petrol arz yeter ol yüksek petrol fiyat arz sıkıntı ol söyle"*
C7:	*"badri petrol fiyat dolar değer düş petrol rafineri yeter dünya yaşa politik geril yüksel ifade"*

Bu cümleleri; uzunluk, konum, benzerlik, tematik kelime ve TF/ISF toplamı olacak şekilde 5 tane öznitelik kullanarak özellik tabanlı sistemle sıralamaya çalışalım. Bu cümlelere gerekli öznitelik çıkarım işlemleri uygulandığında tablo 2'deki gibi bir matris elde edilir.

Tablo 2. Örnek 1'e Ait Öznitelik Matrisi

	ö1 Cümle Uzunluğu	**ö2 Cümle Konumu**	**ö3 Cümle Benzerlik Skoru**	**ö4 Tematik Kelime**	**ö5 TF/ISF skoru**	***TOPLAM***
c1	0,142857	1	0,975813	0,66666667	0,390009	***3,17535***
c2	0,285714	0,857143	1	0,5	0,550441	***3,193298***
c3	0,904762	0,714286	0,751658	0,31578947	0,964177	***3,65067***
c4	0,47619	0,571429	0,54701	0,3	0,725059	***2,619688***
c5	0,904762	0,428571	0,431894	0,21052632	1	***2,975754***
c6	1	0,285714	0,49197	0,19047619	0,915182	***2,883342***
c7	0,714286	0,142857	0,687185	0,33333333	0,861912	***2,739573***

Matrisin her bir satırı ilgili cümlenin öznitelik vektörünü temsil eder. Bu satırlar, yani cümleye ait öznitelik skorları toplandığı zaman cümlenin önemini belirten skor elde edilir. Özetimizi 2 cümleden oluşturmak istediğimizi varsayarsak en yüksek skora sahip 1. ve 3. cümleler özeti oluşturur:

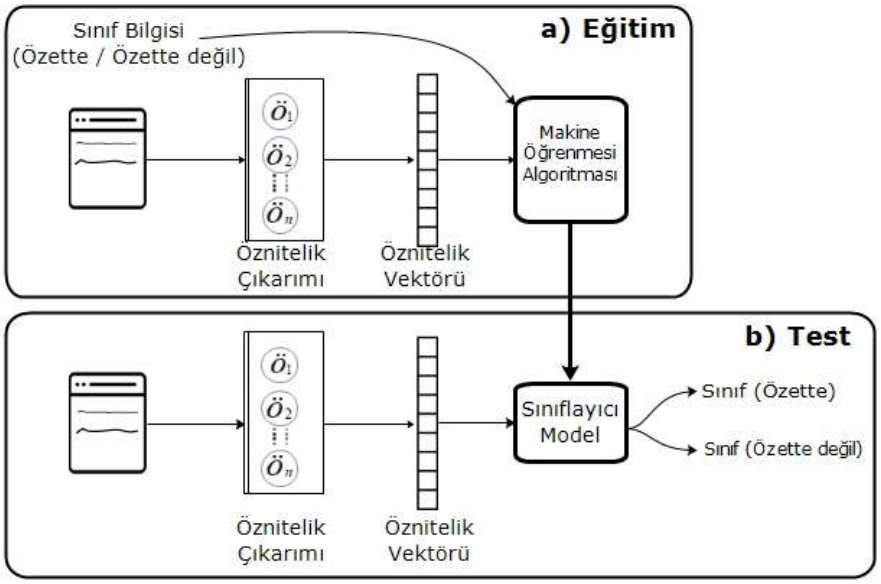
> "Petrol ve Altın Yükselişte. ABD tipi hafif ham petrolün Mayıs teslimi fiyatı, Cuma günü 2,40 dolar yükseldikten sonra bugün de 1,31 dolar artarak varili 107,54 dolardan işlem görüyor."

2.2.2. Makine Öğrenmesi Algoritmaları ile Özellik Tabanlı Özetleme

Klasik özellik tabanlı yaklaşımında, her bir öznitelik diğeriyle eşit öneme sahip kabul edilir. Ayrıca verilen eğitim seti için özniteliklerin gerekli ve gereksiz olduğuna bakılmaksızın hepsi kullanılır. Bu durum özet kalitesinin belirli bir noktadan sonra yükselmesini engelleyebilir. Çözümlerden biri öznitelikleri bilinen cümlenin, örüntü yapısını kendi başına öğrenen makine öğrenmesi yaklaşımı olmuştur.

Makine öğrenmesi, önceden elde edilen veya ortamdaki verilere göre yapıyı öğrenen ve problemin çözümüne göre modelleyen bilgisayar algoritmalarına verilen genel ad olarak tanımlanabilir. Bu algoritmalar, otomatik özetleme sürecinin en önemli aşamalarından birisi olan cümle seçimine katkı sağlamaktadır.

Makine öğrenmesi ile özniteliklerin önemine iki şekilde karar verilebilir. İlki cümlenin önemini verilen özniteliklere göre belirleyen ve özette geçip geçmeyeceğine karar veren sistemdir. Bu açıdan cümle seçim problemi bir sınıflandırma problemine dönüştürülmüş olur. Sınıflandırma ile yapılan cümle seçimi genel hatlarıyla şekil 4'teki gibidir.



Şekil 4. Makine Öğrenmesi Tabanlı Otomatik Özetleme Sistemi

Sınıflandırma siteminde ilk işlem temizlenen cümlelerin vektörlerinin çıkartılmasıdır. Böylece veriler algoritmalar tarafından anlamlandırılabilir hale getirilir. Bir sonraki adım ise algoritmanın eğitimidir. Daha önceden cümlede geçip geçmediğine göre uzmanlar tarafından pozitif / negatif olarak işaretlenmiş cümlelerden oluşan veri setiyle algoritma eğitilir. Eğitim sonunda algoritma, önceki verilerden elde ettiği bilgilere göre ağırlıklarını belirlediği sınıflayıcı modeli oluşturur. Bu model artık yeni gelen bir cümle olduğu zaman, bu cümlenin özette olup olmayacağına karar verir. Sınıflandırma amacıyla kullanılan birkaç algoritmayı şu şekilde sayabiliriz:

2.2.2.1. Naive Bayes

Naive Bayes, Bayes teoreminden faydalanılarak oluşturulmuş sınıflandırma için kullanılan anlaşılabilir ve kolaylıkla uygulanabilir en basit makine öğrenme algoritmalarından biridir. Bu algoritmada tahmin edicilerin (özelliklerin) birbirlerinden tamamen bağımsız olduğu varsayılmaktadır (McCallum, Nigam 1998). Bu varsayım ile tahmin edici özellikler var olan tüm ihtimallere göre değil sadece veri setinden öğrendiklerine göre tahmin işlemi gerçekleştirir. Gerçekte bu bağımsızlık kabul görmese de Bayes algoritması gerçekçi olmayan varsayımla doğrusal problemlerin çözümünde çok iyi performans sergilemektedir.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{12}$$

$P(c \mid x)$ tahmin edici x özellikleri verilince c sınıfında olma olasılığını, $P(x)$ tahmin edicinin, $P(c)$ sınıfın önsel olasılıklarını ve $P(x|c)$ c sınıfında x tahmin edicinin olma olasığılığı göstermek üzere bayes algoritması eşitlik 12'deki gibidir. Algoritmanın sınıflandırmaya uygulanmış halinde ise tahmin edici olarak ilgili verinin öznitelik vektörü kullanılır. Elimizde bir cümleyi temsil etmek için d boyutlu $\{f_1, f_2, f_3, \dots, f_d\} \in f$ vektörü bulunsun. Bu vektörün c_j sınıfında bulunmasının koşullu olasılığı şu şekilde hesaplanır:

$$P(f|c_j) = P(f_1|c_j) * P(f_2|c_j) * \dots P(f_d|c_j) = \prod_{i=1}^{d} P(f_i|c_j) \tag{13}$$

Görüldüğü üzere her bir özniteliğin c sınıfında olma olasılığı çarpılarak ilgili vektörü temsil eden verinin o sınıfta olma olasılığı hesaplanmıştır. Bu algoritmayı örnekle anlatmak için elimizde 100 tanesi spam, geri kalanı normal olan 300 tane mail bulunsun. Bu 100 mail içerisinde "tebrikler" kelimesi 70 defa, "kazandınız" kelimesi ise 50 defa geçtiğini düşünerek yeni gelen "tebrikler kazandınız" içerikli mailin spam olma ihtimalini hesaplayalım:

$$P([\text{tebrikler,kazandınız}]|spam) = P(\text{tebrikler} \mid \text{spam}) * P(kazandınız \mid spam)$$

$$P([\text{tebrikler,kazandınız}]|spam) = \frac{70}{100} * \frac{50}{100} = 0{,}35 \tag{14}$$

Spam olmayan maillerde ise bu kelimeler sırayla 25 ve 20 defa geçsin. Bu durumda gelen mailin normal olma olasılığı:

$$P([\text{tebrikler,kazandınız}]|normal) = \frac{25}{200} * \frac{20}{200} = 0{,}0125 \qquad (15)$$

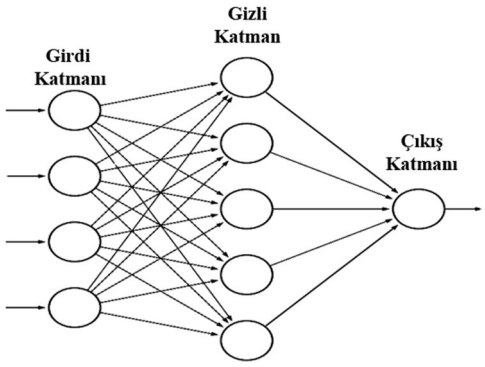
$P([\text{tebrikler,kazandınız}]|spam) > P([\text{tebrikler,kazandınız}]|normal)$ olduğu için bu mail spam sınıfına düşürülür.

Bu yöntem özetleme içinde benzer şekilde kullanılır. Önceden sınıfları bilinen cümlelerle Bayes ağı eğitilerek olasılıklar çıkartılır. Yeni bir doküman geldiğinde ise olasılıklara göre hangi cümlenin özette yer alıp alınmayacağına karar verilir.

2.2.2.2. Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyin ve sinir sisteminin çalışma şeklinden yola esinlenilerek oluşturulan, girdilere dinamik şekilde yanıt verme yoluyla bilgiyi işleyen ve bağlantılarını ortaya çıkartan hiyerarşik bir organizasyondur. Beynin çalışma şeklini taklit eden YSA'nın verilerden öğrenebilme, genelleme çıkarımı ve sınırsız sayıda değişkenle çalışabilmesi gibi farklı avantajları vardır. Taklit edilen sinir hücreleri nöronlar içerirler ve bu nöronlar birbirlerine bağlanarak ağı oluştururlar.

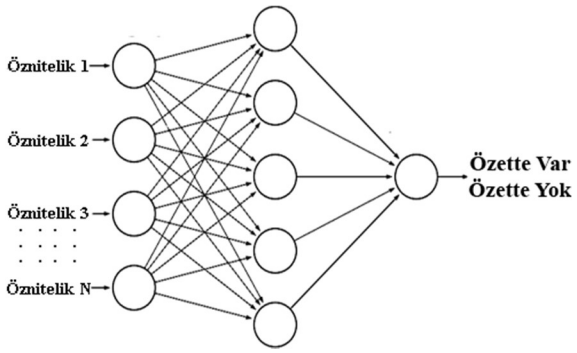
Matematiksel olarak tanımlamak gerekirse YSA, şekil 5'teki gibi numerik değişkenleri nöron içerisinde giriş verisi olarak alıp modeller ve belirli fonksiyonlardan geçirdikten sonra numerik çıktılar üreten yönlü bir ağ yapısıdır (Enríquez, Troyano, López-Solaz 2016). Bahsi geçen fonksiyon aktivasyon fonksiyonu olarak adlandırılır. Nöronlar arasındaki bağlantıların ağırlıkları aktivasyon fonksiyonu ile yinelemeli bir şekilde ayarlanmasıyla yapay sinir ağı öğrenme işlemini gerçekleştirir.



Şekil 5. Yapay Sinir Ağı Örnek Modeli

Oluşturulan yapay sinir ağı tek katmanlı veya çok katmanlı olarak iki farklı şekilde incelenebilir. Tek katmanlı ağ, bir giriş ve bir çıkış katmanından oluşmaktadır. Çok katmanlı ağlar ise bir girdi katmanı, en az bir gizli katman ve çıkış katmanı içerir.

YSA'da en sık kullanılan eğitme yöntemlerinden birisi geri beslemeli modeldir. Bu modelde giriş verisi ağ ile eğitilirken verdiği çıktılar olması gerekenle karşılaştırılır. Çıktı değerleri ile beklenen değerler arasındaki fark hata olarak isimlendirilir. Bu hataya göre ağ üzerinde çıktıdan geriye doğru yayılarak nöronlara arasındaki ağırlıklar değiştirilir. Fark minimize edildiğinde ya da tamamen giderildiğinde elde edilen ağırlıklar, nöronlar arası bağlantılar için en iyi ağırlıklardır.



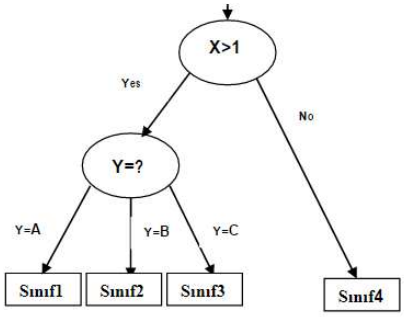
Şekil 6. YSA Metin Özetleme Modeli

YSA ile metin özetleme konusu genellikle öznitelikler kullanılarak ağın eğitilmesine dayanmaktadır. Ağın giriş katmanında cümleleri temsil eden öznitelik vektörleri yer alır. Çıkış katmanı ise özette bulunup bulunmayacağını belirtmektedir. Daha önceden özette yer aldığı bilinen cümleler ile ağ modeli eğitilir. Eğitim sonucunda bir cümlenin özette olup olmayacağını en iyi şekilde tahmin edecek ağırlıklar belirlenmiş olur. Artık model yeni bir cümle ile karşılaştığı zaman bu ağırlıklara göre cümlenin değerini hesaplar ve bu değer hangi tarafa (özette var / özette yok) yakınsa ona göre özette yer alıp almayacağını belirler.

2.2.2.3. Karar Ağaçları

Karar ağaçları, karar problemlinin modellenesi ve çözümlenmesi amacıyla geçmiş verileri kullanarak yeni bir verinin hangi sınıfa ait olduğuna karar vermeye çalışan bir sınıflama algoritmasıdır (Kavzoğlu, Çölkesen 2010). Bu algoritma çizgi, kare, daire gibi geometrik semboller kullanarak karar vericinin problemi kolaylıkla

anlaşılmasını sağlar. Oluşturulan ağaç sorulan sorular ve bunlara verilen cevaplar doğrultusunda hareket eder. Bu soru-cevaplara göre de kurallar oluşturulur.



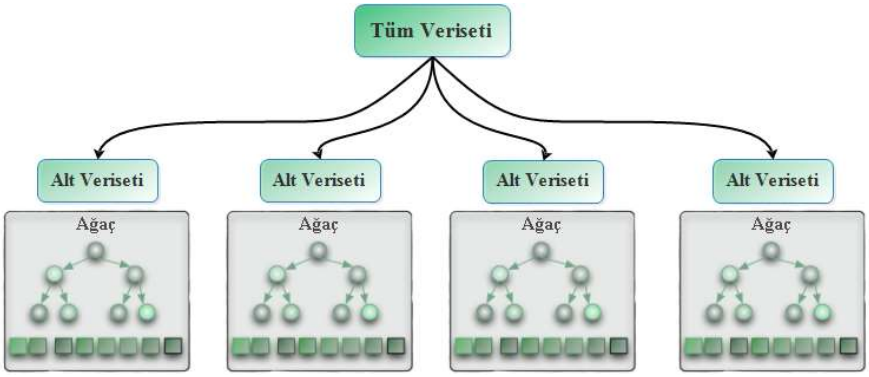
Şekil 7. Örnek Karar Ağacı Modeli

Oluşturulan Ağacın üst kısmı kök düğüm olarak adlandırılır. Bu kök düğüm ilk karar noktasıdır ve en az iki seçenek arasında seçim yapılır. Bu düğüm altında kalan düğümler değişkenlere bağlı olarak gerçekleştirilen durumları temsil eder. Dallar testler ile elde edilen farklı sonuçları, en altta yer alan yaprak düğümleri ise örneklerin yukarıdaki durumlara göre hangi sınıfta olacağını belirtmektedir.

Karar ağacı çalışmasının asıl amacı verilen veri setine göre otomatik olarak en az düğümle en yüksek başarıyı sağlayacak ağacı oluşturmaktır. Bu nedenle zaman içerisinde optimum karar ağacını üretebilmek için ID3, C4.5, CART, C5.0 gibi farklı karar ağacı algoritma yöntemleri geliştirilmiştir.

2.2.2.4. Rastgele Orman

Rastgele Orman algoritması oylamaya bağlı bir toplu öğrenme ve sınıflandırma yöntemidir (Akar, Güngör 2012). Bu yöntem birden fazla karar ağacı sınıflayıcısının birleşimden oluşmaktadır. Buradaki her bir karar ağacı giriş verisi kullanılarak rastgele üretilen alt veri setlerini kullanmaktadır.



Şekil 8. Temsili Rastgele Orman Algoritması Şekli

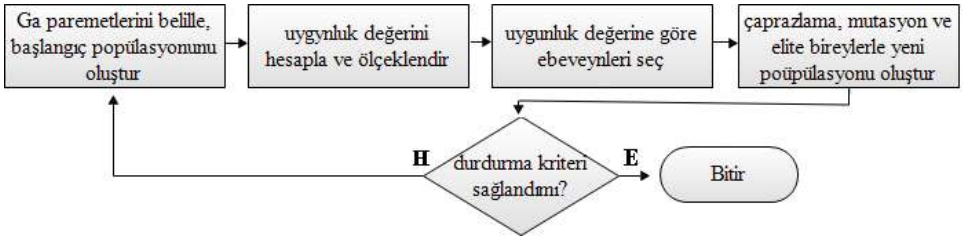
Rastgele Orman algoritmasını başlatılabilmesi için her bir düğümde kullanılacak özellik sayısı (m) ve kaç tane ağaç geliştirileceği (N) olmak üzere 2 parametre belirlenmelidir. Verilerin belirli bir kısmı eğitim seti olarak kullanılarak ağaçlar geliştirilir. Bu ağaçlardaki her bir özellik tüm özellikler içerisinden rastgele olarak seçilir. Böylece bu özellikler ile oluşturulan dallardan en iyisi belirlenir.

2.2.2.5. Sezgisel Algoritmalar

Özniteliklerin önemine karar verilebilecek ikinci yaklaşım ise her özniteliğe farklı ağırlık verilmesidir. Metin açısından daha önemli olan bir özniteliğe daha fazla ağırlık verilerek diğerlerine göre sistemde etkisi arttırılabilir. Böylelikle oluşturulacak özet, ilgili öznitelikten daha fazla yararlanarak daha başarılı olabilir. Bu konuda bazı çalışmalar yapılarak kişiler tarafından belirlenen ağırlıklar ile başarılı özetleme sistemleri oluşturulmuştur. Ancak ağırlıkların kişiler tarafından belirlenmesi çok sayıda deneme işlemi gerektirmektedir. Objektif olmaktan uzak olacak bu yöntem aynı zamanda hangi özniteliğin ne kadar önemli olduğunun bilinmesini gerektirecek uzmanlığa ihtiyaç duyar. Teorik olarak herhangi bir dayanağının bulunmaması da bu problemin çözümü için sezgisel algoritmaların kullanılması fikrini doğurmuştur.

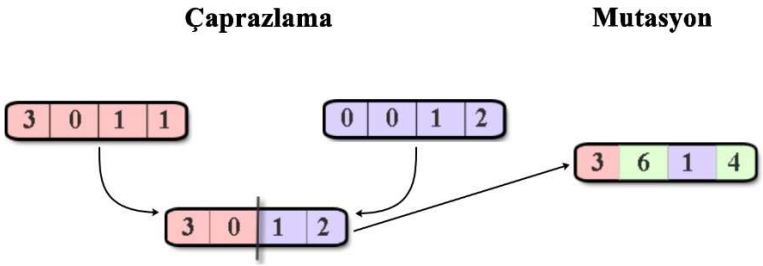
Sezgisel algoritmalar bir probleme kaliteli bir çözüm üretmek için doğal yaşamdaki olaylardan esinlenir. Bu algoritmaların hedefi en uygun çözümü üretmek için amaç fonksiyonunu maksimize ya da minimize etmektir. Amaç fonksiyonu, üretilen çözümün ne derece uygun olduğunu ölçmektedir (Kokash 2005). Bunlar arasında en bilineni genetik algoritmadır.

Genetik algoritma (GA), biyoloji ve evrim teorisinden ilham alan global arama optimizasyon tekniğidir (Holland 1992). Bu teknik daha iyi çözümlerin bulunabilmesi için çözümü kademeli değiştirmektense çözüm popülasyonunu günceller.



Şekil 9. Genetik Algoritma Akış Şeması

Güncelleme esnasında seleksiyon, çaprazlama ve mutasyon işlemleriyle en iyinin hayatta kalması sağlanır. Çaprazlama işlemi bir önceki jenerasyondan gelen iyi niteliklerin kullanılarak yeni genlerin oluşmasını amaçlar. Mutasyon, gen skalasını geniş tutmak için mevcuttan farklı genlerin oluşturulmasıdır. Ayrıca sistematiğe ek olarak rasgele arama operatörü de kullandığından dolayı yerel minimum veya maksimum noktasına takılması önlenir.



Şekil 10. Genetik Algoritmada Mutasyon ve Çaprazlama

Bir özetleme sisteminde genetik algoritma, cümleleri oluşturan özniteliklerin ağırlık değerleri seleksiyon, mutasyon ve çaprazlama gibi teknikleri kullanarak optimize etmeye çalışır. Algoritma içerisinde olası çözümleri muhafaza eden bir popülasyonu belirli şartlar altında veya rastgele olarak üretir. Bu popülasyon içerisindeki hangi çözümün daha iyi olduğuna bir amaç fonksiyonu ile karar verilir. Amaç fonksiyonu bu durumda özetleme için kurulacağından, özetin en iyi olmasını hedefler. Bu sebeple özet kalitesini GA'dan gelen ağırlık parametreleriyle değerlendiren ve bu parametrelerle özetin ne kadar iyi olduğunu belirleyen bir fonksiyona ihtiyaç duyulur. Fonksiyondan dönen gen hayatta kalarak bir sonraki

jenerasyona aktarılır. Jenerasyon, yeni bir popülasyon ile doldurularak işlem tekrarlanır. Belirli bir durdurma kriteri sağlanıncaya ya da popülasyon içerisinde herhangi bir değişiklik olmayıncaya kadar bu işlem devam ettirilir. Elde edilen nihai ağırlıklar cümle özniteliklerine uygulandığı zaman en iyi özeti oluşturan değerlerdir.

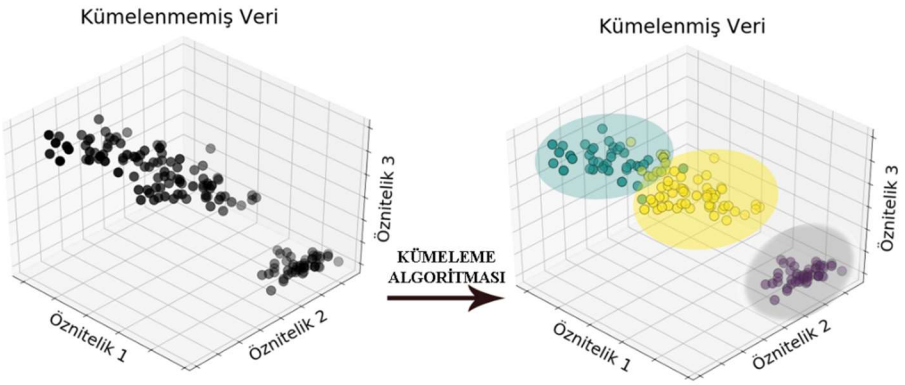
$$Skor(C_i) = w_{i,1} * ö_{i,1} + w_{i,2} * ö_{i,2} + w_{i,3} * ö_{i,3} \dots w_{i,n} * ö_{i,n} = \sum_{i=1}^{n} w_{i,n} * ö_{i,n} \quad (16)$$

Test sürecinde ise eşitlik 16'daki gibi bu ağırlıklar genetik algoritmanın daha önceden görmediği cümle öznitelikleri ile çarpılarak cümle skorları çıkarılır. Skorlarına göre sıralanan cümlelerden en yüksek skora sahip olan özette yer alır.

2.2.3. Kümeleme Yaklaşımı

Kümeleme analizi, verilerin özniteliklerini kullanarak birbirleriyle benzer olanları alt kümelere ayıran ve verinin bütünü hakkında tahmin yapılmasını sağlayan veri madenciliği yaklaşımlarından birisidir (Liu vd. 2010). Başka bir deyişle kümeleme, verileri doğal gruplarına ayırmaktadır. Herhangi bir sınıf etiketine ihtiyaç duymadan uygulanabilmesi sayesinde tıp, ekonomi, bankacılık gibi birçok farklı alanda fayda vermiştir.

Kümeleme yaklaşımları hiyerarşik olan ve olmayanlar olarak ikiye ayrılmaktadır. Hiyerarşik olan kümeleme, veri noktalarını birleştirmeye veya ayrıştırmaya dayanan yöntemlerden oluşmaktadır. Birleştirici yöntemde başlangıçta her bir veri küme gibi kabul edilir ve belirli algoritmalar ile kümeler birleştirilerek üst kümeler oluşturulur. Tam tersi olan ayrıştırıcı kümeleme de ilk başta tüm veriler aynı kümede yer alır ve aşama aşama alt kümelere bölünür.



Şekil 11. Örnek Kümeleme Uzayı

Hiyerarşik olmayan yaklaşımlarda ise veriler belirli kriterlere göre bölümlendirilerek kümelere ayrıştırılır. Bu yaklaşımda oluşturulacak küme sayısı hakkında bir ön bilgi veya önceden küme sayısına karar verilmiş olması gerekmektedir.

Kümeleme işleminin belirli bir veriye uygulanabilmesi için ilk olarak verinin vektör uzayında temsil edilebilmesi yani vektörleştirilmesi gerekmektedir. Vektörden elde edilen benzerlik bilgilerine göre verilerin aynı küme içerisinde yer alması sağlanır. Verilerin vektörleri için genellikle öznitelikler kullanılmaktadır.

Otomatik doküman özetleme sistemlerinde kümelemenin görevi diğer yöntemlere göre farklıdır. Genellikle özetleme için dokümandaki skoru en yüksek veya diğerleriyle benzerliği en fazla olan cümle çıkarılmaya çalışılır. Özet bu bağlamda anlamsal olarak aynı cümlenin tekrarından oluşur ve fazla bilgi içermez. Özellikle aynı konuda birden fazla dokümanı giriş verisi olarak alan çoklu doküman özetleme sistemlerinde böyle problemlerle daha sık karşılaşılır. Kümeleme yönteminde ise benzer cümlelerin kümelenerek her kümeden sadece bir cümle alınması sağlanmaktadır. Böylelikle oluşturulacak özet daha heterojen bir içeriğe sahip olur ve doküman hakkında daha fazla bilgi aktarabilir.

Otomatik özetlemenin kümeleme tarafından yapılabilmesi için öncelikle neyin kümelere ayrıştırılacağı belirlenmelidir. Bu parça genellikle cümle olarak seçilir ve cümleler kümelenir. Sistemin kümeleme yapabilmesi için ikinci adım da küme sayısının belirlenmesidir. Bu sayıyı bulmaya yönelik literatürde iki farklı bakış açısı mevcuttur. İlki veri setine hiyerarşik kümeleme yöntemi uygulayarak küme sayısını belirlemektir. Fakat doküman elde edilen sayıya göre bölüneceği için özet istenilen uzunlukta ayarlanamayabilir. Örneğin, 3 cümleden oluşturulmak istenen bir özet için hiyerarşik yöntemle 7 farklı küme ortaya çıkabilir. Yöntemin sağladığı avantajın yanında hangi kümeden cümlenin seçileceği problemi ortaya çıkar.

İkincisi küme sayısının önceden belirlenmesidir. Bu sayı bir uzman tarafından belirlenebileceği gibi istenilen özet uzunluğuyla aynı olabilir. Dolayısıyla oluşturulacak kümelerden birer cümle seçilmesi yeterli olur. Kümeleme ile yapılan çalışmalarda çoğunlukla küme sayısının önceden belirlendiği tür kullanılmıştır. Bu

türde en bilinen algoritma K-Means (K-Ortalamalar) kümeleme tekniğidir (MacQueen 1967).

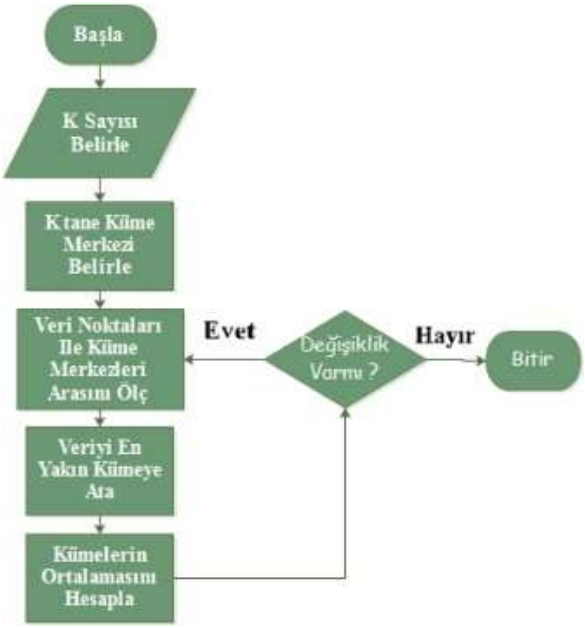
K-Means yöntemi, 50 yılı aşkın süredir en çok kullanılan kümeleme yöntemlerinden birisidir. Bu yöntemin sağladığı en büyük avantaj oluşturulacak küme miktarının önceden belirlenmesidir. Böylelikle veri istenilen miktarda kümeye ayrılır. K-Means yöntemi eşitlik 17'deki amaç fonksiyonunu minimize ederek benzer olan verileri belirli kümeye atamaktır.

$$A_{KM}(X, M) = \sum_{i=1}^{t} \sum_{j=1}^{k} U_{ij} \tag{17}$$

X verileri, M merkez noktaları, k küme sayısı, t veri sayısı, U_{ij} nesne ile küme merkezi arasındaki uzaklığın ölçüsüdür. Uzaklığı belirlemek için kosinüs, manhattan, minkowski gibi iki nokta arası mesafeyi ölçen farklı yöntemler kullanılabilir. Fakat genellikle öklit uzaklığı bu ölçüm için tercih edilmektedir.

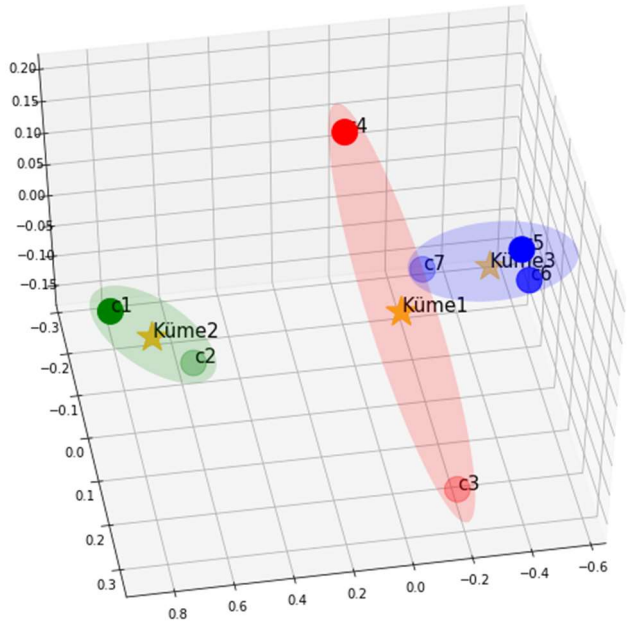
$$\sum_{i=1}^{n} \sqrt{(a_i - b_i)^2} \tag{18}$$

Öklit uzaklığı n boyutlu bir uzayda temsil edilen a ile b noktası arasındaki uzaklığını, bu noktaların vektörel bilgilerini kullanarak hesaplamaktadır. Özetleme sistemlerinde vektörler cümlelerin özniteliklerinden oluştuğu için öklid, öznitelikler arasındaki farka göre kümeyi belirlemektedir (eşitlik 18).



Şekil 12. K-Means Algoritması Akış Şeması

Uzaklık ölçütü için ne kullanılacağı belirlenen bir K-Means algoritması, şekil 12'de gösterilen akış şeması aşamalarından oluşur. 6. adımda görüleceği üzere optimum küme grubu oluşuncaya kadar algoritma optimizasyona devam etmektedir. En uygun kümeleme yerleşimi sağlandığı zaman ise algoritma sonlandırılır.



Şekil 13. Örnek 1'e Uygulanmış Kümeleme Uzayı

Kümeleme ile özetlemeyi göstermek için Örnek 1'de verilen cümlelerimizi K-Means uygulayarak her kümenin en önemli cümlesini bulalım. İlk adım olarak dokümandaki cümlelerin vektörel olarak temsil edilmesi gerekir. Özellik tabanlı özetleme kısmında cümlelerin 5 farklı özniteliği ile vektörleri çıkarılarak tablo 5'teki gibi matris elde edilmiştir. Vektölerin 5 öznitelikle ifade edilmesi aynı zamanda vektör uzayının 5 boyutlu olduğu anlamına gelir.

İkinci adımda bu 5 boyutlu vektör matrisi üzerine k sayısı 3 seçilerek K-Means algoritması uygulanmıştır. 5 boyutlu veriler görsel olarak ifade edilemeyeceği için boyut düşürme yöntemleri ile 3 boyutlu uzayda gösterimi şekil 13'deki gibi sağlanmıştır. Görüleceği üzere ilk küme 3 ve 4; ikinci küme 1 ve 2; üçüncü küme ise 5,6,7. cümleleri kapsayacak şekilde oluşmuştur.

Kümeleme işlemi tamamlandıktan sonra hangi cümlenin kümeyi en iyi temsil ettiği bulunarak cümle çıkarımı adımına geçilir. Kümeyi en iyi temsil eden cümle aynı

zamanda vektör uzayındaki konumu küme merkezine en yakın olan cümledir. Küme merkezinin bulunduğu nokta ile küme içerisindeki verilerin konumları kullanılarak uzaklıklar hesaplanır. Öklid formülü kullanılarak elde edilen uzaklık değerleri tablo 3'te verilmiştir.

Tablo 3. Örnek 1'deki Cümlelerin Küme Merkezliklerine Uzaklıkları

	c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**
Küme 1	-	-	0.515	0.515	-	-	-
Küme 2	0.318	0.318	-	-	-	-	-
Küme 3	-	-	-	-	0.388	0.237	0.604

Örnekte ilk ve ikinci kümede, 2 tane cümle bulunması sebebiyle küme merkezleri bu ikililerin tam ortasında yer almıştır. Cümlelerin küme merkezlik uzaklıklarının aynı olması bununla ilişkilidir. 3. kümede ise 3 cümle vardır ve küme merkezine en yakın olan 6. cümle olarak hesaplanmıştır. Bu değer bilgileri altında ilk kümeden 2., ikinci kümeden 3., son kümeden ise 6. cümle seçilerek özet oluşturulur:

> "Uluslararası piyasalarda petrol ve altın fiyatları yükseldi. ABD tipi hafif ham petrolün Mayıs teslimi fiyatı, Cuma günü 2,40 dolar yükseldikten sonra bugün de 1,31 dolar artarak varili 107,54 dolardan işlem görüyor. Al-Badri, petrol fiyatlarının, doların değerinin düşmesi, petrol rafinerilerindeki yetersizlik ve dünyada yaşanan bazı politik gerilimler nedeniyle yükseldiğini ifade etti."

2.2.4. Gizli Anlam Analizi

Gizli anlam analizi (GAA) metin belgeleri içerisindeki kelimelerin anlamsal düzeyde birbirleriyle ilişkisini gösteren ve çıkartan istatistiksel bir yöntemdir (Landauer, Dumais 1997). Bu yöntem, veriyi vektör tabanlı yaklaşım olan tekil değer ayrışımı ile yeniden yapılandırarak bilgi çıkarımı sağlar. Bu nedenle birçok farklı çalışmada kullanılmıştır. Maillerin sınıflandırılmasında (Strait, Haynes, Foltz 2000), karmaşık problemlerin simüle edilmesinde (Quesada, Kintsch, Gomez 2002), akıllı öğretme sistemlerinde (Graesser vd. 2001), makalelerin sınıfının belirlenmesi (Foltz, Kintsch, Landauer 1998) gibi konularda kullanılabilen Gizli Anlam Analizi, istatistiksel olarak ne kadar güçlü olduğunu göstermiştir (Paul 2007).

Gizli anlam analizinin uygulama aşaması 4 adımdan oluşur. Bu adımlar sırasıyla giriş matrisinin oluşturulması, matrisin vektörleştirilmesi, tekil değer ayrışımı uygulanması ve elde edilen ayrıştırılmış matrislerden bilgi çıkarımıdır. İlk iki adım

metin verilerinin vektörleştirilmesi veya vektör uzayı modelinin oluşturması amacıyla uygulanmaktadır (Dumais 2004). Sonraki aşamalar ise tekil değer ayrışımı ile kelime-kelime, kelime-cümle ve cümle-cümle arasındaki anlamsal bağlantıları ortaya çıkartma sürecidir.

Klasik metin temsil etme yöntemi olan kelime çantasında satırlarda cümlelere sütunlarda ise kelimelere yer verilmektedir. GAA'da ise tam tersi kelimeler satırlarda, cümleler sütunlarda yer almalıdır. Örneklemek gerekirse GAA için n cümleden ve m eşsiz kelimeden oluşan bir doküman için A_{m*n} matrisi şekil 14'teki gibi temsil edilmelidir.

TFIDF	Cümle 1	Cümle 2	Cümle 3
by	0	0	0,552
gunman	0,295	0,434	0,326
is	0	0	0,552
kill	0,499	0	0
killed	0	0,558	0,42
police	0,295	0,434	0,326
the	0,759	0,558	0

Şekil 14. Örnek Kelime-Cümle Matrisi

Satır ve sütunların kesiştiği hücrelerde ilgili doküman ile cümlenin ilişkisini belirten ağırlıklar verilmiştir. Bu ağırlıklandırma türlerine ön işlemler kısmında detaylı şekilde değinilecektir.

2.2.4.1. Tekil Değer Ayrışımı

Çarpanlara ayırma, matris işlemlerinde uzun süredir yararlanılan bir yöntemdir. Bu yöntem bir matrisi birden fazla matrisin doğrusal bir kombinasyonuna dönüştürerek istenilen forma ulaştırılmasını sağlar (Weisstein 2017). En bilinen ayrıştırma yöntemlerinden Tekil Değer Ayrışımı (TDA) (Stewart 1993), temel olarak bir matrisin üç farklı matrisin çarpımına dönüştürülmesidir. Yeni matrisler, orijinal matrisin satır ve sütunları arasındaki ilişkileri ortaya çıkarabilmektedir. Bu nedenle bilgi çıkarımı [4][5], sınıflama (Savas, Eldén 2007) , filtreleme (Konstantinides, Natarajan ve Yovanof 1997) , görüntü ve sinyal işleme (Scharf 1991; Shnayderman, Gusev, Eskicioglu 2006; Andrews ve Patterson 1976) gibi farklı alanlarda Tekil değer ayrışımından yararlanılmıştır (Klema, Laub 1980).

Biçimsel olarak açıklamak gerekirse bir A_{mxn} matrisi TDA ile ayrıştırıldığı zaman ortaya $A_{mxn} = U_{mxm} \sum_{mxn} V^t_{nxn}$ olmak üzere 3 farklı matris meydana gelir. Ortaya çıkan matrislerden U matrisi sol özvektörleri tutan $m \times m$'lik ortogonal, V matrisi sağ özvektörleri tutan $n \times n$'lik ortogonal, $\sum$ matrisi ise giriş matrisi olan A matrisyle aynı boyutta ve özdeğerleri tutan köşegen bir matristir (Kalman 2002:2). Burada belirtilen köşegen matris tipi, köşegenler arası hariç diğer tüm değerleri 0 olan matristir. Ortogonal matris ise $O^{-1} = O^t$ eşitliğini sağlayan yani transpozu tersine eşit olan matrislere denilmektedir. Bu sebeple bu ortogonal matrisin transpozu ile çarpımı birim matris I'a eşit olur.

$$\mathbf{A} = \quad \mathbf{U} \quad \sum \quad \mathbf{V^t}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{m1} & \cdots & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & \cdots & \cdots & u_{mm} \end{bmatrix} * \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & a\sigma_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \cdots & \cdots & \sigma_{mn} \end{bmatrix} * \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & \cdots & \cdots & v_{nn} \end{bmatrix}$$

Şekil 15. Tekil Değer Ayrışımı Matris Gösterimi

U ve V matrisleri ortogonal matrisler olduğu için $U^tU = I$ ve $V^tV = I$ eşitliklerini sağlamaktadır. Bu matrislerin sırasıyla sağ ve sol özvektörleri tuttuğunu ispat etmek gerekirse,

$$AA^t = (U\sum V^t)(V\sum U^t) = U\sum\sum U^t = U\sum^2 U^t \tag{19}$$

$$AA^t = U \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^2 \end{pmatrix} U^t \tag{20}$$

olacaktır. Görüldüğü üzere AA^t matrisinin özvektörleri U iken özdeğerleri $\sigma_1^2 \ldots \sigma_n^2$ olarak sıralanmıştır. Aynı şekilde A^tA matrisi ile V^t'a ait özvektör ve özdeğerler bulunabilir. Bu sebeple tekil değer ayrışımı yaparken U ve V matrislerini elde etmek için özvektör-özdeğer ayrışımı uygulamak gerekir. Bir örnek ile tekil değer ayrışımı açıklanmak istenirse, 2x2 boyutlu bir A matrisimiz bulunsun:

$$A = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \tag{21}$$

Bu matrisin tekil değerlerini bulmak için öncelikle $A^tA - \lambda I = 0$ eşitliğini kullanmak gerekir:

$$A^tA - \lambda I = \begin{bmatrix} 25 & -15 \\ -15 & 25 \end{bmatrix} - \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} 25-\lambda_1 & -15 \\ -15 & 25-\lambda_2 \end{bmatrix} = 0 \qquad (22)$$

Elde edilen matrisin determinantı hesaplanarak $\lambda_1 = 40$ ve $\lambda_2 = 10$ değerleri bulunur. Bunlar özdeğerlerin kareleridir. Kökleri alınarak özdeğerler sırasıyla $\sigma_1 = 6{,}3245$ ve $\sigma_2 = 3.1622$ olarak hesaplanır.

$$\Sigma = \begin{bmatrix} 6{,}3245 & 0 \\ 0 & 3{,}1622 \end{bmatrix} \quad , \quad \Sigma^{-1} = \begin{bmatrix} 0{,}1581 & 0 \\ 0 & 0{,}3162 \end{bmatrix} \qquad (23)$$

A^tA matrisine ait λ değerleri bulunduktan sonra yine bu matrise ait V matrisi elde edilmesi için $(A^tA - \lambda I)$ eşitliğinde sırayla λ değerleri yazılır. $\lambda = 40$ için,

$$(A^tA - \lambda I)v_1 = \begin{bmatrix} -15 & -15 \\ -15 & -15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \qquad (24)$$

eşitliği elde edilir. Bu eşitlikte gerekli hesaplamalar yapılarak A^tA'e ait özvektör hesaplanır.

$$x_2 = -x_1 \quad , \; x_1 = \begin{bmatrix} x_1 \\ -x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad (25)$$

Bulunan değerler vektörün uzunluğuna bölünerek normalize edilir. Bu sebeple öncelikle vektörün uzunluğu hesaplanmalıdır:

$$v_{L_1} = \sqrt{x_1^2 + x_2^2} = \sqrt{1^2 + (-1)^2} = \sqrt{2} \qquad (26)$$

$$v_1 = \begin{bmatrix} x_1/v_{L_1} \\ -x_1/v_{L_1} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 0{,}7071 \\ -0{,}7071 \end{bmatrix} \qquad (27)$$

Bu işlemler aynı şekilde $\lambda = 10$ için de uygularsak

$$v_2 = \begin{bmatrix} 0{,}7071 \\ 0{,}7071 \end{bmatrix} \qquad (28)$$

ikinci özvektör olan v_2 hesaplanır. Bu vektörler V matrisinin sütunlarını oluşturacaktır. Burada dikkat edilmesi gereken önemli hususlardan birisi V matrisi oluşturulurken hangi vektörün kaçıncı sütunda yer alacağıdır. Özdeğerler azalan sırada olduğu için özvektörler ilgili tekil değerin sırasına göre sütun numarası alır. Bizim örneğimiz 2x2'lik olduğu için V matrisinde 2 tane sütun bulunması gerekir.

Özdeğerlerden yüksek olan $\lambda = 40$ ile hesaplanan vektör ilk sütuna, $\lambda = 10$ ile hesaplanan vektör ise ikinci sütuna yerleştirilerek şekil 16'daki gibi V matrisi oluşturulur.

$$V = \begin{matrix} \lambda = 40 & \lambda = 10 \\ \Downarrow & \Downarrow \end{matrix} \begin{bmatrix} 0{,}7071 & 0{,}7071 \\ -0{,}7071 & 0{,}7071 \end{bmatrix}$$

Şekil 16. Tekil Değer Ayrışımıyla V matrisi Oluşturma

V^t ve $\sum$ matrisleri elde edildikten sonra $A = (U\sum V^t)$ eşitliğindeki U matrisi bulunabilir.

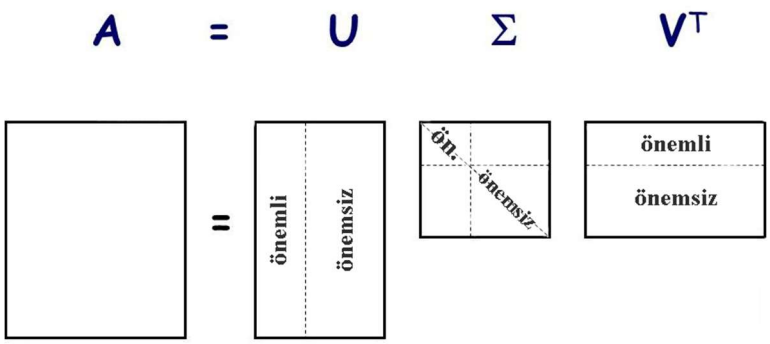
$$U = A V \sum^{-1} \tag{29}$$

$$U = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} 0{,}7071 & 0{,}7071 \\ -0{,}7071 & 0{,}7071 \end{bmatrix} \begin{bmatrix} 0{,}1581 & 0 \\ 0 & 0{,}3162 \end{bmatrix} = \begin{bmatrix} 0{,}4472 & 0{,}8944 \\ 0{,}8944 & -0{,}4472 \end{bmatrix} \tag{30}$$

Son olarak A matrisi U, $\sum$ ve V^t kullanılarak şu şekilde yazılır:

$$A = (U\sum V^t) = \begin{bmatrix} 0{,}4472 & 0{,}8944 \\ 0{,}8944 & -0{,}4472 \end{bmatrix} \begin{bmatrix} 6{,}3245 & 0 \\ 0 & 3{,}1622 \end{bmatrix} \begin{bmatrix} 0{,}7071 & 0{,}7071 \\ -0{,}7071 & 0{,}7071 \end{bmatrix} \tag{31}$$

Bu şekilde TDA ile bir matris 3 farklı matrisin çarpımı tarzında ifade edilmektedir. Bu yöntemin sağladığı diğer büyük avantaj ise matrisin önemli kısımlarını tutarak daha düşük boyuta indirgeyebilmesidir. İndirgeme işleminde hangi verilerin önemli olduğu $\sum$ tekil değer matrisi ile belirlenir. Görüldüğü üzere tekil değerler azalan şekilde sıralanmıştır. Bu değerler aynı zamanda ilgili giriş matrisi için kaç tane önemli kavramın olduğunu göstermektedir. Örneğimiz küçük boyutlu bir matris üzerinde gösterildiği için 2 kavram ortaya çıkmıştır. Ancak çok büyük boyutlu matrislerde belirli bir Rank(kademe)'tan sonra tekil değerler ya çok küçük değere ya da 0'a eşit olmaya başlar. 0 olan değerler matris için hiçbir anlam ifade etmezken, küçük değerlerde önemsiz kabul edilmektedir.



Şekil 17. Tekil Değer Ayrışımında Rank (Kademe)

Değerlerin ve bu değerlerle ilişkili olan U ile V^t matrisindeki satır, sütunların silinmesi matrisi daha düşük boyutta temsil edilmesini sağlar. Özellikle matrisin bir metni temsil ettiği çalışmalarda 2 veya 3 boyutlu uzaya kadar düşürülerek kelimeler, cümleler ve dokümanlar arası ilişki görselleştirilebilir.

2.2.4.2. Tekil Değer Ayrışımı ile Gizli Anlam Analizi

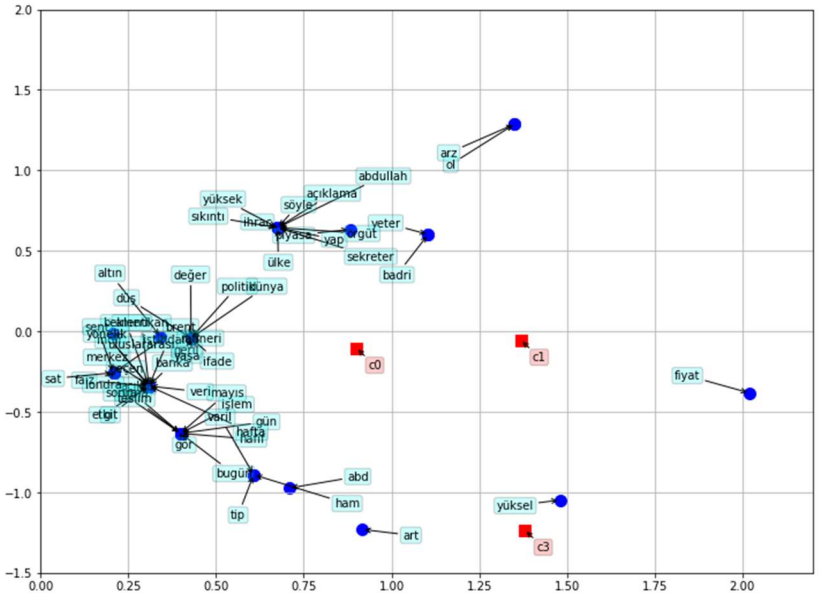
Gizli Anlam Analizi, metnin anlamsal yapısının yakalayarak vektör tabanlı gösterimini sağlayan bir yaklaşımdır. Bu yaklaşımda, metinler arasındaki ilişki içerdiği kelimelere, kelimeler arasındaki ilişki ise beraber geçip geçmediklerine göre modellenebilir. Modelleme aşamasında metin parça (kelime, cümle, doküman) çiftleri arasında vektörel benzerlik karşılaştırılır (Wiemer-Hastings, Wiemer-Hastings, Graesser 2004).

Örnek 2'de yer alan cümlelerin gizli anlam analizini yaparak kelime-cümle ilişkilerini ortaya çıkaralım. İlk olarak bu cümlelere ait terim-cümle vektörleri şekil 18'deki matris ile temsil edilir. Oluşturulan matris şekil 18'deki gibi 61 eşsiz kelimeyi ve 7 cümleyi temsil etmek üzere 61 satır ve 7 sütundan oluşmaktadır.

	c0	c1	c2	c3	c4	c5	c6
abd	0	0	1	0	1	0	0
abdullah	0	0	0	0	0	1	0
altın	1	1	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
örgüt	0	0	0	0	0	1	0
ülke	0	0	0	0	0	1	0

Şekil 18. Örnek 2'e ait Kelime Toplamıyla Oluşturulmuş Terim-Cümle Matrisi

Sonraki adım matrise tekil değer ayrışımı uygulanmasıdır. Elimizde bir m boyutlu matris, TDA ile r boyutlu tekil vektör uzayında temsil edilebilmektedir. r değeri TDA'nın rankını belirtir ve giriş matrisin boyutlarından daha düşük olmalıdır. Bu izdüşürme sonucunda giriş dokümanın gizli anlam yapısı ortaya çıkarılmış olur (Ozsoy 2011). Örnek matrisimize rank 2 seçilerek tekil değer ayrışımı uygulanmıştır. Böylelikle 2 boyutlu uzayda kelimelerin uzaklıkları rahatlıkla görülebilir.



Şekil 19. Kelime ve Cümle İlişkilerini Gösteren Gizli Anlam Uzayı

Şekil 19'te görüldüğü üzere "*arz*" ve "*ol*" kelimeleri sadece bir dokümanda geçtikleri için ilişkileri birbirlerine çok yakındır. Aynı şekilde "*varil*" ve "*ham*" kelimeleri birbirleriyle beraber sık geçtiği için vektör uzayında konumları birbirine yakın izdüşmüştür. Bu yakınlık, kelimeler arasında anlamsal bir bağ olduğunu göstermektedir. "*fiyat*" kelimesi ise neredeyse tüm cümlelerde geçtiği için cümleler açısından bir önem arz etmemektedir. Bunun yanı sıra birçok farklı kelime ile beraber geçmesinin sonucunda herhangi bir kelimeyle anlamsal olarak bağlı değildir. Bu anlamsal ilişkiler hangi cümlenin önemli olduğunu belirlemek için de kullanılmıştır. Önemli cümleleri belirleyebilmesi, otomatik özetleme için gizli anlam analizinden faydalanılmasını sağlamıştır.

2.2.4.2.1. Gong ve Liu Yöntemi

Bir terim-cümle matrisine TDA uygulandığı zaman matris içerisindeki ilişkileri gösteren gizli kavram veya konu adı verilen değerler ortaya çıkar. U matrisi bu kavramlar ile satırlarda yer alan kelimelerin ilişkisini gösterirken, tekil değerler $\sum$ matrisi kavramların önemini azalan şekilde tutmaktadır. Geriye kalan V^t matrisi kavramlar ile sütunlarda yer alan cümlelerin ilişkisini göstermektedir. Ayrıca V^t matrisinin satırları kavram önemine göre azalan düzendedir. Yani ilk satırda bulunan kavram en önemli tekil değerle ilişkiliyken, son satırdaki tekil değer önemsizdir. Gong ve Liu cümlelerin önemini belirlemek için Liu V^t kavram-cümle matrisini kullanmışlardır. Yöntemin uygulanması için öncelikle özetin kaç cümleden oluşacağı belirlenir. Bu değer aynı zamanda TDA yönteminin rank değeri kabul edilir. Yani özeti çıkarılacak dokümana ait terim-doküman matrisi istenilen özet uzunluğu boyutuna düşürülür. Sonraki adımda her bir kavram ile ilişkisi en yüksek olan cümle V^t matrisinden seçilerek özet oluşturulur.

V^t	c1	c2	c3	c4	c5	c6	c7
Kavram 1	0,136	0,206	0,399	0,208	0,309	**0,676**	0,429
Kavram 2	0,022	0,011	**0,634**	0,257	0,34	-0,644	0,042

Şekil 20. Gong ve Liu GAA Yönteminde Cümle Seçini

Örnek 2'e TDA uygulanması sonucu şekil 20'deki gibi V^t kavram-cümle matrisi elde edilir. 2 tane en önemli kavram ve bu kavramlarla en yüksek ilişkisi olan cümleler Gong ve Liu'ya göre dokümanı en iyi yansıtacak cümlelerdir. Bu sebeple özet bu cümlelerden oluşmalıdır. Şekilde görüldüğü gibi ilk kavram ile 6.cümle, ikinci kavram ile 3.cümle en yüksek değere sahiptir. En yüksek değere sahip cümleler özet cümlesi olarak seçilir ve özet oluşturulur:

> "ABD tipi hafif ham petrolün Mayıs teslimi fiyatı, Cuma günü 2,40 dolar yükseldikten sonra bugün de 1,31 dolar artarak varili 107,54 dolardan işlem görüyor. Petrol İhraç Eden Ülkeler Örgütü (OPEC) Genel Sekreteri Abdullah al-Badri de yaptığı açıklamada, piyasaya petrol arzının yeterli olduğunu, yüksek petrol fiyatlarının arz sıkıntısına bağlı olmadığını söyledi."

2.2.4.2.2. Streinberger Yöntemi

Gong ve Liu, matrisin önemini niteleyen kavramları kullanmasına rağmen bu kavramların ağırlıkları olan tekil değerleri hesaba katmamıştır. Bu açıdan her bir kavramın önemi eşit tutulmuştur. Steinberger'e göre ortaya atılan bu yöntemin bir diğer dezavantajı da her bir kavram için sadece bir tane cümle seçilmesidir. Bu dezavantajları aşmak için Steinberger tez çalışmasında "Geliştirilmiş Gizli Anlam Analizi ile Özetleme" yöntemini önermiştir. Bu yöntem cümle seçiminde, gizli kavramlarla beraber bunların ağırlığını belirten tekil değerleri de hesaba katar. Böylelikle doküman açısından daha fazla önem arz eden bir kavramdan birden fazla cümle seçilebilir. Önerilen yöntemde, eşitlik 32'deki gibi her bir cümlenin V_t matrisindeki değerinin karesi, ilgili tekil değerin karesi ile çarpılarak toplanır. Bu şekilde tüm tekil değerler cümle skorunu etkiler. Nihai skorun kök değeri alınarak cümlenin skoru elde edilir.

$$C_k = \sqrt{\sum_{i=1}^{n} v_{k,i}^2 \, \sigma_i^2} \tag{32}$$

Hesaplanan skor o cümlenin doküman açısından önemini belirten skordur. Skorlarına göre azalan şekilde sıralanan cümlelerden, istenilen özet uzunluğu kadar seçilerek dokümana ait özet metin oluşturulur.

Cümle Skorları	
c1	0,89944
c2	1,69499
c3	3,00183
c4	2,21472
c5	2,86461
c6	5,0067
c7	3,75433

Şekil 21. Gelişmiş GAA Örnek Cümle Skorları Tablosu

Örnek 2 üzerine genişletilmiş gizli anlam analizi yöntemi uygulandığı zaman şekil 21'deki cümlelere ait skorlar elde edilir. Yapılan örnekte yine TDA ile matris 2 boyutlu uzaya düşürülmüştür. Görüldüğü üzere Gong ve Liu'nun yönteminde özet için 6.ve 3. cümleler seçilirken, önerilen yöntemde 6 ve 7. cümleler özeti oluşturmuştur

2.2.4.2.3. Çapraz Yöntem

Çapraz yöntem Özsoy vd., tarafından önerilen ve geliştirilmiş gizli anlam analizi ile özetlemenin bir eklentisi olarak kabul edilen bir yöntemdir. Çapraz yöntemde giriş matrisi tıpkı bir önceki önerilen yöntemlerdeki gibi oluşturulur. Sonrasında bu matrise TDA uygulanarak kelime-kavram, tekil değer ve kavram-cümle matrisleri elde edilir. Geliştirilmiş GAA yönteminden farklı olarak her bir kavramın ortalama değerini alarak, bu ortalama değer altında kalan kavram değerleri 0'a eşitlenmiştir.

	V^t Kavram-Cümle Matrisi							*Ortalama*
Kavram 1	~~0,809~~	~~1,873~~	7,011	~~1,905~~	~~4,206~~	20,067	8,095	***6,28086***
Kavram 2	~~0,011~~	~~0,003~~	9,226	~~1,524~~	~~2,654~~	9,523	~~0,04~~	***3,283***
Cümle Skoru	*0*	*0*	***16,24***	*0*	*0*	***29,59***	*8,095*	

Şekil 22. Çapraz Yöntem Kavram-Cümle Matrisi

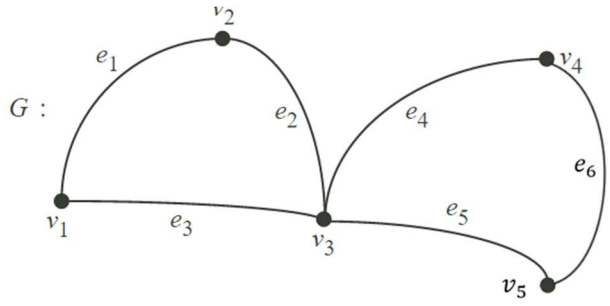
Amaç, bir şekilde ilişkisi olmasına rağmen cümle açısından önemli olmayan kavramların cümleyle bağlantısız hale getirmektir. Böylelikle sadece cümle ile yüksek ilişkisi olan kavramlar hesaba katılmış olur. Örnek 2'e çapraz yöntem uygulanması sonucu elde edilen skorlar şekil 22'deki gibidir. Görüldüğü üzere Geliştirilmiş GAA yöntemiyle yapılan uygulamadan farklı olarak 7.cümle değil 3.cümle seçilmiştir.

2.2.5. Çizge Tabanlı Özetleme Sistemleri

2.2.5.1. Çizge Teorisi

Çizge, birden fazla nesnenin modellenmesi için düğüm (köşe) ve bu düğümleri birbirine bağlayan kenarlardan oluşan veri yapıları olarak tanımlanır. Çizge teoremi ise bu modellemenin matematiksel çalışma alanına verilen isimdir (Mihalcea, Radev 2011:1).

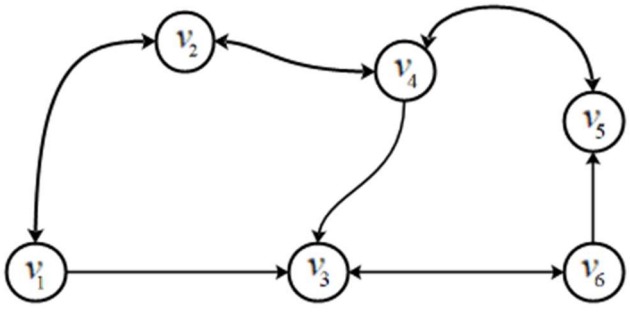
Matematiksel olarak bir G çizgesi V'nin köşeleri ve E'nin köşe ikililerinin bağlantısından oluşan kenarları belirttiği $G = (V, E)$ ile ifade edilmektedir (Ruohonen 2013). Farklı kaynaklarda çizge yerine ağ (network), köşe yerine düğüm (node) ve kenar yerine bağlantı (link) ifadeleri yer alır.



Şekil 23. 5 Düğüm 6 Kenarlı Çizge Örneği

Örneğin, şekil 23'de 5 düğümden oluşan ve bu düğümleri birbirine bağlayan 6 kenardan oluşan G çizgesinin model temsili gösterilmektedir. Eğer bir çizge üzerinde düğümlere yerleştirilen nesneler arasındaki kenar ilişkileri numerik olarak ifade edilebiliyorsa ağırlıklı çizge olarak tanımlanır. Bu ağırlıklar, düğüm ikilileri arasındaki bağlantıyı ölçen değerler olarak kabul edilir ve nesneler arasındaki ilişkinin gücünü belirtir.

Çizge modelleri kenarların yönlerine göre yönlü ve yönsüz olarak 2'ye ayrılır. Eğer kenarlar çift taraflıysa bu yönsüz çizgedir. Örneğin, iki bilgisayarın bağlantılı olduğu bir ağda veri akışı çift taraflı olduğu için yönsüz çizgeye örnek teşkil edebilir. Şekil 24'teki çizge modelinde v_1 ile v_2 arasındaki veya v_4 ile v_5 arasındaki bağlantılar yönsüz ilişkiyi belirtmektedir.

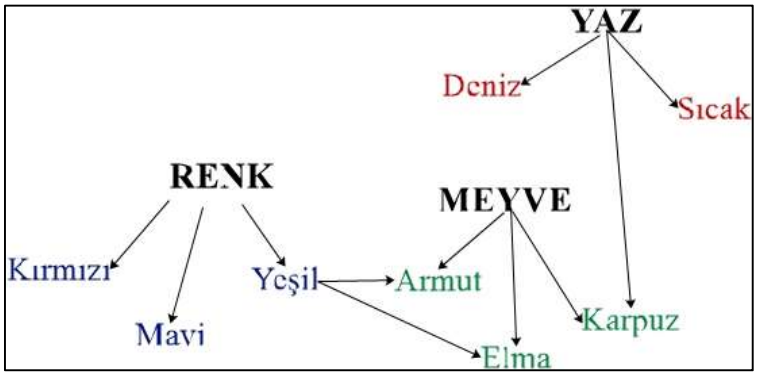


Şekil 24. Yönlü - Yönsüz Çizge Örneği

Yönlü çizgede ise sadece bir taraftan diğer tarafa bağlantı gönderilir. Yine aynı şekildeki v_4'ten v_3'e giden bir bağlantı olmasına rağmen tam tersi bağlantı olmadığı

için yönlü çizge bağlantısıdır. Bu modele seçim oylaması örnek olabilir. Bir kimse diğerine oy verebilir ancak ondan oy almayabilir.

Çizge modellerinden yararlanılarak Ağ sistemleri (Phillips, Swiler 1998), görüntü işleme (Felzenszwalb, Huttenlocher 2004), boyut düşürme (Lee, Verleysen 2012) gibi birçok konu için en kısa yol, gezgin satıcı, minimum yayılan ağaç gibi farklı algoritmalar ortaya atılmıştır. Bu algoritmalar ile düğümlere yerleştirilen nesnelerin ilişkileri bilgi edinmek için çıkartılır ve bu bilgiler kullanılarak nesne hakkında yorum yapılır.



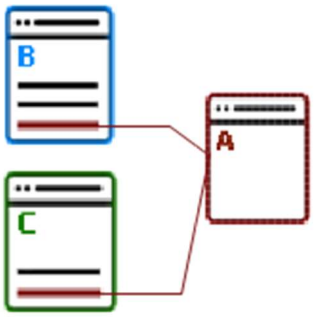
Şekil 25. Çizge Üzerinde Sözcük Türü İlişkileri Modeli

Doğal Dil işleme problemlerinde de çizge ile şekil 25'teki gibi sözcük türü ilişkileri, fikir analizi veya link yapıları kolaylıkla modellenebilmektedir. Bu sebeple çizge modellerinden bu alanda sıkça faydalanılmıştır (Wang vd. 2011; Paredaens, Peelman, Tanca 1995; Agirre, Soroa, Stevenson 2010). Bu alanlardan biriside otomatik doküman özetlemedir.

2.2.5.2. PageRank (PR) Algoritması

PageRank, internet sitelerinin çizge üzerindeki düğümlerine yerleştirilerek öneminin hesaplandığı bir sıralama algoritmasıdır. Bu algoritma web sitelerinin içeriğini değil ağ üzerindeki bağlantılarını kullanır. Bu bağlantılar geri (backlink) ve ileri (forwardlink) linklerdir. Algoritmada temel prensip sitenin popülaritesine göre önemini belirlemesidir. Popülarite siteye gelen bağlantı sayılarıyla doğru orantılı olarak artmaktadır. Örneğin, A sitesinden B sitesine giden bir link varsa, A sitesi B sitesine bir nevi oy vermektedir. Eğer bir site farklı birçok siteden oy alıyorsa popülaritesi yani sıralaması yüksektir. Bunun yanı sıra PR algoritmasının en önemli

katkılarından birisi, sadece bağlantılı site sayısına değil bununla beraber kaliteli bağlantıları da hesaba katmasıdır. Yani bağlantı aldığı sayfaların kaliteli olması, sitenin kendi önemini de arttırmaktadır. Örneğin, *msn.com*'dan bağlantısı olan A sitesi, onlarca önemsiz blogdan bağlantı almış B sitesinden daha yüksek öneme sahip olur. Böylelikle bir site değerlendirilirken diğer sitelerin etkisi de hesaba katılmış olur.

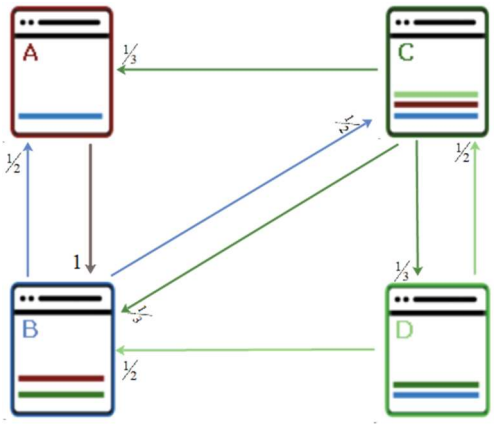


Şekil 26. PageRank Algoritmasıı BackLink-ForwardLink

Bir $G = (V, E)$ çizgesi olduğunu varsayalım. V düğümdeki(vertex) siteleri, E kenarlardaki(edge) bağlantıyı göstersin. A sitesi V 'nin bir elemanı olmak üzere PR değeri eşitlik 33'teki gibi hesaplanmaktadır:

$$PR(A) = (1 - d) + d * \sum_{v \in In(V_i)} \frac{PR(v)}{Out(v)} \quad (33)$$

Eşitlikte belirtilen $In(V_i)$ A sitesine bağlantı veren tüm v sitelerini, $PR(v)$ v sitesinin o an ki PR skorunu, $Out(v)$ ise v sitesinden kaç siteye bağlantı verildiğini göstermektedir. v sitesinin oyu, verdiği bağlantı sayısına bölünerek eşit şekilde dağıtılır. d değeri ise sönümleyici faktör olarak tanımlanmaktadır. 0 ile 1 arasında değer alan sönümleyici faktör kullanıcının bir sayfadan diğer sayfaya geçiş ihtimalini göstermektedir. Başlangıçta her sitenin PR sırası kendisine oy veren site sayılarına göre belirlenir. Sonrasında d değeri kullanılarak sitelerin PR sıralamaları özyinelemeli olarak güncellenir. Belirli bir doyum noktasına ulaştıktan sonra sitelerin sıralaması sabit kalır ve nihai PR sırası belirlenmiş olur.



Şekil 27. PageRank Siteler Arası Link Paylaşımı

Şekil 27'de A, B, C ve D siteleri ve bu sitelerin içerdiği geri bağlantıya (backlink) göre diğer sitelere verdiği puan temsil edilmiştir. Resimde görüldüğü üzere A sitesi, sadece B sitesinin bağlantısını taşıdığı için tüm oyunu B sitesine vermiştir. Benzer şekilde C sitesinde ise diğer tüm sitelerin bağlantıları mevcut olması sebebiyle oyu 3e bölünmüştür. Başlangıçta A ve C siteleri 2'şer siteden bağlantı aldığı için aynı öneme sahipken, PR algoritması ile özyinelemeli olarak hesaplandığında A'nın sıralaması C'den daha üstte yer alır.

2.2.5.3. Çizge ile Özetleme

Metin belgelerinin otomatik özetlemesine ilişkin kullanılan en yaygın yaklaşım içeriğine göre cümlelerin ağırlıklandırılmasıdır. Bu ağırlıklandırmalar kullanılarak en önemli cümle tespit edilmeye çalışılır. Çizge tabanlı özetleme yaklaşımında ise dokümanın önemli kısmı düğümlerden elde edilen bilgilerle belirlenir. Bu amaçla oluşturulan modelde köşeler anlamsal veya bilişsel parçaları (kelime, cümle veya dokümanın tamamı) ile kenarlar ise bu parçaların arasındaki bağlantılar ile özdeşleştirilmiştir. Çizge modelleri ile özetleme genellenirse şu adımlardan oluşmaktadır:

1. Yapılacak uygulamaya en uygun metin parçalarının belirlenip çizge üzerindeki düğümlere yerleştirilmesi,
2. Bu metin parçalarını birbirleriyle bağlamak için aralarındaki ilişkinin belirlenmesi ve bu ilişkiye göre düğümler arasındaki bağlantıların çizilmesi,

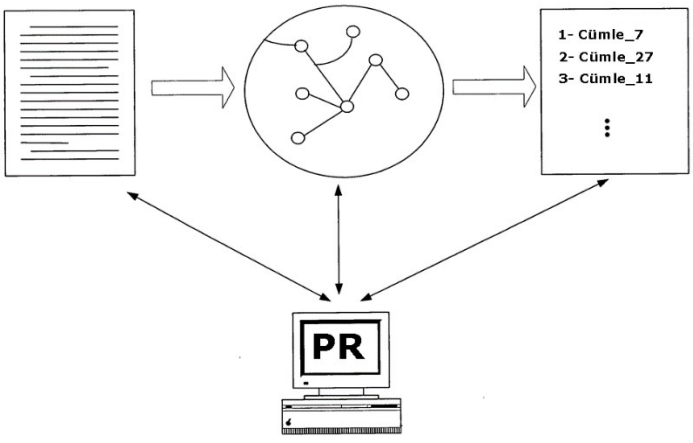
3. İstenilen yakınsaklık skorları elde edilinceye kadar çizge tabanlı algoritmanın çalıştırılması,
4. Elde edilen skorlara göre düğümlerin sıralanması ve bu sıralamaya göre düğümlerdeki metin parçalarının seçilmesi,
5. Seçilen parçaların birleştirilerek özetin oluşturulmasıdır.

2.2.5.3.1. TextRank Algoritması

Çizge modeli ile yapılan ilk özetleme sistemlerinden birisi TextRank (TR) algoritmasıdır. TextRank algoritmasında yukarıda belirtilen ilk adımdaki bir çizge oluşturulur ve çizgenin düğümlerine cümleler yerleştirilir. Cümleler arasındaki bağlantılar ve bu bağlantıların katsayıları ise cümlelerin birbirlerine benzerlik skoru ile ilişkilidir. TextRank algoritmasında varsayılan benzerlik yöntemi eşitlik 34'teki gibidir.

$$Benzerlik\ (V_i, V_j) = \frac{\{w_k | w_k \in V_i\ \&\ w_k \in V_j\}}{\log(V_i) + \log(V_j)} \tag{34}$$

Eşitlikteki V'ler düğümlere yerleştirilen cümleleri temsil etmektedir. w_k'lar iki cümle arasında geçen ortak kelimelerin sayısıdır. Kısacası cümleler arası bağlantı katsayısı, cümle ikilileri arasındaki ortak geçen kelime sayısının cümle uzunluklarının logaritmik değerine bölünerek hesaplanmaktadır.



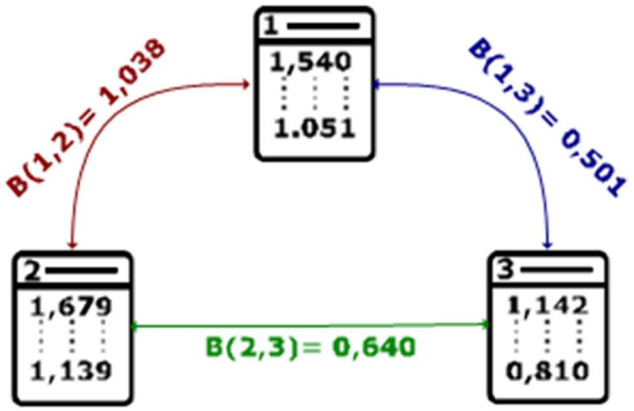
Şekil 28. TextRank Algoritması Çalışma Mantığı

Sonraki aşamada cümleler düğümlerde benzerlik skorları bağlantılarda yer alacak şekilde çizge modeli oluşturulur. Cümlelerin başlangıç skorları, bağlantılı

oldukları cümlelerle benzerliklerinin toplamına eşittir. Oluşturulan model, PageRank algoritmasının metin için özelleştirilmiş hali kullanılarak her bir cümlenin önemi hesaplanır:

$$TextRank(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} * TextRank(V_j) \qquad (35)$$

Eşitlikte belirtilen V_i skoru hesaplanmaya çalışılan cümleyi, V_j bu cümlenin benzerliği bulunan diğer cümleleri belirtmektedir. w'ler ise cümleler arasındaki ağırlığı yani benzerlik skorlarını temsil etmektedir. Sönümleyici faktör d'e bağlı olarak algoritma özyinelemeli şekilde belirli bir doyum noktasına kadar çalıştırılarak cümlelerin nihai ya da başka bir deyişle TextRank skorları elde edilmiş olur. Daha sonra cümleler elde TextRank skorlarına göre sıralanır. Sıkıştırma oranına göre belirlenen sayıda en yüksek puana sahip cümle birleştirilerek özet metin oluşturulur.



Şekil 29. TextRank Algoritması Düğümler Arası Benzerlik Örneği

Örnek 1'deki ilk 3 cümleye varsayılan TextRank benzerlik fonksiyonu uyguladığımız zaman elde edeceğimiz çizge modeli şekil 29'daki gibi olur. Başlangıçta cümlenin skorları, benzerlik skorları toplamına eşit iken TR algoritması ile koşturulduğu zaman ilk cümle 1.051, ikinci cümle 1.139 ve üçüncü cümle 0.810 puan elde eder. Bu skorlara bakıldığı zaman üç cümle içerisinden dokümanı en iyi temsil eden ikinci cümle olarak karar verilir.

TextRank algoritmasının belirleyici nokta cümleler arasındaki bağlantıyı gösteren benzerlik formülüdür. Cümlelerin birbirleriyle ilişkilerini gösterecek farklı benzerlik yöntemleriyle daha başarılı sonuçlar elde edilebilir. Bu amaçla metin verileri arasındaki benzerliği ortaya çıkartacak farklı benzerlik yöntemleri de TextRank

algoritmasıyla beraber kullanılabilir. En sık kullanılan benzerlik yöntemlerini şu şekilde sıralayabiliriz:

2.2.5.3.1.1. Kosinüs Benzerliği

İki vektör arasındaki açı, vektörler arasındaki farklılığı ölçmek için kullanılır. Bu açının kosinüsü ise sayısal benzerliği belirtir. Bununla beraber eğer vektörler belirli birim uzunluğa sabitlenirse, bu vektörler arasındaki kosinüs açısı iki vektörün iç çarpımına eşit olacaktır.

$$\cos(\Phi) = \frac{\overrightarrow{S_i} * \overrightarrow{S_j}}{\|\overrightarrow{S_i}\| \, \|\overrightarrow{S_j}\|} \tag{36}$$

$\overrightarrow{S_i}$ ilk ve $\overrightarrow{S_j}$ ikinci cümlenin vektörünü belirtmek üzere iki cümle kosinüs değeri eşitlik 36'daki gibi iç çarpım ile hesaplanır. Bu iki cümle arasındaki kosinüs değeri ne kadar az ise iki cümle birbirine o kadar benzemektedir.

2.2.5.3.1.2. Jaccard Benzerliği

Diğer bir benzerlik ölçütü olan Jaccard benzerliği ise eşitlik 36'teki gibi iki cümle arasındaki örtüşen kelime miktarının yine o iki cümlenin uzunlukları toplamına bölünmesiyle hesaplanır.

$$Jaccard(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{37}$$

İki cümle arasındaki Jaccard ölçütü eğer 0'a eşitse bu iki cümle hiç benzemiyor, eğer 1'e eşitse tam benzer demektir.

2.2.5.3.1.3. En uzun ortak alt küme benzerliği

En uzun ortak alt küme (Longest Common Subsequence, LCS), iki dizi arasındaki en uzun ortak elemen altkümesini bulmayı amaçlayan bir metottur. Buradaki şart, ortak altkümelerin sıralı olmasıdır. Örneğin, A=$\{a, c, d, e\}$, B=$\{d, a, c, e\}$ olmak üzere iki tane dizimiz bulunsun. Bu iki dizinin sıralamaları bozulmadan içerdiği ortak altdiziler = $\{(a, c), (a, e), (c, e), (d, e), (a, c, e)\}$'dir. Bu dizilerden en uzun olan (a, c, e) ise bu iki dizinin en uzun ortak alt kümesidir. Aynı zamanda en uzun ortak küme uzunluğu 3, iki dizi arasındaki LCS skorunu belirtmektedir. Çoğu zaman bu yöntem, doğal dil işleme konularında iki metnin benzerliğinin ölçümünde

kullanılmıştır. Tez kapsamında ise LCS benzerliği çizge üzerindeki iki düğümün bağlantısını hesaplamak için uygulanmıştır.

2.2.5.3.2. LexRank Algoritması

LexRank bir dokümanın özetini çıkartmak için çizge yapısını ve PR algoritmasını kullanan diğer bir önemli algoritmadır. Bu algoritmanın ağırlıklı ve ağırlıksız çizgeler için 2 farklı modeli mevcuttur.

Ağırlıksız modelde temel varsayım dokümanı yansıtan önemli cümlelerin merkezi olmasıdır. Merkeziliğin hesaplanması için iki parametreye ihtiyaç duyulmaktadır. Bunlardan ilki iki cümle arası benzerlik skoru, ikincisi ise bu skorlar kullanılarak cümlenin genel merkezilik skorudur.

$$\text{idf} - \cos(\text{x}, \text{y}) = \frac{\sum_{w \in x,y} tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\sum_{x_i \in x} (tf_{x_i,x} idf_{x_i})^2} * \sqrt{\sum_{y_i \in y} (tf_{y_i,y} idf_{y_i})^2}} \quad (38)$$

$tf_{w,j}$ j cümlesinde geçen w kelime sayısı, idf_w w kelimesinin ters doküman frekansını belirtmek üzere iki cümle arasındaki skor için eşitlik 38'deki özelleştirilmiş kosinüs benzerliği kullanılmıştır. Dokümandaki her cümle ikilisi için bu işlem tekrarlanarak $A = N * N$'lık benzerlik matrisi oluşturulmuştur.

Bir sonraki aşama ise cümlelerin merkezilik skorlarını hesaplamaktır. Merkezi olmayan önemsiz değerleri temizlemek için bir eşik değeri belirlenir. A matrisinde bu eşik değeri üstünde kalanlara 1, diğerlerine 0 verilerek merkezi olmayan değerler temizlenir. Bunu daha iyi açıklamak için örnek 2'deki cümlelerin özelleştirilmiş kosinüs benzerlik matrisini çıkartalım:

	c1	c2	c3	c4	c5	c6	c7
c1	1	0,4973	0,0111	0,0068	0,0099	0,0111	0,018
c2	0,4973	1	0,0086	0,0034	0,0078	0,082	0,0125
c3	0,0111	0,0086	1	0,2393	0,0529	0,0032	0,0433
c4	0,0068	0,0034	0,2393	1	0,0229	0,0032	0,0305
c5	0,0099	0,0078	0,0529	0,0229	1	0,0029	0,0046
c6	0,0111	0,082	0,0032	0,0032	0,0029	1	0,0746
c7	0,018	0,0125	0,0433	0,0305	0,0046	0,0746	1

Şekil 30. LexRank Algoritması Benzerlik Matrisi Örneği

Eşik değeri olarak 0,05 seçtiğimiz zaman merkezilik dereceleri tablosu şekil t'deki gibi olacaktır.

	c1	c2	c3	c4	c5	c6	c7	*Merkezilik Derecesi*
c1	1	1	0	0	0	0	0	***2***
c2	1	1	0	0	0	1	0	***3***
c3	0	0	1	1	1	0	0	***3***
c4	0	0	1	1	0	0	0	***2***
c5	0	0	1	0	1	0	0	***2***
c6	0	1	0	0	0	1	1	***3***
c7	0	0	0	0	0	1	1	***2***

Şekil 31. LexRank Merkezilik Dereceleri Matrisi

Görüldüğü üzere 2, 3 ve 6. cümleler en fazla benzerlik oyunu alarak en merkezi cümleler olarak hesaplanmıştır. Ancak PR algoritmasında belirtildiği gibi sadece oy değil oyun nerden geldiği de önemlidir. Bu sebeple merkezilik derecesi matrisi PR algoritmasıyla koşturularak, merkezilik derecesi daha doğru şekilde saptanmalıdır.

$$p(u) = \frac{d}{N} + (1-d) * \sum_{v \in In(u)} \frac{p(v)}{\deg(v)} \tag{39}$$

v oy veren cümleleri, $\deg(v)$ ise oy veren cümlenin merkezilik derecesi olmak üzere merkezilik derecesi eşitlik 39'daki özelleştirilmiş PR algoritmasıyla tekrar hesaplandığı zaman şekil 32'deki gibi cümlelerin nihai merkez skorları elde edilir.

Cümleler	Merkezlik Skorları
c1	0,119048
c2	**0,166667**
c3	0,154762
c4	0,154762
c5	0,119048
c6	0,130952
c7	0,154762

Şekil 32. LexRank Merkezilik Skorlarına Göre Cümle Önemi

Tabloda da görüldüğü üzere 7 cümle arasından LexRank skoruna göre en önemlisi 2. cümledir.

Ağırlıklı LexRank modelinde ise herhangi bir eşik değeri ile kısıtlanma bulunmaz. Böylelikle olabilecek bilgi kaybının önüne geçilir. Bu model eşitlik 40'daki özelleştirilmiş PR fonksiyonu ile her cümlenin sırasını hesaplamaktadır.

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in In(u)} \frac{idf\text{-}\cos(u,v)}{\sum_{z \in In(v)} idf\text{-}\cos(z,v)} p(v) \tag{40}$$

Bu modelde bir u cümlesinin hesaplaması yapılırken, bu cümleye oy veren v cümlelerinin idf-cos benzerlik toplamı, v cümlelerin diğer cümlelerle olan benzerlik toplamına bölünerek belirlenmektedir.

3. UYGULAMA

3.1. Özetleme Öncesi Ön İşlemler

Otomatik doküman özetleme sistemi birden fazla adımdan oluşan bir süreçtir. Bu sürecin amacı özeti çıkartılacak doküman metninin algoritmalar tarafından iyi bir şekilde işlenebilir hale getirilmesidir. Bu sebeple giriş doküman metni bazı işlemlere tabi tutulur. Bu işlemlerin amacı hem dokümanları belirli bir standarda sokmak hem de dokümanlar içerisindeki bulunan kelimelerden oluşan sözlüğün boyutunu düşürmektir. Böylece hem başarıya etkisi olmayan kelimeler sisteme dâhil edilmemiş olur hem de algoritmanın çalışma süresi azaltılabilir.

3.1.1. Gereksiz Kelimelerin Temizlenmesi

Gereksiz kelime çıkartımı doküman içerisindeki herhangi bir anlam ifade etmeyen ancak sıkça geçen edat, bağlaç gibi kelimelerin temizlenmesidir. Metin içerisindeki bu gereksiz kısımlar temizlenerek hem algoritmanın daha hızlı ve efektif çalıştırılması sağlanmış olur, hem de bu kelimeler sebebiyle çıkabilecek hata payı azaltılması hedeflenir.

3.1.2. Kelime Kökü Çıkartma

Kök bulma (stemming), sözcükleri en yalın hallerine çevirme işlemi olarak tanımlanabilir. Örneğin çoğul ekinin isimlerden atılması, fiil çekim eklerinin fiilden ayrılarak fiil kökünün ayrılması gibi işlemler kök bulma olarak adlandırılır. Kelimeler köklere ayırmaksızın vektör tabanlı işlemlerde kullanıldığında kelimenin kökü ile onun ekli halleri aynı anlama geldiği halde farklı vektörler üretecektir. Bu durum anlamsal açıdan problemlere yol açabileceği gibi gereksiz yere öznitelik sayısının artarak veri setinin büyümesine neden olacaktır. Kök bulma sayesinde kelimelere ilişkin frekans değerleri hesaplanırken kökü aynı olan kelimelerin ayrı bir kelime ya da öznitelik gibi değerlendirilmesinin önüne geçilmiş olur. Belirtilen nedenlerden ötürü kelimeleri köklere ayırma işlemi metin madenciliği uygulamalarında başarımı arttıran ve sıkça başvurulan ön işleme adımlardan birisi olarak değerlendirilmektedir.

3.1.3. Kelime Etiketleme

Hangi dil olduğuna bakılmaksızın tüm kelimeler işlevlerine göre belirli bazı gruplara ayrılmıştır. Bu gruplara genel olarak sözcük türleri denilmektedir. Dokümanda yer alan kelimelerin türlerinin (özne, isim, sıfat, yüklem vb) belirlenmesi işlemi POS (part of speech) tagging olarak adlandırılmaktadır. Sözcük türleri doküman içerisindeki herhangi bir kelime ve bu kelimeyle komşu olan diğer kelimeler içinde bazı bilgiler verebilir. Bu sebepten dolayı doküman özetleme yapılmadan önce dokümandaki kelimelerin sözcük türlerinin belirlenmesi faydalı olabilir. Örneğin, özet çıkartılacak dokümandaki benzerlik sadece isim ve fiillerle yapılmak istenebilir. Bu durumda doküman içerisindeki sözcük türü sadece fiil ve isim olan kelimeler tutularak benzerlik ölçülür.

3.1.4. Dokümanın Parçalanması

Doküman özetleme konusunda en önemli adımlardan birisi de dokümanın cümle ya da paragraflara parçalanmasıdır. Özellikle çıkarıcı doküman özetleme yöntemleri için ilgili dokümanın doğru bir şekilde bölümlendirilmesi başarıyı oldukça etkileyen bir faktördür.

$$D = (S_1, S_2, \dots, S_n) \tag{41}$$

Bazı çalışmalarda doküman belirli bir kelime sayısına göre bölümlendirilse de en sık kullanılan cümlelere bölümlendirilmesidir. Böylece her bir cümle ayrı ayrı bir giriş verisi gibi kullanılarak aralarındaki ilişki ve hangi cümlenin dokümanı en iyi şekilde temsil ettiği belirlenebilir.

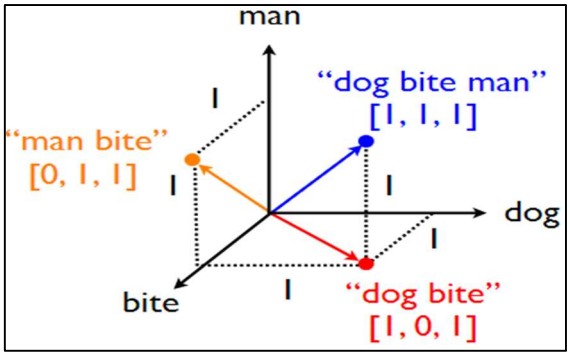
3.1.5. Vektörleştirme

Dokümanın önişlemlerden geçirilmesi sonucunda içerisinde bulunan gereksiz ve başarıyı düşürebilecek etmenler temizlenmiş olur. Bu adımdan sonra artık dokümanın yani metin verisinin algoritmaların anlayabileceği şekilde temsil edilmesi gerekmektedir. Bu işleme dokümanın vektörleştirilmesi denilmektedir. Vektörleştirme işlemi için öncelikle vektörlerin gösterileceği uzay modelinin kurulması gerekir. Vektör uzay modeli doküman ve kelimeleri yeterli büyüklükteki bir boyutta temsil edebilmemizi sağlayan cebirsel bir yöntemdir. Vektör uzayına iz düşürülen kelimelerin arasındaki ilişki temsili vektörler kullanılarak bulunmaya çalışılır. Kelime

çantası (bag-of-words) olarak bilinen bu yöntemde kelimelerin dilbilgisi yapısı veya doküman içerisindeki geçiş sırası önemsiz kabul edilir (*Vector space model* 2017).

	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1
doc_3	0	1	1

Şekil 33. Örnek Dokümanlar Ve Kelime Frekans değerleri



Şekil 34. Örnek Dokümanlara Ait Vektör Uzayı

Örneğin Şekil 33'teki gibi 3 dokümandan oluşan verilerimiz olsun. Eğer ilgili terim dokümanda geçiyorsa 1 geçmiyorsa 0 değerini alsın. Matrisin her bir satırı ilgili dokümanı temsil eden vektör olarak adlandırılmaktadır. Vektörlerden elde edilen bilgilere göre oluşturulan uzay modeli ve bu uzay modelindeki gösterimi şekil 34'deki gibi olur. Oluşturulan vektör uzayında her bir kelimeye dokümandaki önemini temsil edecek şekilde bir ağırlık veya yukarıda belirtildiği gibi değer verilir. Bu ağırlıklandırma 3 faklı şekilde yapılır:

a) ***Binary (ikili) Ağırlıklandırma:*** Eğer kelime dokümanda geçiyorsa 1, geçmiyorsa 0 değerini alır.
b) ***Geçme Sıklığı:*** Kelimenin ağırlığı şekil 35'teki gibi ilgili dokümanda ne kadar geçiyorsa o miktarda olarak belirlenmesidir.

Terim-Frekans Matrisi	by	gunman	is	kill	killed	police	the
Doküman 1	0	1	0	1	0	1	2
Doküman 2	0	1	0	0	1	1	1
Doküman 3	1	1	1	0	1	1	0

Şekil 35. Geçme Sıklığı Kelime Ağırlıklandırma Yöntemi

c) ***Terim Frekansı:*** Terim frekansı kelimenin ağırlığını, ilgili dokümanda geçme sayısını doküman uzunluğuna bölerek hesaplayan bir ağırlıklandırma yöntemidir. Bu yöntem en sık kullanılan kelime ağırlıklandırma yöntemlerinden birisidir.

Terim - Frekans Matrisi	by	gunman	is	kill	killed	police	the
Doküman 1	0	0,2	0	0,2	0	0,2	0,4
Doküman 2	0	0,25	0	0	0,25	0,25	0,3
Doküman 3	0	0,2	0	0	0,2	0,2	0

Şekil 36. Terim Frekansı Kelime Ağırlıklandırma Yönntemi

d) ***Terim Frekansı-Ters Doküman Frekansı ile Ağırlıklandırma:*** Tf-idf yöntemi bir kelimenin doküman veya ilgili veri setinde ne kadar önemli olduğunu belirtmek için kullanılan istatistiksel bir yöntemdir. Bu yöntem ilgili terimin geçme sayısı (TF), yine ilgili terimin dokümandaki seyrekliği (IDF) ile çarpılır. Örneğin, 100 tanesinde "araba" kelimesi geçen 1000 doküman bulunsun.

$$w_{i,j} = tf_{i,j} * idf\left(\frac{N}{df_i}\right) = \left(\frac{10}{200}\right) * \log\left(\frac{1000}{100}\right) = 0.2 \qquad (42)$$

Bunların içerisinden 200 kelimeden oluşan bir dokümanda da 10 kere "*araba*" kelimesi geçsin. Bu durumda ilgili kelimenin ilgili doküman için TF-IDF değeri eşitlik 41 ile hesaplanır.

3.2. Veri setleri

Tez uygulaması dâhilinde toplam 4 farklı veri seti kullanılmıştır. Veri setleri farklı dillerden seçilerek özetleyici sistemin tek dile bağımlı değil global başarısının ölçümü hedeflenmiştir. Ayrıca her bir veri seti 3 farklı şekilde modellenmiştir. Bunlar sırasıyla normal, gereksiz kelimeler temizlenmiş ve kök alınmış modellerdir. Elde edilen sonuçlar ışığında daha önce literatürde rastlanmayan gereksiz kelime ve kök bilgilerinin Türkçe özetleme üzerine etkisi de tezin sonuç kısmında paylaşılmıştır.

3.2.1. DUC 2002 Veri seti

DUC (Document Understanding Conference, Mesaj Anlamlandırma Konferansı), otomatik özetleme sistemlerine artan ilgi sonucu ortaya çıkmış çalıştaylardır. Çalıştayların amacı otomatik özetleme ve değerlendirme sistemlerini daha ileri seviyeye taşımaktır (*Document Understanding Conferences*, 2002). İlki 2000 yılında yapılan bu çalıştay zamanla konferans şeklini almış ve 2007'e kadar devam etmiştir. Çalıştaylar vasıtasıyla otomatik özetleme konusunda yapılacak çalışmaların yıldan yıla yol haritaları belirlenerek daha iyi özetleyici ve değerlendirici sistemlerin medyana getirilmesi hedeflenmiştir. Bu hedeflere ulaşılmasını kolaylaştırmak için her sene ayrıca belirli görevler dâhilinde veri seti ve veri setlerine ait özetlerin katılımcılar tarafından oluşturulması sağlanmıştır. Tez uygulamasında bu veri setlerinden birisi olan DUC 2002 kullanılmıştır.

Veri seti toplam 567 haberden oluşmaktadır. Ancak bazı haberler birbirlerine benzer olması sebebiyle 557 haber tezde kullanılmıştır. Her bir haberin özeti DUC katılımcıları tarafından yeni cümlelerle ifade edilecek şekilde özetlenmiştir. Bu sebeple orijinal dokümanda geçmemesine rağmen özette bulunan farklı kelime ve kelime grupları mevcuttur.

3.2.2. 120 Haber Türkçe Veri seti

Tez uygulama kısmında kullanılacak Türkçe veri setlerinden ilki farklı kaynaklardan elde edilmiş 120 haberden oluşan dokümanlardır. Bu dokümanlara ait özetler, dokümanı en iyi temsil eden cümlelerin veri setini oluşturan kişiler tarafından seçilmesiyle oluşturulmuştur. Bu nedenle özette geçen cümleler, dokümanda geçen cümlelerle birebir aynıdır.

3.2.3. MultiLing Veri seti

Otomatik özetleme sistemlerinde önemli bir başlıkta sistemin dil bağımlı olup olmadığıdır. Eğer özetleme sistemi bir dilde iyi çalışıyorken farklı bir dilde düşük performans veriyor ya da çalışmıyorsa dil bağımlıdır. Özetleme sistemlerinin dil bağımlılığı, doğal dil işleme çalışmalarını geliştirmek için yapılan TAC (Text Analysis Conference) çalıştayında ele alınmış ve dil bağımlılığını ölçmek için MultiLing veri seti oluşturulmuştur.

Tez çalışması dâhilinde MultiLing 2015 (*MultiLing*, 2015) veri seti dil farkının özetleme sistemleri üzerine etkisinin incelenmesi için kullanılmıştır. Veri seti içerisinden seçilen İngilizce, Almanca, Fransızca, İspanyonca, İtalyanca ve Türkçe dillerden her birinde 30 doküman ve dokümanların özetleri bulunmaktadır. MultiLing veri seti içerisinde daha birçok farklı dil olmasına rağmen, diğer dillere ait gereksiz kelime ve kök çıkartıcı yazılım kütüphanelerinin eksikliği dolayısıyla bu 6 dil seçilmiştir.

3.2.4. Habercom Veri seti

Otomatik özetleme sistemleri araştırmacılar tarafından uzun süredir çalışılmış olmasına rağmen Türkçe dil üzerine yapılan çalışma sayısı azdır. Bunun temel nedenlerinden birisi Türkçe dilinde özetlemeye uygun yeterince veri setinin bulunmamasıdır. Dolayısıyla yapılacak bir özetleme sisteminin Türkçe dili üzerindeki başarısı iyi bir şekilde ölçülmeyebilir. Bu nedenle tezde ikinci bir Türkçe veri seti olarak Haber.com sitesinden elde edilen 4 farklı kategoriye ait 400 haber toplanmıştır. Her haberin özeti olarak haberin açıklama kısmında yer alan kısa paragraf kullanılmıştır.

3.3. Kıyaslama Ölçütleri

Otomatik özetleme sistemlerinde üzerinde durulması gereken önemli aşamalardan bir diğeri özetin değerlendirilmesi işlemidir. Otomatik özetleme sistemi tarafından oluşturulan bir özet ne kadar iyi, kaliteli ve orijinal özete yakın olursa özetleme sistemi de o ölçüde iyi bir sistem olarak kabul edilir. Bu nedenle sistemin oluşturduğu özetin başarısı belirli standartlar doğrultusunda ve iyi bir şekilde belirlenmelidir. İnsanlar tarafından bir özetin başarısı uyumluluk, özü yansıtma, dilbilgisel, okunabilirlik, içerik gibi pek farklı şekilde değerlendirilebilir (Mani 2001). Ancak büyük ölçekli çalışmalarda, insanların tüm özetleri okuyup orijinal olanla karşılaştırarak kalitesini belirlemesi oldukça zor ve maliyetli bir süreçtir. Bu durum sistem özetini, orijinal özet (gold özet) ile hızlı ve doğru bir şekilde karşılaştırarak değerlendirebilecek ölçütlerin ihtiyacını ortaya çıkarmıştır. Gold özet, konusunda uzman kişiler tarafından ilgili dokümanı en iyi temsil edecek şekilde hazırlanan özettir.

$$Hassasiyet = \frac{|S \cap T|}{|S|} \tag{43}$$

$$Keskinlik = \frac{|S \cap T|}{|T|} \tag{44}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recal} \tag{45}$$

Bu amacı gerçekleştirmek için ortaya çıkan klasik yaklaşımda sistem ve orijinal özette geçen kelime veya cümlelerin örtüşmesine dikkat edilerek değerlendirilir. S sistem tarafından oluşturulan, T ise orijinal özette geçen kelime veya cümle olmak üzere iki özet arasındaki hassasiyet, keskinlik ve f değerleri eşitlik 43,44 ve 45'teki gibi hesaplanmaktadır. Hassasiyet doğru sonuçların olması gereken doğru sonuçlara, keskinlik doğru sonuçların tüm sonuçlara, f skoru ise bu ikilinin harmonik ortalamasıyla ilgi bilgi veren ölçütlerdir. Bu değerler genellikle 0 ile 1 arasındadır. 0 değeri özetler arasındaki benzerliğin hiç olmadığını, 1 ise özetlerin birebir aynı olduğunu göstermektedir.

Çalışmalarda Saggion vd. (2002), kosinüs benzerliği, unigram-bigram örtüşmesi ve en uzun ortak alt dizi yöntemleri; n-gram birlikte geçme istatistikleri (Lin, Hovy 2003) gibi farklı ölçütleri özet değerlendirme sürecinde başarıyla kullanılmıştır. Ancak bu konuda herhangi bir standart o döneme kadar getirilememiştir. Mesaj

Anlamlandırma Konferanslarında (DUC) bu konu ele alınmış ve standart ölçüt kullanmak amacıyla 2004 yılından itibaren Rouge metrikleri kullanılmaya başlanmıştır.

Rouge (Recall-Oriented Understudy for Gisting Evaluation) ölçütleri Lin (2004) tarafından 2004 yılında ortaya koyulan ve aday özet (sistem tarafından oluşturulan özet) kalitesini insanlar tarafından oluşturulan referans özet (orijinal veya ideal özet) ile karşılaştırarak değerlendiren bir pakettir. Rouge kapsamında bir özeti değerlendirebilecek 3 ana ölçüt ortaya atılmıştır.

3.3.1. Rouge-N

Rouge-N, referans özet ile aday özet arasındaki hassasiyet değerini n-gram örtüşmesine göre puanlayan bir yöntemdir. N-gram bir kelime dizisindeki n sayısına göre bölünen parçaların tekrar oranını bulmaya çalışır. Örneğin, "*metin için otomatik doküman özetleme*" cümlesini kelime bazlı 3-gram'a göre bölmek istediğimizde $C = \{(metin\ için\ otomatik),\ (için\ otomatik\ doküman),$
$(otomatik\ doküman\ özetleme)\}$ olacak şekilde 3 elemanlı bir küme elde ederiz. Rouge-N'de iki özeti bu şekilde bölerek ortak eleman sayılarına göre puanlama yapmaktadır.

$$Rouge - N\ = \frac{\sum_{S \in \{ReferansCümleleri\}} \sum_{gram_n} Toplam_{eşleşen}(gram_n)}{\sum_{S \in \{ReferansCümleleri\}} \sum_{gram_n \in S} Toplam(gram_n)} \qquad (46)$$

S referans özetteki bir cümleyi, n ise $gram$ sayısını belirtmek üzere Rouge-N eşitlik 46'daki gibi hesaplanmaktadır. Eşitliğin pay kısmında iki cümle arasındaki eşleşen $n-gram$ sayısı yer alırken payda kısmında referans özetteki toplam $n-gram$ sayısı yer almaktadır. Bu ölçütü bir örnek üstünde göstermek için 2 cümlemiz bulunsun:

ÖRNEK 2:

- **Aday cümle:** *"Hurricane Gilbert swept toward the Dominican Republic on Sunday",*
- **Referans cümle:** *"Hurricane Gilbert is moving toward the Dominican Republic".*

Bu cümlelere 2-grama göre ayırırsak

- **Aday cümle:** *{ Hurricane Gilbert, Gilbert swept, swept toward, toward the, the Dominican, Dominican Republic, Republic on, on Sunday },*

- **Referans cümle:** *{ Hurricane Gilbert, Gilbert is, is moving, moving toward, toward the, the Dominican, Dominican Republic }*

olacak şekilde kelime ikilileri oluşur. Ortak geçen ikililer altı çizili şekilde belirtilmiştir. Bunlar kullanılarak Rouge-2 hesaplandığı zaman:

$$Rouge - N = \frac{4}{7} = 0{,}571 \tag{47}$$

benzerlik skoru elde edilmiş olur.

3.3.2. Rouge-S

Rouge-S, Rouge-N'e çok benzer şekilde çalışan ancak N-gram yerine Skip-Bigram örtüşmesiyle adayın başarısını değerlendiren ölçüttür. N-gram yöntemde belirlenen N sayısına göre pencere oluşturulur ve hiçbir kelime atlanmadan pencere kaydırılarak cümle parçalanır. Skip-Bigram modelde ise $N = 2$ olarak sabitlenir ve cümle kelime sayısı m ile temsil edilmek üzere $Kombinasyon(m, 2)$ sayısı kadar ikili kelimeye ayrıştırılır. Örnek 2'deki cümlelere Skip-Bigram uygulandığında:

- **Aday cümle:** *{ "Hurricane Gilbert", "Hurricane swept", "Hurricane toward", "Hurricane the", "Hurricane Dominican", "Hurricane Republic", "Hurricane on", "Hurricane Sunday", "Gilbert swept", "Gilbert toward", "Gilbert the", "Gilbert Dominican", "Gilbert Republic", "Gilbert on", "Gilbert Sunday", "swept toward", "swept the", "swept Dominican", "swept Republic", "swept on", "swept Sunday", "toward the", "toward Dominican", "toward Republic", "toward on", "toward Sunday", "the Dominican", "the Republic", "the on", "the Sunday", "Dominican Republic", "Dominican on", "Dominican Sunday", "Republic on", "Republic Sunday", "on Sunday"},*
- **Referans cümle:** *{ "Hurricane Gilbert", "Hurricane is", "Hurricane moving", "Hurricane toward", "Hurricane the", "Hurricane Dominican", "Hurricane Republic", "Gilbert is", "Gilbert moving", "Gilbert toward", "Gilbert the", "Gilbert Dominican", "Gilbert Republic", "is moving", "is toward", "is the", "is Dominican", "is Republic", "moving toward", "moving the", "moving Dominican", "moving Republic", "toward the", "toward Dominican", "toward Republic", "the Dominican", "the Republic", "Dominican Republic" }*

elde ettiğimi Skip-Bigram kelimeler bu şekildedir. Aday cümle toplam 36, referans cümle ise toplam 28 ikiliden oluşmuştur. İki cümle arasındaki Skip-BiGram eşleşme sayısı ise 15'tir.

$$\text{Hassasiyet}_{\text{Skip2}} = \frac{\text{Skip}-\text{BiGram(X,Y)}}{\text{Kombinasyon(m,2)}} \quad (48)$$

$$\text{Keskinlik}_{\text{Skip2}} = \frac{\text{Skip}-\text{BiGra (X,Y)}}{\text{Kombinasyon(n,2)}} \quad (49)$$

$$\text{F}_{\text{Skip2}} = \frac{(1+ß^2)*\text{H}_{\text{Skip2}}*\text{K}_{\text{Skip2}}}{\text{H}_{\text{Skip2}}+ß^2*\text{K}_{\text{Skip2}}} \quad (50)$$

Eşleşme sayısı, ilgili cümlelerde bulunan tüm Skip-BiGram sayısına bölünerek sırasıyla hassasiyet, keskinlik skorları hesaplanır. Skorlar belirli bir fonksiyona tabi tutulur ve iki cümle arasındaki nihai Rouge-S skorunu belirten F-skoru elde edilir.

3.3.3. Rouge-L

Rouge-L, bölüm 2.2.5.3.1.3'te yer verilen En uzun ortak alt küme (LCS) yönteminin özet ölçümüne uygulanmış halidir. Bu yöntem ile iki farklı özet arasındaki en uzun olan ortak kelime altdizisi kullanılmaktadır. Burada şart ortak kelime altdizisinin iki cümlede de aynı sırada olmasıdır. Rouge-L cümle seviyesinde ve özet seviyesinde olmak üzere 2 tiptedir. Cümle seviyesindeki Rouge-L'de aday ve referans özetlerin her birisi tek bir cümle gibi düşünülerek ikili arasındaki en uzun alt küme bulunur. Örnek 2'ye bu yöntemi uyguladığımızda 6 kelimeden oluşan bir dizi elde edilir.

$$\text{Hassasiyet}_{\text{lcs}} = \frac{\text{LCS(X,Y)}}{\text{m}} \quad (51)$$

$$\text{Keskinlik}_{\text{lcs}} = \frac{\text{LCS(X,Y)}}{\text{n}} \quad (52)$$

$$\text{F}_{\text{lcs}} = \frac{(1+ß^2)*\text{H}_{\text{lcs}}*\text{K}_{\text{lcs}}}{\text{H}_{\text{lcs}}+ß^2*\text{K}_{\text{lcs}}} \quad (53)$$

Aday özet X, referans özet Y, aday özet uzunluğu m ve referans özet uzunluğu n ile temsil edilmek üzere bir dokümanın Rouge-L için hassasiyet, keskinlik ve f değerleri eşitlik 51,52 ve 53'teki gibi hesaplanmaktadır. $LCS(X,Y)$ değerleri iki dokümanın kelime bazında en uzun ortak altdizindeki küme sayısını belirtmektedir. ß ise herhangi bir katsayı değeridir. Bu katsayı eğer $F/A = F/K$ ise K/A oranına eşittir. Ancak bazı çalışmalarda ß değeri 1 olarakta alınmıştır. Örnek 2 üzerinde Rouge-L değerlerini hesaplarsak:

- *LCS(Aday, Referans)={ Hurricane, Gilbert, toward, the, Dominican, Republic}*

$$\text{Hassasiyet}_{\text{lcs}} = \frac{\text{LCS(X,Y)}}{\text{m}} = \frac{6}{9} = 0{,}667 \quad (54)$$

$$\text{Keskinlik}_{\text{lcs}} = \frac{\text{LCS(X,Y)}}{\text{n}} = \frac{6}{8} = 0{,}75 \quad (55)$$

$$\text{F}_{\text{lcs}} = \frac{(1+ß^2)*\text{H}_{\text{lcs}}*\text{K}_{\text{lcs}}}{\text{H}_{\text{lcs}}+ß^2*\text{K}_{\text{lcs}}} = \frac{(1+1^2)*0{,}667*0{,}75}{0{,}667+\ (1^2*0{,}75)} = \frac{1}{1{,}417} = 0{,}706 \quad (56)$$

skorları elde edilir. F_{lcs} skoruna göre yorum yapmak gerekirse aday cümle %70,6 orijinal cümleyi temsil etme kapasitesine sahiptir.

Diğer tip olan özet seviyesinde Rouge-L ise referans özetteki her bir cümle ile aday özetteki cümlelerin ayrı ayrı benzerliğine bakılır ve bu benzerlikler toplanarak birleştirilir. Cümle seviyesindeki Rouge-L'den farklı olarak formüller,

$$\text{Hassasiyet}_{\text{lcs}} = \frac{\sum_{i=1}^{u} \text{LCS}_{\cup}\ (\text{r}_i\ ,\text{C})}{\text{m}} \quad (57)$$

$$\text{Keskinlik}_{\text{lcs}} = \frac{\sum_{i=1}^{u} \text{LCS}_{\cup}\ (\text{r}_i\ ,\text{C})}{\text{n}} \quad (58)$$

$$F_{lcs} = \frac{(1+ß^2)*H_{lcs}*K_{lcs}}{H_{lcs}+ß^2*K_{lcs}} \quad (59)$$

ile değiştirilir. Bu formüllerde geçen $r_i \in u$ referans özet cümlelerini, $c_i \in C$ ise aday özetin cümlelerini temsil etmektedir.

3.4. Yöntemlerin Uygulanması ve Elde Edilen Sonuçlar

Tezin uygulama kısmında literatürde en sık kullanılan özetleme yaklaşımları farklı varyasyonlar ile modellenerek veri setlerine uygulanmıştır. Klasik yöntem dışında sezgisel algoritma, çizge tabanlı algoritmalar, gizli anlam analizi, kümeleme ve yapay sinir ağlarından oluşturulan 9 farklı özetleyici sistem oluşturulmuştur.

Tablo 4. Veri setlerine İlişkin Genel Bilgiler

Veri seti	Doküman Sayısı	Cümle Sayısı	Dil
Türkçe Haber veri seti	120	2544	Türkçe
Habercom veri seti	400	5143	Türkçe
DUC 2002	557	16675	İngilizce
MultiLing	300	47200	Karışık

Genel bilgileri tablo 4'te paylaşılan 4 farklı veri seti normal (herhangi bir işlem yapılmamış), gereksiz kelimeler temizlenmiş ve kelime kökleri alınmış olarak 3 farklı şekilde tekrar derlenmiştir. Kullanılan her veri seti modeli belirli bir standarta

uydurularak algoritmaların veri setlerine göre kodlanmasının önüne geçilmiştir. Standart olarak her dokümanın ilk satırı başlık, diğer bir satıra bir cümle olacak şekilde; özet ise her bir satıra bir cümle olacak şekilde metin dosyaları halinde saklanmıştır. DUC ve MultiLing, kaynakları tarafından XML formatında halka açık olarak paylaşılmış veri setleridir. XML formatındaki veri için Python ElementTree XML API (*ElementTree*, 2017) paketi kullanılarak dokümanın yapısı standarta uydurulmuştur.

Dokümanlara uygulanan ön işlemlerin ilki olan gereksiz kelimeleri temizlemek maksadıyla Türkçe için lokalde oluşturulan bir liste; diğer diller için ise "*many-stop-words"* kütüphanesi (Inc 2017) kullanılmıştır. Özellik tabanlı yaklaşımlarda bazı kelime türleri öznitelik olarak kabul etmektedir. Bu sebeple kelime türlerini çıkartmak için RippleTagger (Stenström 2016) kütüphanesinden faydalanılmıştır. Kelime köklerinin bulunması içinse Türkçe veri setleri için İTÜ DDİ aracı kullanılmıştır (Eryigit 2014). Araç içerisinde yer alan "morfolojik çözümleyici" modülü, verilen kelimenin kök, gövde hallerini ve aldığı ekleri saptamaktadır. Bu modül vasıtasıyla veri setindeki tüm dokümanlar kök haline çevrilmiştir. İngilizce dili için sözlük tabanlı pyHunspell (Latinier 2017), diğer diller için Snowball Stemmer (Shibukawa 2015) kütüphanelerinin kök bulucu modülleri ile kelimeler köklerine ayrılmıştır. Ön işlemlerin son aşamasında, farklı dillere uygulanabilir paketleri bulunmasından dolayı NLTK Punkt Tokenizer (Bird 2017) kütüphanesiyle dokümanlar cümlelere bölünmüştür. Hangi veri setine hangi öntemizlik kütüphanesinin kullanıldığı tablo 5'te kısa şekilde gösterilmektedir.

Tablo 5. Önişlem için Veri Setlerinde Kullanılan Python Kütüphaneleri

Veri seti	Doküman Bölücü	Kök Çıkarıcı	Gereksiz Kelimeler	Kelime Etiketleyici
Türkçe Haber veri seti	NLTK Punkt Küt.	ITU NLP Tool	Lokal Veri	RippleTagger Küt.
Habercom veri seti	NLTK Punkt Küt.	ITU NLP Tool	Lokal Veri	RippleTagger Küt.
DUC 2002	NLTK Punkt Küt.	pyHunspell Küt.	many-stop-words Küt.	RippleTagger Küt.
MultiLing	NLTK Punkt Küt.	Snowball Küt.	many-stop-words Küt.	RippleTagger Küt.

Oluşturulacak özet uzunluğunu belirten sıkıştırma oranı cümleler baz alınarak %20 olarak seçilmiştir. Yani eğer bir haber 5 cümleden oluşuyorsa sistem tarafından oluşturulan özeti 1 cümleden oluşmaktadır. Ayrıca 5 cümleden az oluşan dokümanlar için özet uzunluğu 1 seçilerek özetin boş olması engellenmiştir. Orana göre ondalıklı olan sayılar yuvarlanarak tam sayıda cümle seçilmesi sağlanmıştır. Multiling veri seti

içerisindeki bazı dokümanların uzunluğu çok fazla olması sebebiyle üst limit olarak özet 15 cümle ile sınırlandırılmıştır. Diğer veri setlerine herhangi bir üst limit kısıtlaması uygulanmamıştır.

Klasik yöntem, cümlelerin bazı yapısal ve içeriksel öznitelik değerlerine göre sıralandığı yaklaşımdır. Tez çalışmasında klasik yöntem için cümlelere ait 8 farklı öznitelik kullanılmıştır. Bunlar sırasıyla uzunluk, konum, TF/ISF, başlık, özel isim, tematik kelime, numerik veri, benzerlik ile ilgili olan özniteliklerdir. Uzunluk özniteliği için cümleden geçen harf sayısı değil, cümlenin kaç kelimeden oluştuğu bilgisi kullanılmıştır. Tematik kelime olarak gereksiz kelimeler hariç dokümanda en çok geçen 5 kelime seçilmiştir. Özel isim içinse etiketleyici kullanılarak kelimelerin özel isim olup olmadığı belirlenmiştir. Cümlelerin skorlarını hesaplanırken öznitelikleri toplamı esas alınmıştır.

GA uygulamasında amaç özette yer alacak cümlelerin skorlarını belirlerken kullanılan özniteliklere ait ağırlık değerlerinin optimize edilmesidir. Klasik yöntemde her bir özniteliğin ağırlığı 1 kabul edilerek skor hesaplanır. Ancak bazı durumlarda özniteliklerin ağırlıkları değişebilir. Ağırlıkların elle tek tek denenmesi yerine bir optimizasyon algoritmasıyla en uygun değerlerin bulunması hem daha mantıklı hem de daha hızlı bir çözümdür. Bu nedenle tez kapsamında her bir öznitelik için ilgili ağırlık değerini temsil eden en uygun w_i^* değerleri GA ile saptanmıştır. GA algoritmasının eğitimi için çeşitli deneyler sonucunda popülasyon sayısı 80, jenerasyon sayısı 200, çaprazlama oranı 0.9 ve mutasyon oranı 0.1 olacak şekilde en uygun parametreler belirlenmiştir. Algoritmanın amaç fonksiyonu, özet yardımıyla elde edilen Rouge-1 skorunun maksimize edilmesidir. Her olası çözümden gelen ağırlıklar yardıyla cümle skorları hesaplanarak özet oluşturulmuş ve ilgili özetin Rouge-1 skoru hesaplanmıştır. Her jenarasyonda en iyi özeti veren çözüm yani öznitelik ağırlıkları bir sonraki jenerasyona aktarılarak en başarılının bulunması hedeflenmiştir.

YSA yöntemi, ilgili cümlenin özette geçip geçmemesine bağlı olarak eğitilen bir sınıflayıcı şeklinde çalıştığından sadece 120 Türkçe haber veri setine uygulanmıştır. Bunun sebebi bu veri setinde yer alan özetlerin direk olarak orijinal dokümanlardaki cümlelerden seçilmesidir. Diğer veri setlerinin özetleri birebir orijinal

dokümanda geçmediğinden dolayı etiketleme sırasında oluşabilecek belirsiz durumlar nedeniyle YSA bu veri setlerine uygulanmamıştır. YSA modeli için 8 önziteliği temsilen 8 giriş, özette geçip geçmediği için ise 2 çıkıştan oluşan çok katmanlı algılayıcı tercih edilmiştir. Modelin gizli katman sayısı farklı tipteki (normal, gereksizler temizlenmiş, kök alınmış) veri seti için 5 ile 500 arasında optimize edilerek normal için 55, gereksiz kelimelerden temizlenmişte 35, kök alınmışta 5 olarak belirlenmiştir.

Eğitim gerektiren algoritmalar (YSA, GA) için veri setleri %75'i eğitim, %25'i test olacak şekilde bölünmüştür. Her dokümanın özet uzunluğu, özet cümleleri, cümlelerin sırası gibi eğitim/test başarısını etkileyecek bilgiler farklı olduğu için veri setlerinde çapraz geçerlilik uygulanmasına ihtiyaç duyulmamıştır.

Kümeleme yaklaşımı için hem hızlı hemde yaygın kullanımı nedeniyle K-Means algoritması kullanılmıştır. Bu algoritma önceden belirlenen k küme sayısını parametre olarak almaktadır. Sistemimizde bu sayı, oluşturulacak özetin uzunluğuna eşit olarak seçilmiştir. Örneğin %20 sıkıştırma oranı ile 10 cümleden oluşan haberin özeti 2 cümleden oluşacaktır. Dolayısıyla k küme sayısı da 2 olarak seçilecektir. Sistemin iterasyon sayısı 300 olarak belirlenmiş; cümlelerin merkeze uzaklığı öklid ile hesaplanmıştır. Her kümeden merkeze en yakın cümleler seçilerek ilgili dokümana ait özet oluşturulmuştur.

Gizli anlam analizi için Steinberger ve Jezek'in önerdiği geliştirilmiş GAA yaklaşımı tercih edilmiştir. Bunun sebebi yöntemin sadece birkaç tekil değeri değil, ilgili cümledeki tüm tekil değerleri hesaba katarak daha doğru bir skor üreteceği fikridir. Cümlelerin vektörlere çevirirken öznitelikler yerine kelimelerin geçme sıklığı bilgisi kullanılmıştır. Vektörler birleştirilerek terim-cümle matrisi elde edilmiş ve sparsesvd (Rehurek 2013) kütüphanesi kullanılarak bu matrise tekil değer ayrışımı uygulanmıştır.

Çizge tabanlı yaklaşıma ait LexRank, TexRank, Jaccard uygulanmış TextRank, LCS uygulanmış TextRank olmak üzere 4 farklı sıralama algoritması kullanılmıştır. LexRank algoritması için Sumy (Belica 2013) kütüphanesinden yararlanılmıştır. Cümlelerin kosinüs benzerliği için eşik 0.1 olarak belirlenmiştir. LCS TextRank algoritmasında, cümleler arası LCS skoru cümlelerin uzunluklarına bölünerek

normalize edilmiştir. Diğer TetRank algoritmalarında herhangi bir ekstra düzeltme yapılmamıştır. Diğer sıralama algoritmalarında herhangi bir değişiklik yapılmadan kullanılmıştır.

Tüm işlemler core i7 4 çekirdek / 8 izlek işlemcili, 16 gb RAM'li bir bilgisayarda, Python dilinde kodlanarak Anaconda Spyder platformu üstünde çalıştırılmıştır.

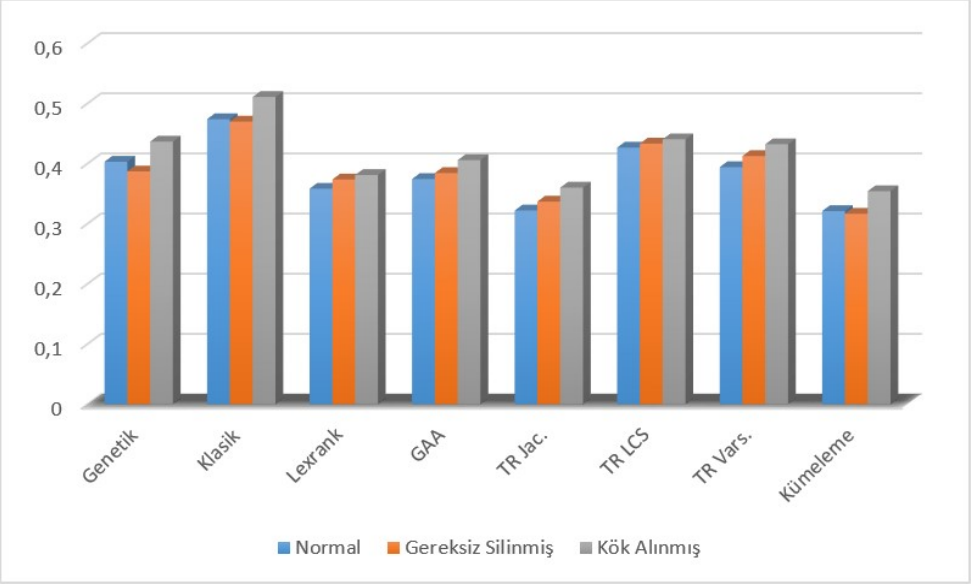
Her veri seti ve algoritmaya ait sonuçlar tablolar, ortalama başarı skorları ise şekiller halinde olacak şekilde eklenmiş ve yorumlanmıştır. Ortalama başarı, ilgili algoritma ve veri setinin başarısına Rouge-1, Rouge-2, Rouge-S ve Rouge-L F skorlarının ortalaması alınarak hesaplanmıştır.

3.4.1. 120 Türkçe Haber Veri Setinden Elde Edilen Sonuçlar

120 Türkçe haber veri setinden elde edilen tüm istatistiksel sonuçlar tablo 7'de paylaşılmıştır. Skorlara bakıldığı zaman, klasik yöntem tüm ölçütlerde en iyi sonuçları elde etmiştir. Genetik algoritma Rouge-1; LCS TextRank Rouge-2, Rouge-L ve Rouge-S ölçütlerinde diğer en iyi yöntemlerdir.

Tablo 6. 120 Türkçe Haber Veri Setinden Elde Edilen Rouge Değerleri

YÖNTEM	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Anma	Rouge-L Keskinlik	Rouge-L F skor
Klasik	*Normal*	0,5768	0,5084	0,3181	0,5278	0,7257	0,4921
	Gereksizler Silinmiş	0,5713	0,5107	0,3116	0,5287	0,7143	0,4858
	Kök Alınmış	0,6271	0,5373	0,3699	0,5534	0,7574	0,5089
Genetik	*Normal*	0,5139	0,4529	0,2245	0,4713	0,7877	0,4211
	Gereksizler Silinmiş	0,4902	0,4396	0,2026	0,4643	0,7585	0,4153
	Kök Alınmış	0,5878	0,4992	0,2049	0,5182	0,7645	0,4545
Küme.	*Normal*	0,4365	0,3486	0,1556	0,3552	0,6138	0,3440
	Gereksizler Silinmiş	0,4197	0,3458	0,1540	0,3556	0,6532	0,3467
	Kök Alınmış	0,4953	0,3853	0,1615	0,3873	0,6740	0,3744
GAA	*Normal*	0,5043	0,4138	0,1853	0,4369	0,5290	0,3942
	Gereksizler Silinmiş	0,5057	0,4274	0,1976	0,4495	0,5457	0,4069
	Kök Alınmış	0,5758	0,4439	0,1810	0,4684	0,5654	0,4226
Lexrank	*Normal*	0,4787	0,3899	0,1742	0,4155	0,6063	0,3899
	Gereksizler Silinmiş	0,4881	0,4119	0,1836	0,4358	0,6484	0,4107
	Kök Alınmış	0,5328	0,4186	0,1633	0,4446	0,6478	0,4097
Jaccard TR	*Normal*	0,4109	0,3495	0,1622	0,3751	0,6519	0,3654
	Gereksizler Silinmiş	0,4199	0,3713	0,1691	0,3988	0,6929	0,3868
	Kök Alınmış	0,4801	0,3951	0,1618	0,4192	0,7091	0,4031
LCS TR	*Normal*	0,5231	0,4702	0,2519	0,4699	0,7929	0,4613
	Gereksizler Silinmiş	0,5288	0,4800	0,2567	0,4789	0,7934	0,4678
	Kök Alınmış	0,5696	0,4898	0,2298	0,4845	0,8005	0,4726
TR	*Normal*	0,5088	0,4387	0,2034	0,4610	0,6145	0,4255
	Gereksizler Silinmiş	0,5211	0,4617	0,2189	0,4834	0,6408	0,4486
	Kök Alınmış	0,5858	0,4825	0,1996	0,5054	0,6633	0,4608



Şekil 37. 120 Türkçe Haber için Algoritmaların Ortalama Başarı Değerleri

Aynı veri setine ait şekil 37'deki ortalama sonuçlara bakıldığı zaman en düşük skorların kümeleme ve Jaccard uygulanmış TextRank algoritmalarının aldığı gözükmektedir. Bununla beraber klasik yöntem, genetik algoritma ve kümeleme yöntemlerinde gereksiz kelime temizliği başarıyı düşürürken diğer yöntemlerde arttırmıştır.

Aynı veri setinin YSA yöntemi ile elde edilen karışıklık matrisi skorları tablo 8'dedir. İşlem görmemiş dokümanlar içerisinde 389 cümlenin özette geçtiği, 38 cümlenin ise özette geçmediği doğru olarak model tarafından tahmin edilebilmiştir. Buna karşılık 46 cümle özette geçmesine rağmen, YSA bunları özette olmayan cümle olarak işaretlemiştir.

Tablo 7. 120 Türkçe Haber YSA Karışıklık Matrisi

Karışıklık Matrisi	Normal		Gereksizler Silinmiş		Kök Alınmış	
	Özette	***Özette Değil***	***Özette***	***Özette Değil***	***Özette***	***Özette Değil***
Özette	389	160	344	143	328	140
Özette Değil	46	38	63	87	61	108

Tablodan elde edilebilecek başka bir sonuçta önişlem temizlik seviyesi arttıkça özette geçen cümle tahmin oranının düşmesi, özette geçmeyen cümle tahminin oranının ise artmasıdır.

Tablo 8. YSA ile Elde Edilen Doğruluk, Anma, Keskinlik ve F Skor Değerleri

	Normal	Gereksizler Silinmiş	Kök Alınmış
Doğruluk *(Accuracy)*	0,6746	0,6766	0,6845
Hassasiyet *(Recall)*	0,8943	0,8452	0,8432
Keskinlik *(Precision)*	0,7086	0,7064	0,7064
Doğru Negatif Oranı	0,1919	0,3783	0,4355
F-Skor	0,7907	0,7696	0,7687

YSA yönteminin karışıklık matrisindeki sonuçlara göre elde ettiği başarı sonuçları tablo 9'de görüldüğü gibidir. Doğruluk oranı doğru tahminlerin tüm tahminlere oranını gösterir. F skoru ise hassasiyet ve keskinliğin ortalamasıyla ilgili istatistiksel bilgi verir (Powers 2011). Bu değerler doğrultusunda özetleyici sistemin özette geçen/geçmeyen cümleleri tahmin etme başarısı normal veriler için %67,5 gereksiz kelimelerden temizlenenler için %67,7 ve kök alınanlar için %68,5 olduğu görülmektedir.

3.4.2. DUC Veri Setinden Elde Edilen Sonuçlar

İngilizce dili için kullanılan DUC 2002 veri setinden elde edilen Rouge skorları tablo 10 görüldüğü gibidir. Genetik algoritma Rouge-1,Rouge-2 ve Rouge-L ölçütleri açısından DUC 2002 veri seti için en iyi sonuçları elde etmiştir. Bu algoritmayı sırayla klasik yöntem ve GAA takip etmektedir. Rouge-S metriğinde ise klasik yöntem diğerlerinden belirgin bir farkla daha başarılıdır.

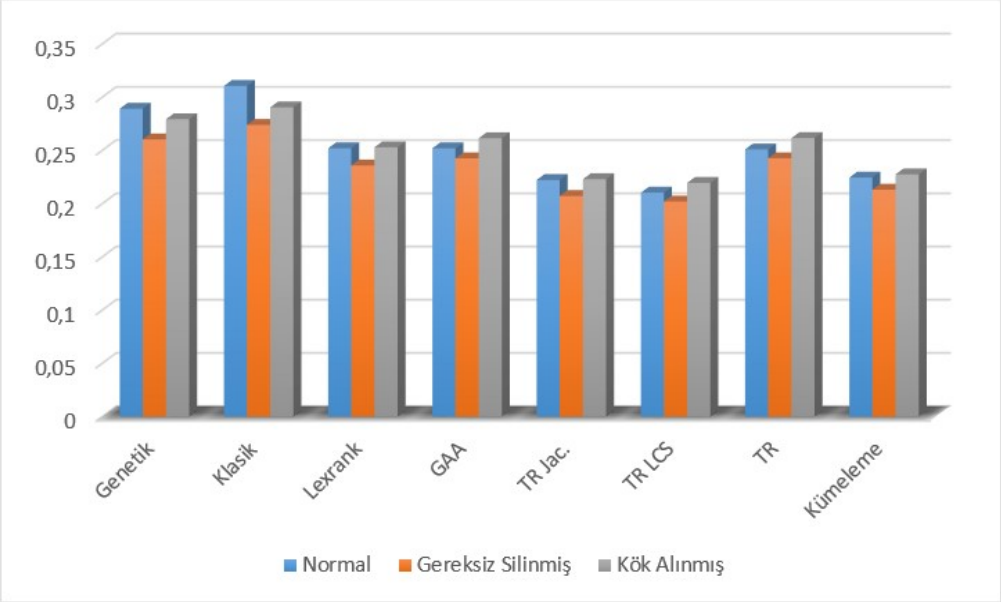
Tablo 9. DUC Veri Setinden Elde Edilen Rouge Değerleri

YÖNTEM	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Anma	Rouge-L Keskinlik	Rouge-L F skor
Klasik	***Normal***	0,5203	0,2644	0,2033	0,3662	0,2614	0,2543
	Gereksizler Silinmiş	0,4626	0,2315	0,1454	0,3544	0,2656	0,2562
	Kök Alınmış	0,4921	0,2419	0,1623	0,3687	0,2745	0,2655
Genetik	***Normal***	0,5269	0,2644	0,1004	0,3637	0,2742	0,2650
	Gereksizler Silinmiş	0,4654	0,2216	0,1009	0,3488	0,2674	0,2535
	Kök Alınmış	0,5050	0,2389	0,1091	0,3668	0,2770	0,2641
Kümeleme	***Normal***	0,4207	0,1773	0,0753	0,2780	0,2498	0,2243
	Gereksizler Silinmiş	0,3725	0,1611	0,0852	0,2721	0,2684	0,2336
	Kök Alınmış	0,4031	0,1708	0,0921	0,2867	0,2802	0,2438
GAA	***Normal***	0,4907	0,2266	0,0741	0,3341	0,2129	0,2164
	Gereksizler Silinmiş	0,4569	0,2132	0,0805	0,3376	0,2142	0,2194
	Kök Alınmış	0,5005	0,2263	0,0883	0,3566	0,2252	0,2307
Lexrank	***Normal***	0,4769	0,2194	0,0789	0,3270	0,2407	0,2326
	Gereksizler Silinmiş	0,4211	0,1942	0,0856	0,3208	0,2555	0,2422
	Kök Alınmış	0,4567	0,2088	0,0939	0,3380	0,2645	0,2521

Tablo 9. (devamı) DUC Veri Setinden Elde Edilen Rouge Değerleri

TextRank Jaccard	*Normal*	0,4052	0,1803	0,0767	0,2818	0,2506	0,2264
	Gereksizler Silinmiş	0,3521	0,1630	0,0795	0,2770	0,2668	0,2340
	Kök Alınmış	0,3834	0,1743	0,0868	0,2959	0,2834	0,2475
TextRank LCS	*Normal*	0,3784	0,1651	0,0728	0,2664	0,2545	0,2248
	Gereksizler Silinmiş	0,3431	0,1570	0,0777	0,2692	0,2677	0,2298
	Kök Alınmış	0,3769	0,1695	0,0862	0,2883	0,2861	0,2453
TextRank Varsayılan	*Normal*	0,4757	0,2226	0,0792	0,3295	0,2305	0,2263
	Gereksizler Silinmiş	0,4298	0,2090	0,0893	0,3329	0,2476	0,2424
	Kök Alınmış	0,4697	0,2228	0,0986	0,3525	0,2596	0,2551

Ancak şekil 38'teki ortalama başarılara bakıldığı zaman klasik yöntemin biraz daha başarılı olduğu görülür. Buna genetik algoritma yönteminin Rouge-S metriğinden çok düşük skorlar alması sebep gösterilebilir. Rouge-S metriği cümle içerisindeki geçen tüm ikili kelime kombinasyonlarını karşılaştırmasına rağmen yöntemlerin elde ettiği skorlar düşük çıkmıştır. Bu bulgu özette geçmesine rağmen orijinal doküman içerisinde geçmeyen kelimelerin çok olduğuna bir gösterge olabilir.



Şekil 38. DUC Veri Seti için Algoritmaların Ortalama Başarı Sonuçları

En başarısız yöntemler ise Jaccard uygulanmış TextRank, LCS Uygulanmış TextRank ve kümelemedir. Dikkat çekici diğer bir nokta ise İngilizce dilinde gereksiz kelimelerin başarıyı arttırmak yerine ciddi bir şekilde düşürmesidir. Kelimelerin köklerinin alınması ise genel anlamda başarıyı arttırmıştır.

3.4.3. Habercom Veri Setinden Elde Edilen Sonuçlar

Türkçe dilinin başarısını ölçmek için kullanılan diğer veri seti 4 farklı kategoriden oluşan Haberco verisetidir. Özetleyici sistemlerin her bir kategoride elde ettiği ortalama Rouge ölçütler sonuçları tablolarda verilmiştir.

Tablo 10. Habercom Veri Setinden Elde Edilen Klasik Yöntem Rouge Değerleri

TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hasssasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	*Normal*	0,2983	0,0973	0,0433	0,2423	0,1369	0,1285
	Gereksizler Silinmiş	0,2940	0,1036	0,0455	0,2545	0,1454	0,1356
	Kök Alınmış	0,4373	0,1492	0,0820	0,3398	0,1819	0,1788
Magazin	*Normal*	0,2316	0,0803	0,0346	0,1787	0,1638	0,1371
	Gereksizler Silinmiş	0,2435	0,0941	0,0433	0,2010	0,1932	0,1587
	Kök Alınmış	0,3406	0,1207	0,0773	0,2490	0,2288	0,1942
Siyaset	*Normal*	0,3714	0,1530	0,0319	0,2996	0,1174	0,1127
	Gereksizler Silinmiş	0,3801	0,1694	0,0368	0,3303	0,1329	0,1297
	Kök Alınmış	0,5066	0,2128	0,0555	0,4036	0,1532	0,1580
Spor	*Normal*	0,3107	0,1275	0,0499	0,2499	0,1691	0,1502
	Gereksizler Silinmiş	0,3434	0,1458	0,0644	0,2932	0,1984	0,1795
	Kök Alınmış	0,4489	0,1904	0,0985	0,3597	0,2356	0,2174

Klasik yönteme ait sonuçlar tablo 10'daki gibidir. Klasik yöntemin ortalama skorlarında Rouge-1 ve Rouge-2 için en yüksek başarıyı siyaset kategorisindeki kök alınmış dokümanlar elde etmiştir. Rouge-L için ise spor kategorisinin kök alınmış dokümanlar en yüksek skoru elde etmiştir.

Tablo 11. Habercom Veri Setinden Elde Edilen Genetik Algoritma Rouge Değerleri

TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	*Normal*	0,2823	0,0714	0,0179	0,2302	0,0930	0,1014
	Gereksizler Silinmiş	0,2743	0,1007	0,0226	0,2430	0,1041	0,1142
	Kök Alınmış	0,4576	0,1613	0,0526	0,3548	0,1617	0,1759
Magazin	*Normal*	0,2479	0,0707	0,0282	0,1936	0,1460	0,1359
	Gereksizler Silinmiş	0,2541	0,0901	0,0367	0,2081	0,1835	0,1603
	Kök Alınmış	0,3365	0,1098	0,0563	0,2463	0,2128	0,1866
Siyaset	*Normal*	0,3901	0,1650	0,0291	0,3163	0,1096	0,1159
	Gereksizler Silinmiş	0,3928	0,1834	0,0392	0,3478	0,1381	0,1452
	Kök Alınmış	0,5351	0,2562	0,0532	0,4404	0,1601	0,1711
Spor	*Normal*	0,2006	0,0552	0,0300	0,1485	0,1661	0,1226
	Gereksizler Silinmiş	0,2195	0,0663	0,0404	0,1828	0,2065	0,1580
	Kök Alınmış	0,2863	0,0939	0,0575	0,2151	0,2351	0,1854

Tablo 11'de sonuçları paylaşılan genetik algoritma yönteminde, siyaset kategorisi en iyi, ekonomi kategorisi ikinci en iyi skorları elde eden haber türleridir. Haber özetleri olarak bilgi paragrafı kullanılması, Rouge-S skorlarının çok düşük olmasına yol açmıştır. Bu gösterge, bilgi paragrafında ve içerikte geçen ortak kelime ikililerinin çok az olduğunu gösterir.

Tablo 12. Habercom Veri Setinden Elde Edilen LexRank Rouge Değerleri

TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	*Normal*	0,2608	0,0745	0,0222	0,2124	0,1213	0,1071
	Gereksizler Silinmiş	0,2639	0,0793	0,0242	0,2297	0,1300	0,1193
	Kök Alınmış	0,4091	0,1263	0,0465	0,3260	0,1780	0,1651
Magazin	*Normal*	0,2218	0,0735	0,0305	0,1674	0,1886	0,1339
	Gereksizler Silinmiş	0,2233	0,0904	0,0377	0,1929	0,2324	0,1622
	Kök Alınmış	0,3177	0,1127	0,0611	0,2420	0,2698	0,1956
Siyaset	*Normal*	0,3442	0,1301	0,0257	0,2743	0,1267	0,1105
	Gereksizler Silinmiş	0,3540	0,1508	0,0357	0,2942	0,1572	0,1346
	Kök Alınmış	0,4630	0,1773	0,0496	0,3586	0,1781	0,1528
Spor	*Normal*	0,2640	0,0974	0,0304	0,2149	0,1619	0,1275
	Gereksizler Silinmiş	0,3093	0,1298	0,0486	0,2693	0,2109	0,1713
	Kök Alınmış	0,3921	0,1568	0,0593	0,3092	0,2336	0,1867

Tablo 12'te LexRank yönteminin elde ettiği skorlara bakıldığı zaman diğerlerinde olduğu gibi en yüksek skorları elde eden tür siyaset kategorisidir. Spor ve ekonomi kategorileride ikinci sıradadır. Ancak ilginçtir ki Rouge-L metriği açısından en yüksek skoru genelde en düşük skorları almış olan magazin kategorisi elde etmiştir.

Tablo 13. Habercom Veri Setinden Elde Edilen GAA Rouge Değerleri

TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	*Normal*	0,2294	0,0690	0,0122	0,1833	0,0635	0,0649
	Gereksizler Silinmiş	0,2253	0,0721	0,0125	0,1947	0,0681	0,0704
	Kök Alınmış	0,3756	0,1081	0,0253	0,2933	0,1073	0,1102
Magazin	*Normal*	0,2018	0,0591	0,0178	0,1501	0,0979	0,0970
	Gereksizler Silinmiş	0,2058	0,0773	0,0208	0,1708	0,1178	0,1146
	Kök Alınmış	0,3269	0,1031	0,0434	0,2374	0,1631	0,1588
Siyaset	*Normal*	0,3382	0,1388	0,0194	0,2723	0,0865	0,0890
	Gereksizler Silinmiş	0,3507	0,1534	0,0249	0,2946	0,0988	0,1027
	Kök Alınmış	0,4712	0,1853	0,0358	0,3751	0,1220	0,1271
Spor	*Normal*	0,2220	0,0682	0,0138	0,1783	0,0825	0,0819
	Gereksizler Silinmiş	0,2445	0,0848	0,0196	0,2066	0,1035	0,1023
	Kök Alınmış	0,3661	0,1308	0,0402	0,2816	0,1411	0,1402

Gizli anlam analizi ile siyaset kategorisi hariç diğer türlerin skorları tablo 13'den anlaşılacağı gibi birbirine çok yakındır. Siyaset türünün skorları ise belirgin şekilde daha yüksektir.

Tablo 14. Habercom Veri Setinden Elde Edilen Jaccard TextRank Rouge Değerleri

TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	*Normal*	0,2556	0,0820	0,0323	0,2112	0,1320	0,1221
	Gereksizler Silinmiş	0,2505	0,0887	0,0408	0,2257	0,1544	0,1433
	Kök Alınmış	0,3616	0,1205	0,0629	0,2982	0,2025	0,1870
Magazin	*Normal*	0,1677	0,0530	0,0243	0,1341	0,1515	0,1175
	Gereksizler Silinmiş	0,1827	0,0639	0,0303	0,1579	0,1940	0,1440
	Kök Alınmış	0,2787	0,0904	0,0573	0,2133	0,2440	0,1884
Siyaset	*Normal*	0,2992	0,1181	0,0260	0,2482	0,1151	0,1116
	Gereksizler Silinmiş	0,3336	0,1479	0,0370	0,2875	0,1420	0,1389
	Kök Alınmış	0,4374	0,1815	0,0521	0,3542	0,1697	0,1675

Tablo 14. (devamı) Habercom Veri Setinden Elde Edilen Jaccard TextRank Rouge Değerleri

Spor	***Normal***	0,2318	0,0890	0,0378	0,1949	0,1563	0,1322
	Gereksizler Silinmiş	0,2657	0,1139	0,0517	0,2340	0,2062	0,1750
	Kök Alınmış	0,3782	0,1597	0,0814	0,3056	0,2449	0,2197

Jaccard uygulanmış TextRank algoritması bu veri setinde ortalama bir başarı göstermiştir (Tablo 14). Diğer yöntemlere benzer şekilde magazin kategorisine ait özetler düşük skorlar elde etmiştir.

Tablo 15. Habercom Veri Setinden Elde Edilen LCS TextRank Rouge Değerleri

TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	***Normal***	0,2459	0,0749	0,0332	0,2074	0,1381	0,1269
	Gereksizler Silinmiş	0,2430	0,0841	0,0407	0,2211	0,1558	0,1444
	Kök Alınmış	0,3515	0,1160	0,0629	0,2896	0,2026	0,1873
Magazin	***Normal***	0,1609	0,0471	0,0217	0,1284	0,1480	0,1130
	Gereksizler Silinmiş	0,1827	0,0626	0,0299	0,1562	0,1908	0,1421
	Kök Alınmış	0,2651	0,0881	0,0535	0,2033	0,2416	0,1856
Siyaset	***Normal***	0,2964	0,1154	0,0278	0,2460	0,1210	0,1174
	Gereksizler Silinmiş	0,3338	0,1488	0,0399	0,2921	0,1474	0,1448
	Kök Alınmış	0,4188	0,1717	0,0550	0,3438	0,1756	0,1723
Spor	***Normal***	0,2243	0,0899	0,0384	0,1891	0,1562	0,1323
	Gereksizler Silinmiş	0,2516	0,1070	0,0487	0,2236	0,2009	0,1691
	Kök Alınmış	0,3633	0,1546	0,0826	0,3067	0,2569	0,2282

Tablo 15'te LCS uygulanmış TextRank algoritmasının Habercom dokümanlarından elde ettiği sonuçlar paylaşılmıştır. En yüksek Rouge-1 ve Rouge-2 skoru siyaset kategorisindeki kökleri alınan dokümanların özetleri elde etmiştir. Rouge-S ve Rouge-L için ise spor kategorisinin kökleri alınan dökümanları en yüksek skorları alan türdür.

Tablo 16. Habercom Veri Setinden Elde Edilen TextRank Rouge Değerleri

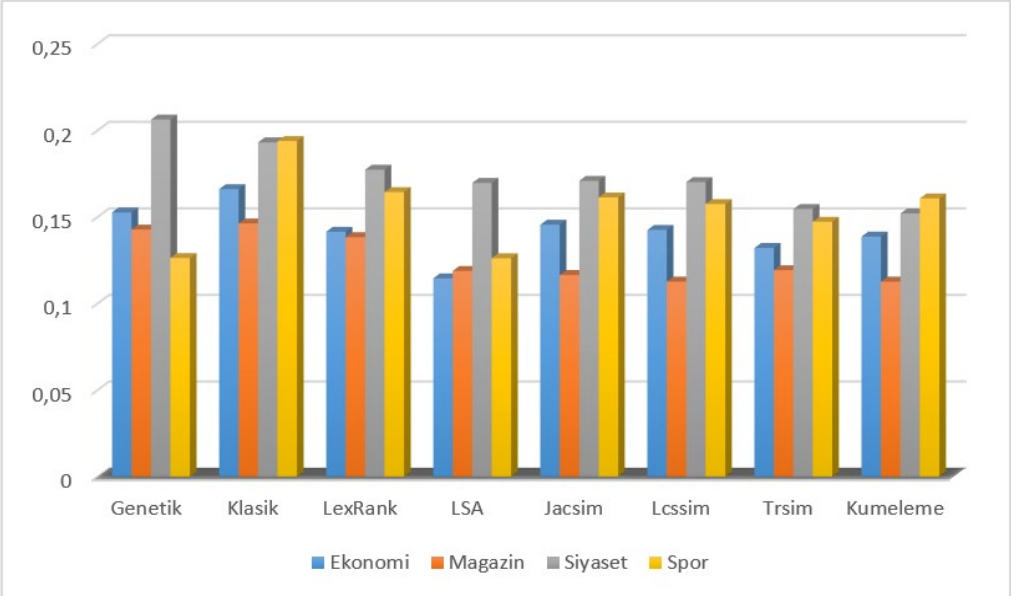
TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	***Normal***	0,2423	0,0721	0,0250	0,1968	0,1088	0,0999
	Gereksizler Silinmiş	0,2435	0,0862	0,0262	0,2150	0,1099	0,1063
	Kök Alınmış	0,3725	0,1185	0,0441	0,2895	0,1583	0,1489
Magazin	***Normal***	0,1957	0,0593	0,0216	0,1535	0,1287	0,1129
	Gereksizler Silinmiş	0,1968	0,0703	0,0236	0,1670	0,1484	0,1278
	Kök Alınmış	0,3011	0,0980	0,0498	0,2253	0,2029	0,1746
Siyaset	***Normal***	0,2968	0,1112	0,0161	0,2378	0,0906	0,0866
	Gereksizler Silinmiş	0,3233	0,1351	0,0226	0,2747	0,1071	0,1064
	Kök Alınmış	0,4346	0,1654	0,0322	0,3396	0,1273	0,1255
Spor	***Normal***	0,2292	0,0803	0,0245	0,1885	0,1180	0,1059
	Gereksizler Silinmiş	0,2750	0,1083	0,0398	0,2357	0,1550	0,1389
	Kök Alınmış	0,3815	0,1530	0,0579	0,2971	0,1914	0,1729

Orijinal TextRank algoritması, LCS uygulanmış versiyona benzer bir sonuç görüntüsü ortaya çıkartmıştır (Tablo 16). Tek fark Rouge-L metriğinde magazin türü diğerlerine göre en yüksek skoru elde etmiştir. Ancak genel olarak Rouge-2, Rouge-S ve Rouge-L skorlarının çok düşük olması, oluşturulan özetlerin bilgi paragrafına çok benzemediğini göstermektedir.

Tablo 17. Habercom Veri Setinden Elde Edilen Kümeleme Rouge Değerleri

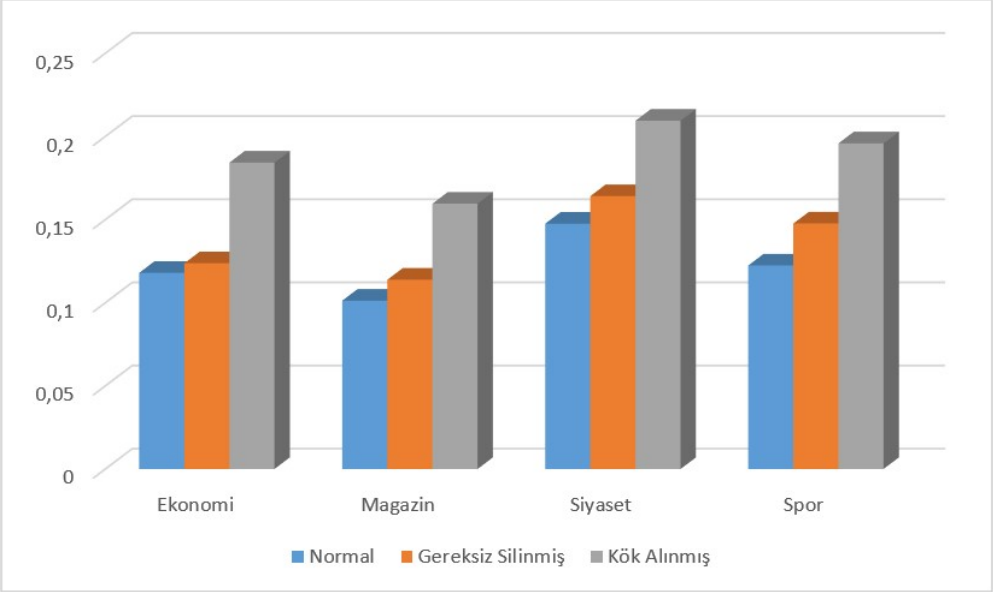
TÜR	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
Ekonomi	***Normal***	0,2499	0,0753	0,0290	0,2050	0,1270	0,1205
	Gereksizler Silinmiş	0,2438	0,0753	0,0336	0,2118	0,1421	0,1311
	Kök Alınmış	0,3610	0,1125	0,0552	0,2864	0,1836	0,1772
Magazin	***Normal***	0,1852	0,0533	0,0241	0,1423	0,1564	0,1212
	Gereksizler Silinmiş	0,1818	0,0633	0,0285	0,1504	0,1874	0,1362
	Kök Alınmış	0,2611	0,0787	0,0472	0,1911	0,2302	0,1717
Siyaset	***Normal***	0,2974	0,1175	0,0290	0,2371	0,1199	0,1155
	Gereksizler Silinmiş	0,2911	0,1188	0,0305	0,2412	0,1324	0,1265
	Kök Alınmış	0,3790	0,1359	0,0393	0,2910	0,1525	0,1443
Spor	***Normal***	0,2471	0,0983	0,0398	0,2025	0,1639	0,1363
	Gereksizler Silinmiş	0,2864	0,1232	0,0533	0,2418	0,2038	0,1719
	Kök Alınmış	0,3600	0,1444	0,0694	0,2837	0,2341	0,1993

Kümeleme algoritması, tablo 17'de görüleceği gibi magazin ve siyaset kategorilerinde en kötü sonuçları alırken, spor ve ekonomi kategorilerinde nispeten daha iyi sonuçlar elde etmiştir.



Şekil 39. Haber Kategorisine Göre Yöntemlerin Ortalama Başarıları

Şekil 39'daki algoritmalara kategori bazında başarılarına bakıldığında, klasik yöntem siyaset dışında diğer tüm kategorilerde en iyi özetleyici sistem olduğu gözükmektedir. Siyaset kategorisinde ise genetik algoritmanın diğerine göre ciddi bir oranda üstünlüğü vardır. Buna rağmen genetik algoritma spor kategorisinde en düşük skorları gösteren yaklaşım olmuştur.



Şekil 40. Habercom Veri Setinde Kategorilere Göre Ortalama Rouge Skorları

Şekil 40'ta, kategorilerin özet başarısına bakıldığı zaman ise siyaset haberlerinin en iyi, magazin haberlerinin ise en kötü olduğu görülmektedir. Bu veri setinde hem gereksiz kelimelerin temizlenmesi hem de kelime köklerinin alınması bariz bir şekilde özet başarısını olumlu yönde etkilemiştir.

3.4.4. Multiling Veri Setinden Elde Edilen Sonuçlar

Tez kapsamında kullanılan son veri MultiLing veri setinin 6 dili alınarak oluşturulmuştur. Her dil farklı bir veri seti ele alınarak algoritmalarla test edilmiştir. Ayrıca dokümanların 3 farklı tipte ifade edilmesi gereksiz kelimelerin ve kelime köklerinin dil açısından özetlemeye etkisini ortaya çıkartmıştır.

Tablo 18. MultiLing Veri Setinden Elde Edilen Klasik Yöntem Rouge Değerleri

DİL	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,3891	0,0778	0,0457	0,2448	0,0777	0,0832
	Gereksizler Silinmiş	0,2511	0,0444	0,0163	0,1533	0,0534	0,0576
	Kök Alınmış	0,3147	0,0464	0,0230	0,1772	0,0588	0,0631
İNG.	*Normal*	0,4232	0,1467	0,1620	0,2623	0,1452	0,1613
	Gereksizler Silinmiş	0,3287	0,1018	0,0851	0,2111	0,1292	0,1432
	Kök Alınmış	0,3730	0,1060	0,1038	0,2208	0,1322	0,1469
İSP.	*Normal*	0,3924	0,1539	0,1268	0,2846	0,1282	0,1413
	Gereksizler Silinmiş	0,3293	0,1019	0,0614	0,1998	0,0924	0,1019
	Kök Alınmış	0,3971	0,1094	0,0766	0,2148	0,0953	0,1048
FRA.	*Normal*	0,3896	0,1228	0,0931	0,2541	0,1059	0,1160
	Gereksizler Silinmiş	0,3116	0,0709	0,0516	0,1764	0,0847	0,0935
	Kök Alınmış	0,3715	0,0857	0,0627	0,1902	0,0865	0,0952
ITA.	*Normal*	0,4443	0,1065	0,0382	0,2518	0,0643	0,0677
	Gereksizler Silinmiş	0,2964	0,0650	0,0184	0,1730	0,0515	0,0548
	Kök Alınmış	0,3502	0,0606	0,0236	0,1778	0,0525	0,0559
TÜR.	*Normal*	0,2645	0,0576	0,0671	0,1427	0,0929	0,1023
	Gereksizler Silinmiş	0,2595	0,0661	0,0573	0,1394	0,0917	0,1005
	Kök Alınmış	0,3331	0,0738	0,0922	0,1656	0,1077	0,1184

Klasik yöntemin bu verisetinden elde ettiği sonuçlar tablo 18'de paylaşılmıştır. Herhangi bir temizliğe maruz kalmamış veriler içerisinde en yüksek Rouge-1 skoru İtalyanca diline ait görülmektedir. Kökleri alınmış cümlelerde ise İspanyolca dilinde en yüksek Rouge-1 skoru elde edilmiştir.

Tablo 19. MultiLing Veri Setinden Elde Edilen Genetik Algoritma Rouge Değerleri

DİL	Model	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,4349	0,0812	0,0137	0,2574	0,0613	0,0640
	Gereksizler Silinmiş	0,2826	0,0424	0,0064	0,1504	0,0341	0,0355
	Kök Alınmış	0,3602	0,0524	0,0098	0,1858	0,0438	0,0456
İNG.	*Normal*	0,4548	0,1389	0,0228	0,2729	0,1244	0,1371
	Gereksizler Silinmiş	0,3767	0,0989	0,0248	0,2196	0,0984	0,1084
	Kök Alınmış	0,4281	0,1003	0,0279	0,2245	0,1011	0,1114
İSP.	*Normal*	0,4391	0,1655	0,0133	0,3063	0,1009	0,1083
	Gereksizler Silinmiş	0,3443	0,0841	0,0132	0,1811	0,0567	0,0605
	Kök Alınmış	0,4302	0,0918	0,0173	0,2002	0,0622	0,0664
FRA.	*Normal*	0,4869	0,1614	0,0178	0,2770	0,0928	0,0996
	Gereksizler Silinmiş	0,3573	0,0982	0,0162	0,2000	0,0698	0,0752
	Kök Alınmış	0,4310	0,1146	0,0202	0,2208	0,0771	0,0831
ITA.	*Normal*	0,4910	0,1260	0,0126	0,2633	0,0582	0,0606
	Gereksizler Silinmiş	0,3076	0,0480	0,0085	0,1749	0,0467	0,0493
	Kök Alınmış	0,3794	0,0566	0,0100	0,1913	0,0465	0,0487
TÜR.	*Normal*	0,2814	0,0536	0,0176	0,1523	0,0767	0,0850
	Gereksizler Silinmiş	0,2656	0,0527	0,0172	0,1328	0,0672	0,0745
	Kök Alınmış	0,3317	0,0494	0,0210	0,1457	0,0740	0,0821

Genetik algoritmanın aynı veri setinden elde ettiği sonuçlara bakıldığında Rouge-1 skorları açısından klasik yöntemden daha başarılı olduğu görülmektedir. Ancak diğer ölçüt türleri için aynı şey söz konusu değildir. Italyanca dilinde alınan 0.49 Rouge-1 skoru bu yöntemin elde ettiği en yüksek skordur. Ayrıca özetin orijinal

dokümandan seçilen cümlelerle değilde yeniden ifade edilerek yazılması diğer ölçütlerin düşük olmasının sebebi olarak görülebilir.

Tablo 20. MultiLing Veri Setinden Elde Edilen LexRank Rouge Değerleri

DİL	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,4033	0,0874	0,0168	0,2514	0,0765	0,0815
	Gereksizler Silinmiş	0,2647	0,0455	0,0096	0,1606	0,0536	0,0575
	Kök Alınmış	0,3322	0,0564	0,0141	0,1850	0,0605	0,0648
İNG.	*Normal*	0,4103	0,1431	0,0289	0,2572	0,1466	0,1627
	Gereksizler Silinmiş	0,3202	0,0877	0,0315	0,1990	0,1293	0,1419
	Kök Alınmış	0,3774	0,1055	0,0385	0,2242	0,1432	0,1578
İSP.	*Normal*	0,4329	0,1885	0,0169	0,3000	0,1194	0,1302
	Gereksizler Silinmiş	0,3166	0,0980	0,0197	0,1999	0,0918	0,1012
	Kök Alınmış	0,4018	0,1121	0,0261	0,2172	0,0962	0,1056
FRA.	*Normal*	0,4624	0,1626	0,0222	0,2679	0,1061	0,1156
	Gereksizler Silinmiş	0,3059	0,0711	0,0205	0,1741	0,0835	0,0922
	Kök Alınmış	0,3729	0,0822	0,0259	0,1885	0,0876	0,0966
İTA	*Normal*	0,4513	0,1043	0,0137	0,2460	0,0618	0,0650
	Gereksizler Silinmiş	0,2855	0,0559	0,0103	0,1636	0,0535	0,0574
	Kök Alınmış	0,3512	0,0595	0,0130	0,1872	0,0573	0,0612
TÜR.	*Normal*	0,2902	0,0671	0,0285	0,1563	0,1061	0,1158
	Gereksizler Silinmiş	0,2569	0,0580	0,0253	0,1324	0,0931	0,1013
	Kök Alınmış	0,3333	0,0731	0,0334	0,1616	0,1080	0,1184

LexRank yönteminin tablo 20'deki elde ettiği skorlarından en iyi sonuçların Fransızca, İtalyanca ve İspanyolca dillerinden elde ettiği anlaşılmaktadır. Türkçe dilinden ise diğer yöntemlerde olduğu gibi bu yöntemde de düşük skorlar almıştır.

Tablo 21. MultiLing Veri Setinden Elde Edilen GAA Rouge Değerleri

DİL	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,3990	0,0785	0,0120	0,2341	0,0594	0,0624
	Gereksizler Silinmiş	0,2680	0,0490	0,0057	0,1488	0,0371	0,0389
	Kök Alınmış	0,3381	0,0585	0,0082	0,1761	0,0437	0,0459
İNG.	*Normal*	0,4102	0,1292	0,0233	0,2492	0,1182	0,1307
	Gereksizler Silinmiş	0,3377	0,0880	0,0224	0,1910	0,0902	0,0997
	Kök Alınmış	0,3899	0,0993	0,0273	0,2104	0,1001	0,1106
İSP.	*Normal*	0,4340	0,1829	0,0153	0,2992	0,1063	0,1149
	Gereksizler Silinmiş	0,3457	0,0966	0,0152	0,1881	0,0674	0,0729
	Kök Alınmış	0,4217	0,1057	0,0201	0,2098	0,0752	0,0813
FRA.	*Normal*	0,4516	0,1414	0,0178	0,2526	0,0882	0,0951
	Gereksizler Silinmiş	0,3149	0,0625	0,0119	0,1575	0,0548	0,0591
	Kök Alınmış	0,3735	0,0728	0,0153	0,1734	0,0605	0,0653
ITA.	*Normal*	0,4499	0,0996	0,0109	0,2378	0,0521	0,0542
	Gereksizler Silinmiş	0,3057	0,0548	0,0063	0,1627	0,0369	0,0385
	Kök Alınmış	0,3780	0,0612	0,0086	0,1857	0,0421	0,0439
TÜR.	*Normal*	0,2984	0,0672	0,0203	0,1481	0,0733	0,0812
	Gereksizler Silinmiş	0,2691	0,0602	0,0171	0,1253	0,0618	0,0685
	Kök Alınmış	0,3547	0,0723	0,0247	0,1504	0,0751	0,0832

Tablo 21'de sonuçları paylaşılan gizli anlam analizide diğer yöntemlere benzer olarak Rouge-1 metriğinde İtalyanca, Fransızca ve İspanyolca dillerinde iyi sonuç

verirken Türkçe dilinde düşük skorlar elde etmiştir. Rouge-2 açısından ise İngilizce dilinden nispeten iyi sonuçlar elde edilmiştir.

Tablo 22. MultiLing Veri Setinden Elde Edilen Jaccard TextRank Rouge Değerleri

DİL	Model	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,3165	0,0711	0,0215	0,2115	0,0949	0,1045
	Gereksizler Silinmiş	0,2042	0,0414	0,0132	0,1359	0,0704	0,0779
	Kök Alınmış	0,2398	0,0504	0,0171	0,1598	0,0835	0,0925
İNG.	*Normal*	0,3313	0,1135	0,0325	0,2242	0,1793	0,1901
	Gereksizler Silinmiş	0,2281	0,0650	0,0302	0,1700	0,1704	0,1617
	Kök Alınmış	0,2640	0,0757	0,0362	0,1914	0,1854	0,1808
İSP.	*Normal*	0,3268	0,1347	0,0260	0,2389	0,1695	0,1862
	Gereksizler Silinmiş	0,2270	0,0741	0,0229	0,1678	0,1339	0,1408
	Kök Alınmış	0,2861	0,0857	0,0309	0,1840	0,1407	0,1499
FRA.	*Normal*	0,3609	0,1171	0,0296	0,2184	0,1398	0,1533
	Gereksizler Silinmiş	0,2201	0,0548	0,0220	0,1496	0,1179	0,1249
	Kök Alınmış	0,2667	0,0642	0,0280	0,1650	0,1256	0,1337
ITA.	*Normal*	0,3570	0,0685	0,0163	0,2045	0,0749	0,0801
	Gereksizler Silinmiş	0,1812	0,0367	0,0110	0,1279	0,0747	0,0745
	Kök Alınmış	0,2325	0,0411	0,0132	0,1453	0,0753	0,0768
TÜR.	*Normal*	0,2160	0,0492	0,0236	0,1314	0,1168	0,1202
	Gereksizler Silinmiş	0,1811	0,0454	0,0207	0,1078	0,1123	0,1056
	Kök Alınmış	0,2306	0,0551	0,0294	0,1352	0,1461	0,1328

Jaccard uygulanmış TextRank algoritmasının farklı dillerden elde ettiği sonuçlar tablo 22'de gösterilmektedir. Bu yöntem, Rouge-1 ve Rouge-2 açısından diğerlerine göre düşük olmasına karşın Rouge-L değerleri yüksektir.

Tablo 23. MultiLing Veri Setinden Elde Edilen LCS TextRank Rouge Değerleri

DİL	Model	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,2980	0,0647	0,0227	0,2025	0,1034	0,1146
	Gereksizler Silinmiş	0,2015	0,0398	0,0133	0,1344	0,0727	0,0806
	Kök Alınmış	0,2309	0,0472	0,0177	0,1515	0,0854	0,0946
İNG.	*Normal*	0,3023	0,1028	0,0307	0,2181	0,1942	0,1987
	Gereksizler Silinmiş	0,2229	0,0629	0,0291	0,1654	0,1736	0,1601
	Kök Alınmış	0,2565	0,0702	0,0351	0,1825	0,1883	0,1774
İSP.	*Normal*	0,3085	0,1252	0,0260	0,2380	0,1859	0,1996
	Gereksizler Silinmiş	0,2108	0,0729	0,0217	0,1649	0,1373	0,1435
	Kök Alınmış	0,2610	0,0797	0,0285	0,1763	0,1466	0,1534
FRA.	*Normal*	0,3232	0,1061	0,0320	0,2078	0,1642	0,1745
	Gereksizler Silinmiş	0,2113	0,0513	0,0220	0,1449	0,1231	0,1279
	Kök Alınmış	0,2509	0,0593	0,0287	0,1569	0,1360	0,1389
ITA.	*Normal*	0,3285	0,0685	0,0206	0,1947	0,0899	0,0960
	Gereksizler Silinmiş	0,1723	0,0396	0,0107	0,1271	0,0797	0,0791
	Kök Alınmış	0,2192	0,0396	0,0140	0,1398	0,0790	0,0820
TÜR.	*Normal*	0,2098	0,0402	0,0291	0,1264	0,1243	0,1253
	Gereksizler Silinmiş	0,1439	0,0122	0,0129	0,1111	0,1545	0,1229
	Kök Alınmış	0,2211	0,0515	0,0293	0,1283	0,1504	0,1314

Jaccard uygulanmış TextRank'a benzer şekilde LCS uygulanmış TextRank algoritması da MultiLing verisetindeki farklı dillerdeki dokümanlarda genellikle

birbirlerine yakın skorlar elde etmiştir. Rouge-L açısından ise diğer algoritmalara göre daha iyi skorlar elde ettiği görülmektedir (Tablo 23).

Tablo 24. MultiLing Veri Setinden Elde Edilen TextRank Rouge Değerleri

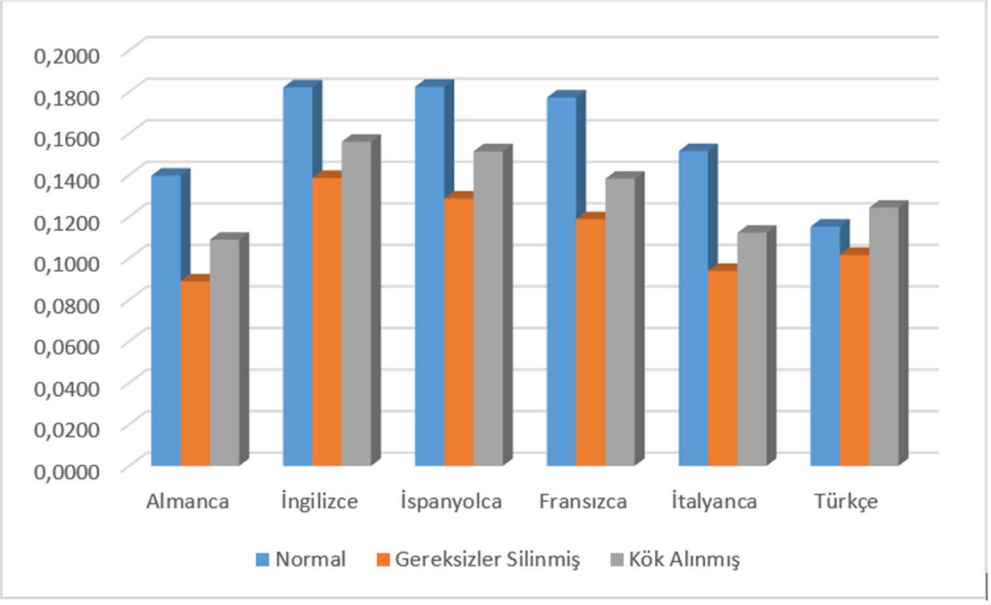
DİL	Model	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,3930	0,0833	0,0155	0,2384	0,0702	0,0746
	Gereksizler Silinmiş	0,2514	0,0469	0,0092	0,1510	0,0501	0,0537
	Kök Alınmış	0,3109	0,0577	0,0127	0,1826	0,0605	0,0648
İNG.	*Normal*	0,4033	0,1306	0,0282	0,2497	0,1344	0,1494
	Gereksizler Silinmiş	0,2912	0,0808	0,0287	0,1902	0,1246	0,1373
	Kök Alınmış	0,3379	0,0942	0,0349	0,2092	0,1355	0,1494
İSP.	*Normal*	0,4254	0,1786	0,0169	0,2944	0,1133	0,1233
	Gereksizler Silinmiş	0,3086	0,0999	0,0194	0,2003	0,0914	0,1008
	Kök Alınmış	0,3712	0,1080	0,0241	0,2148	0,0954	0,1049
FRA.	*Normal*	0,4502	0,1545	0,0205	0,2594	0,0984	0,1069
	Gereksizler Silinmiş	0,3088	0,0740	0,0189	0,1778	0,0815	0,0896
	Kök Alınmış	0,3596	0,0840	0,0228	0,1935	0,0871	0,0958
ITA.	*Normal*	0,4478	0,1006	0,0126	0,2414	0,0574	0,0602
	Gereksizler Silinmiş	0,2615	0,0535	0,0087	0,1564	0,0490	0,0524
	Kök Alınmış	0,3373	0,0595	0,0116	0,1827	0,0543	0,0578
TÜR.	*Normal*	0,3143	0,0717	0,0320	0,1600	0,1071	0,1171
	Gereksizler Silinmiş	0,2692	0,0643	0,0232	0,1493	0,0931	0,1027
	Kök Alınmış	0,2322	0,0603	0,0215	0,1281	0,0885	0,0966

Orijinal TextRank algoritması, tablo 24'teki sonuçlarına göre farklı benzerlik yöntemleri uygulanmış versiyonlarından daha başarılıdır. Ayrıca diğer yöntemlerde olduğu gibi İtalyanca ve Fransızca dillerine ait dökümanların özetleri diğer dillere göre daha yüksek Rouge skorları elde etmiştir.

Tablo 25. MultiLing Veri Setinden Elde Edilen Kümeleme Rouge Değerleri

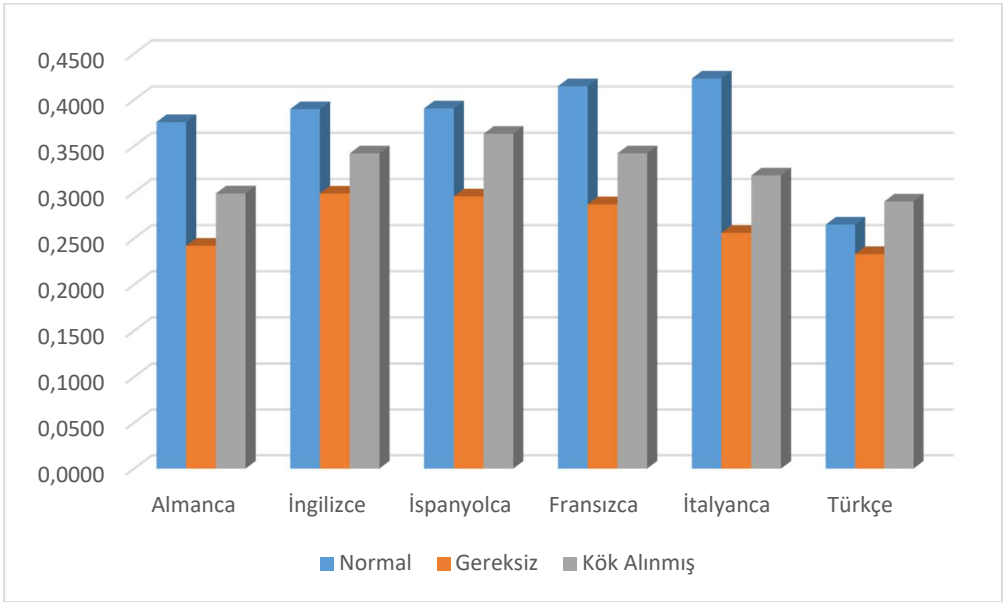
DİL	MODEL	Rouge-1	Rouge-2	Rouge-S	Rouge-L Hassasiyet	Rouge-L Keskinlik	Rouge-L F skor
ALM.	*Normal*	0,3741	0,0719	0,0210	0,2222	0,0850	0,0923
	Gereksizler Silinmiş	0,2134	0,0387	0,0123	0,1342	0,0632	0,0699
	Kök Alınmış	0,2633	0,0511	0,0169	0,1593	0,0749	0,0828
İNG.	*Normal*	0,3850	0,1308	0,0335	0,2364	0,1650	0,1804
	Gereksizler Silinmiş	0,2840	0,0730	0,0342	0,1720	0,1403	0,1490
	Kök Alınmış	0,3105	0,0724	0,0359	0,1810	0,1500	0,1578
İSP.	*Normal*	0,3676	0,1498	0,0233	0,2520	0,1492	0,1652
	Gereksizler Silinmiş	0,2822	0,0850	0,0268	0,1617	0,1085	0,1195
	Kök Alınmış	0,3377	0,0922	0,0329	0,1778	0,1194	0,1307
FRA.	*Normal*	0,3940	0,1224	0,0290	0,2212	0,1288	0,1430
	Gereksizler Silinmiş	0,2658	0,0610	0,0254	0,1411	0,1017	0,1093
	Kök Alınmış	0,3110	0,0637	0,0302	0,1586	0,1080	0,1173
İTA.	*Normal*	0,4152	0,0870	0,0172	0,2260	0,0715	0,0765
	Gereksizler Silinmiş	0,2379	0,0436	0,0116	0,1373	0,0615	0,0675
	Kök Alınmış	0,2981	0,0570	0,0161	0,1546	0,0673	0,0738
TÜR.	*Normal*	0,2454	0,0528	0,0281	0,1254	0,1100	0,1131
	Gereksizler Silinmiş	0,2192	0,0484	0,0257	0,1116	0,1026	0,1039
	Kök Alınmış	0,2840	0,0537	0,0332	0,1304	0,1160	0,1184

Son yöntem olan kümelemeye ait Rouge skorları tablo 25'te paylaşılmıştır. Kümeleme yönteminde Rouge-1'de İtalyanca, Rouge-2'de İspanyolca, Rouge-L ölçütünde İngilizce dilleri en iyi skorların alındığı dillerdir.



Şekil 41. Dillere Göre Ortalama Skorlar

MultiLing veri setinin şekil 42'deki ortalama sonuçlara bakıldığı zaman skorların genellikle düşük kaldığı görülmektedir. Bunun temel sebebi doküman özetlerinin 15 cümle ile sınırlandırılmış olması gösterilebilir.

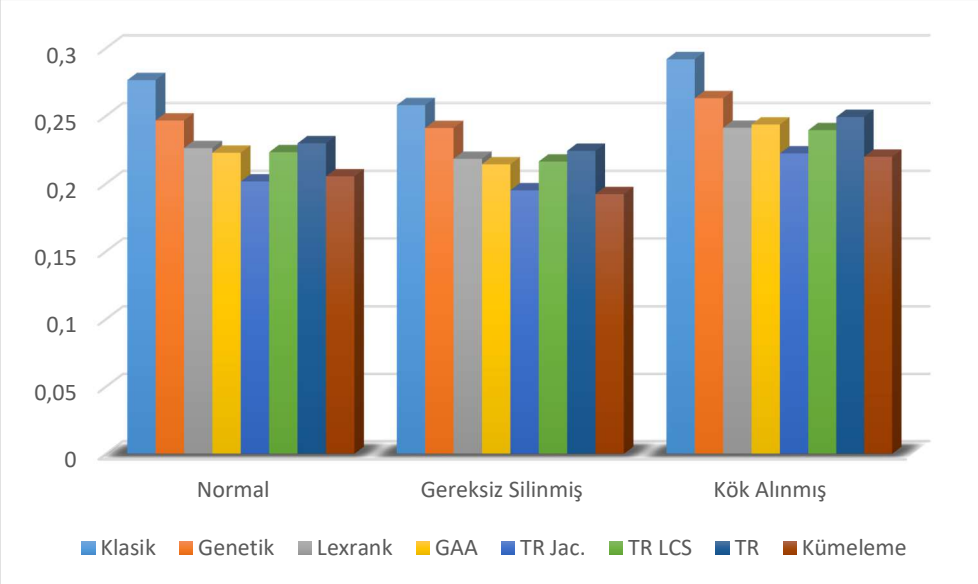


Şekil 42. Dillere Göre Ortalama Rouge-1 Skorları

Dillere göre ortalam Rouge-1 skorlarının paylaşıldığı şekil 42'e göre temizlenmemiş dokümanlar arasında Rouge-1 ölçütü açısından en yüksek skorlar İtalyanca dilinden elde edilmiştir. Sonrasında ise Fransızca dili gelmektedir. Ancak şekil 41'deki genel ortalamaya bakıldığı zaman en yüksek skorlar sırasıyla İngilizce ve İspanyolca dillerindeki dökümanlara aittir. Bunun sebebi olarak Fransızca ve İtalyanca dilleri üzerinde kelime bölme ve en uzun ortak altdizi algoritmalarının yeterince verimli ve doğru çalışmadığı olarak belirtilebilir. Türkçe dili ise tüm metrikler açısından en düşük skorları elde eden dil olmuştur.

Elde edilen sonuçlar dil açısından karşılaştırıldığı zaman Türkçe, Almanca ve İtalyanca dilinde oluşturulan sistem özetlerinin başarılı olmadığı gözlemlenmiştir.

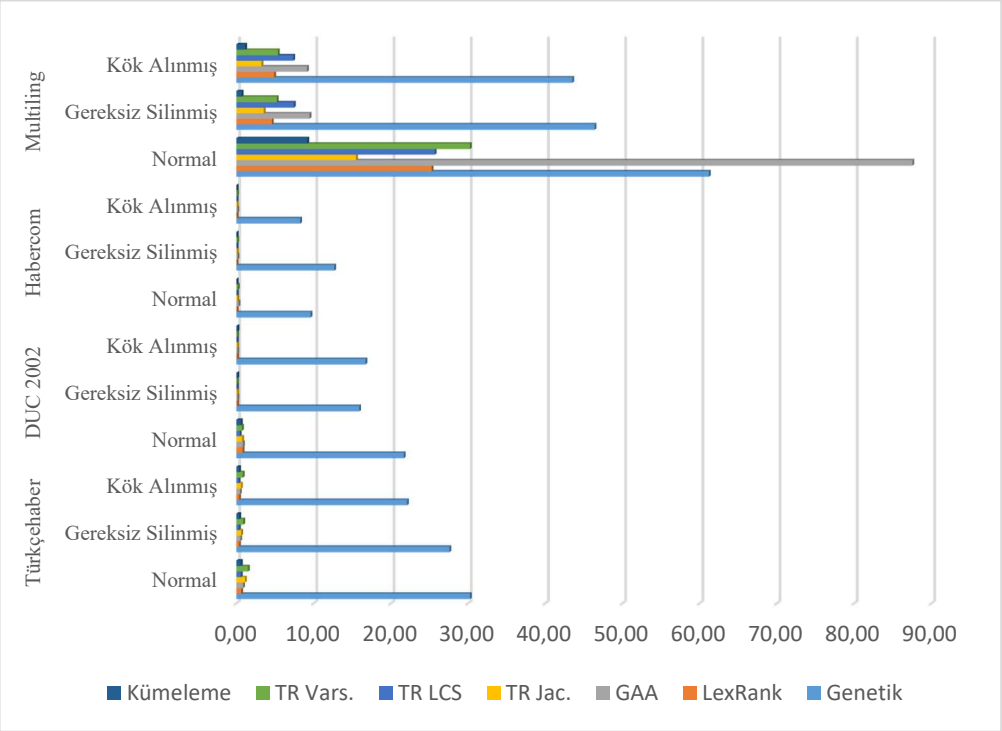
Ortaya çıkan önemli bulgulardan bir diğeri gereksiz kelimelerin temizlenmesinin ve kök alınmasının Türkçe hariç tüm dillerde başarıyı olumsuz şekilde etkilemiş olmasıdır. Uygulamada kullanılan diğer veri setlerinde kök alınmış modelin genellikle daha başarılıydı. Bu bulgu ışığında kök alınmış modelin düşük performansı MultiLing veri setine özgü olduğu söylenebilir.



Şekil 43. Algoritmaların Tüm Veriler Üzerindeki Genel Sonuçları

Şekil 43'de algoritmaların temizleme tiplerine göre tüm veri seti üzerindeki ortalama sonuçları çizelge ile gösterilmiştir. Buradan görüldüğü üzere klasik yaklaşım genel ortalamada en iyi özetleme yöntemidir. Bu yöntemi diğer bir özellik tabanlı

yaklaşım olan genetik algoritma ve TextRank algoritmasaı izlemektedir. Jaccard uygulanmış TextRank ve kümeleme ise en düşük skorları elde eden yöntemler olmuşlardır.



Şekil 44. Algoritmaların Çalışma Sürelerine İlişkin Çizelge (Saniye/Doküman)

Makine öğrenmesi konularında elde edilen sonuçlar kadar yöntemlerin çalışma süreleri de genel başarısını etkileyen bir faktördür. Bu durum özetleme sistemi içinde geçerlidir. Bir sistem ne kadar iyi sonuç verirse versin istenilen süre içerisinde özeti vermeyecek kadar yavaş çalışıyorsa efektif değildir ve teorik başarının ötesine geçemez.

Tablo 26. Algoritmaların Çalışma Sürelerine İlişkin Bilgiler (Saniye/Doküman)

Veriseti	Model	Genetik	LexRank	GAA	TR Jac.	TR LCS	TR Vars.	Kümeleme
Türkçehaber	***Normal***	30,25	0,63	0,88	1,12	0,59	1,51	0,58
	Gereksiz Silinmiş	27,60	0,35	0,52	0,63	0,34	0,90	0,38
	Kök Alınmış	22,11	0,37	0,48	0,60	0,31	0,84	0,36
DUC 2002	***Normal***	21,68	0,82	0,86	0,76	0,45	0,74	0,60
	Gereksiz Silinmiş	15,91	0,11	0,15	0,11	0,10	0,09	0,14
	Kök Alınmış	16,72	0,11	0,14	0,11	0,10	0,11	0,15
Habercom	***Normal***	9,61	0,06	0,28	0,13	0,08	0,19	0,06
	Gereksiz Silinmiş	12,69	0,04	0,16	0,08	0,06	0,11	0,05
	Kök Alınmış	8,27	0,04	0,14	0,08	0,06	0,10	0,05
Multiling	***Normal***	61,21	25,30	87,54	15,50	25,70	30,26	9,20
	Gereksiz Silinmiş	46,40	4,60	9,49	3,55	7,45	5,24	0,70
	Kök Alınmış	43,51	4,90	9,16	3,27	7,38	5,38	1,15

Uygulama esnasında elde edilen çalışma süreleri tablo 26'ta, karşılaştırılmalı çizelgesi ise şekil 44'te saniye/ doküman cinsinden gösterilmiştir. Sürelere bakıldığı zaman genellikle bir doküman 1 saniyeden az zamanda özetlenmiştir. İstisnai olarak MultiLing veri seti diğerlerine göre daha yavaştır. MultiLing veri setindeki dokümanların çok uzun olması (300-700 cümle) bu durumu tetikleyen ana unsurdur.

4. SONUÇ

İnternet çağıyla beraber erişilebilir bilgi kaynağı daha önceki zamanlarda olmadığı kadar yüksektir. Fakat birçok kaynak içerisinden, kullanıcının ihtiyaç duyduğunun bulunup belirlenmesi zaman alıcı ve zor iş olması nedeniyle bir sorun haline dönüşmüştür. Bu problemin çözümlerinden birisi dokümanların önemli kısımlarını tutarak kısaltan otomatik özetleme sistemleridir.

Bu tez çalışmasında, çıkarıcı türündeki otomatik doküman özetleme yöntemleri detaylı bir şekilde ele alınmıştır. Bir özetleyici sistemin nasıl olması gerektiğinin yanı sıra literatürde ortaya atılan ve önemli başarı elde eden 9 özetleme yaklaşımı örneklerle incelenmiştir. Yaklaşımlar hem Türkçe hem İngilizce hem de farklı dillerden oluşan 4 ayrı veri seti üzerinde test edilerek yöntemlerin başarısı, gereksiz kelimelerin, kök alma işleminin ve dilin etkisi ortaya koyulmuştur.

Sonuç tablo ve şekillerine bakıldığı zaman ilk göze çarpan bulgu, ön temizleme işlemlerinin özetleme başarısı üzerindeki etkileridir. Gereksiz kelimelerin temizlenmesi neredeyse tüm algoritma ve veri setlerinde Rouge değerlerinin düşmesine neden olmuştur. Bu duruma sebep olabilecek etkenlerden birisi sistem tarafından gereksiz kelimelerden arındırılarak oluşturulan özet cümlelerinin, arındırılmamış orijinal özetlerle karşılaştırılması olarak düşünülebilir. Ancak uygulamamızda, gereksiz kelime etkisinin tam ölçümü için özetler de gereksiz kelimelerden temizlenerek değerlendirilmiştir. Bu bilgiler çerçevesinde gereksiz kelimelerin temizlenmesinin özetleme başarısına olumsuz bir etki yaptığı söylenebilir. En azından kullanılan veri setleri için durum böyledir.

Bir diğer temizleme işlemi olan kelimelerin köklerinin alınması, MultiLing veri seti hariç, beklendiği gibi özet başarısını arttırmıştır. Özellikle Türkçe dokümanlarda %10 - %50 arasında, İngilizcede ise %3 - %7 arasında artış görülmüştür. Aynı yaklaşımın iki farklı dile etkisinde bu denli uyuşmazlık olması iki şekilde yorumlanabilir. İlki Türkçe veri seti için kullanılan İTÜ DDİ aracının Türkçe diline tam uyumlu bir şekilde geliştirilmesi; buna zıt olarak İngilizce kök alma kütüphanelerinin yetersiz kalmasıdır. İkincisi ise sondan eklemeli bir dil olan Türkçe'de birçok kelimenin kökleri belirli bazı eklerin (*"-ler", "-dı", "-mak"* gibi) kaldırılmasıyla bulunabilirken, İngilizce veya diğer diller için sürecin bu kadar kolay

olmamasıdır. MultiLing veri setinde dil olarak sadece Türkçe'de kök alınmasının başarıyı arttırması da bu düşünceyle açıklanabilir. Sonuç olarak diğer diller için sözlük bazlı olsun veya olmasın kök bulucu sistemler yetersiz durumdadır.

Temizleme işlemleri genellikle özet başarısını arttırmak ve işlem süresini düşürme amacıyla yapılır. Kullanılan YSA modelinde ise temizlik seviyesi arttıkça özette geçen cümlenin tahmin oranı azalmış, geçmeyen cümlelerin tahmin oranı ise yükselmiştir. Bu oranlar, tablo 8'deki hassasiyet sonucunun düşerken, doğru negatif değer oranın artmasından da anlaşılmaktadır. Bu sonuçla, özette geçmeyen cümlelerin tespitinde temizlik önişleminin olumlu etkisi olduğu saptanmıştır. Genel başarı oranına bakıldığında ise temizliğin etkisi 120 Türkçe haber veri setine pozitif yönde olmuştur.

120 Türkçe haber veri setinin özetleri, orijinal dokümanlardan birebir seçilen cümlelerden oluşması sebebiyle elde edilen skorlar diğer veri setlerindekilere göre yüksektir. Ancak bu veri seti kullanılarak yapılan diğer çalışmamızda (Kaynar vd. 2017) Rouge skorları tezde elde edilenlerden çok daha yüksek çıkmıştır. Bunun temel nedeni, çalışmada sistemin üreteceği özet uzunluğunun orijinal özetteki uzunlukla eşit tutulmasıdır. Tez çalışmasında ise literatürdeki kabul gören %20 oran seçilmiştir. Ancak tez uygulamasında elde edilen sonuçlara bakıldığı zaman özellikle Tükçe dilindeki haberler için %20 oranının yeterli olmadığı savunulabilir. Bunun yerine bir uzman görüşüyle bir haber özeti için olması gereken uzunluk belirlenip buna göre özetleme sistemi modeli oluşturulması daha doğru bir adım olacaktır.

Bir diğer Türkçe veri seti olan Habercom dokümanlarının elde ettiği ortalama skorların düşük olduğu görülmektedir. Aslında bu veri seti özetleme başarısını ölçmek için tam uygun değildir. Çünkü özet olarak haberleri tanıtıcı bilgi paragrafları kullanılmıştır. Fakat yapılan testler, kısa bilgi paragraflarının haberin içeriğini yeterince yansıtmadığını ortaya koymuştur. Ayrıca sonuç tablolarında genetik algoritmanın bu veri seti üzerinde dalgalı bir Rouge değeri elde ettiği görülmektedir. Örneğin genetik algoritma siyaset kategorisinde en yüksek skorları alırken, spor kategorisinde en düşük sonuçları elde etmiştir. Bu sebeple farklı kategorilerdeki dokümanlar için alana özgü, spesifik özetleyici sistemlerin oluşturulmasının daha doğru olacağı sonucuna varılmıştır.

DUC veri setinde Rouge-1 metriğinin yüksek çıkmasına rağmen Rouge-S ve Rouge-L metriklerinin düşük olması, özette geçen ancak orijinal dokümanda geçmeyen kelimelerin çok olmasına bağlanabilir. Örneğin bir dokümanda geçen "*hurricane*" kelimesi ile dokümanın özetinde geçen "*storm*" kelimesi, benzer anlamda kullanılmasına rağmen örtüşmedikleri için Rouge metriği bunları alakasız kabul eder. Dokümandan birebir seçilen cümlelerle özetlerin oluşturduğu 120 Türkçe haber veri setine bakıldığında ise Rouge-1, Rouge-2 ve Rouge-L metriklerinin birbirleriyle doğru orantılı olduğu görülmektedir. Çünkü kelimeler birebir örtüşmektedir. Bu iki olgu özetlerin karşılaştırılırken salt kelime örtüşmesinin yeterli değerlendirme kriterini sağlamadığını gösterir.

MultiLing verisetinin sonuçlarına bakıldığında farklı dillerin genel özetleticiler üzerindeki etkisi görülebilmektedir. Diğer diller arasında özellikle Türkçe ve Almanca dillerinin özetleme başarısı düşük kalmıştır. Fransızca ve İspanyolca dillerinde ise hem Rouge-1 hem de ortalama başarı açısından yüksek skorlar elde edilmesi, bu dillerdeki dokümanlardan elde edilecek sistem özetlerinin orijinal özete benzer olacağı çıkarımı yapılabilir.

Uygulama sonuçlarına algoritmaların başarı penceresinden bakacak olursak, klasik yöntemin eski olmasına rağmen yeni yöntemlerden daha iyi sonuçlar elde ettiği görülebilir. Veri setlerinin farklı tüm tiplerinde (normal, gereksiz kelime temizlenmiş, kök alınmış) klasik yöntem diğer yöntemlerden daha yüksek skorlar elde etmiştir. Bu yöntemi genel olarak bir diğer öznitelikleri kullanan yaklaşım olan genetik algoritma takip etmiştir. Bu bulgular ışığında dokümanların yapısal ve içeriksel bazı özniteliklerle ifade edilmesinin doğru olduğunu göstermektedir.

Diğer bir algoritma grubu olan TexRank, Jaccard uygulanmış TextRank, LCS uygulanmış TextRank ve LexRank yöntemlerine bakıldığı zaman aralarında en iyinin varsayılan benzerlik yöntemi yani orijinal TextRank algoritması olduğu belirgindir. Buna rağmen "*en önemli cümle diğer cümlelere en çok benzeyendir*" fikri ile ortaya atılan TextRank algoritmaları genel anlamda başarılı olamamıştır. Sonuç olarak, bir cümlenin diğer cümlelere benzemesinin dokümanı temsil etme açısından tek başına yeterli olmadığı anlaşılmıştır. Ayrıca doküman içerisinden benzer olmayan cümlelerin

seçilerek daha kapsayıcı bir özet oluşturulacağı temeline dayanan kümeleme algoritması, en düşük skorları elde eden yöntemlerden birisidir.

Anlamsal seviyede benzerliği yansıtan GAA yönteminde de istenilen kadar yüksek başarılar elde edilememiştir. Testler sonucunda kelime bazında anlamsal benzerliğin iyi bir şekilde yakalanmasına rağmen cümle bazında yeterince iyi olmadığı gözlemlenmiştir. Dokümanları tekil olarak değilde bütünce anlamsal uzayda temsil edip cümle benzerliklerini incelemek başarıyı arttırmaya yönelik bir hamle olabilir.

Bu alanda yapılacak sonraki çalışmalarımızda eşanlamlı olup eşsesli olmayan bir ölçütün değerlendirme kriteri olarak eklenmesi, devamında özet başarısının daha doğru şekilde belirlenmesi hedeflenecektir. Bununla beraber, cümleleri niteleyebilecek yeni öznitelikler üzerinde durularak cümle seçimi aşamasında daha doğru bir yol izlenilmesi amaçlanmıştır. Ayrıca farklı diller için daha uygun kök bulucu kütüphanelerin geliştirilmesi üzerine yoğunlaşılarak, Türkçe dilindeki etki diğer dillerde sağlanmaya çalışılacaktır. Tez kapsamında ele alınan özetleme yaklaşımlarının, web veya mobil uygulama ile kullanıcının hizmetine sunularak pratik başarının ölçülmesi ileriki çalışmalarımızdan birisidir. Tüm bunların yanı sıra özetlemenin resim, video gibi diğer medyalara uygulanabilmesi konusunda araştırmalar yapılarak sağladığı çeşitli avantajlardan bu alanlarda da yararlanılacaktır.

KAYNAKLAR

Agirre, Eneko, Soroa Aitor ve Stevenson Mark (2010). "Graph-based Word Sense Disambiguation of biomedical documents". *Bioinformatics* 26 (22):2889–2896.

Akar Özlem ve Güngör Oğuz (2012). "Rastgele orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması". *Jeodezi ve Jeoinformasyon Dergisi.* 139–146.

Alguliev Rasim M., Aliguliyev Ramiz M., Hajirahimova Makrufa S., ve Mehdiyev Chingiz A. (2011). "MCMR: Maximum coverage and minimum redundant text summarization model". *Expert Systems with Applications* 38 (12):14514–14522.

Altan, Zeynep (2004). "A Turkish automatic text summarization system". *Proceedings of IASTED International Conference on Artificial Intellegence and Applications*.

"American National standard for writing abstracts" (1977). *IEEE Transactions on Professional Communication* PC-20 (4):252–254.

Andrews Harry C. ve Patterson Claude L. (1976). "Singular value decompositions and digital image processing". *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24 (1):26–53.

Aristoteles Herdiyeni Y., Ridha Ahmad ve Adisantoso Julio (2012). "Text feature weighting for summarization of documents in bahasa indonesia using genetic algorithm". *International Journal of Science Issues. ISSN*, 1694-0814.

Azmi Aqil M. ve Al-Thanyyan Suha (2012). "A text summarizer for Arabic". *Computer Speech & Language* 26 (4):260–273.

Baxendale, P. B. (1958). "Machine-Made Index for Technical Literature: An Experiment". *IBM Journal of Research and Development* 2 (4):354–361.

Belica, Mišo (2017). *Sumy*. 28 Kasım 2017 tarihinde *https://github.com/miso-belica/sumy* adresinden erişildi.

Bellaachia Abdelghani ve Al-Dhelaan Mohammed (2014). "Multi-document hyperedge-based ranking for text summarization". *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 1919–1922. ACM.

Bellare Kedar, Sarma Anish Das, Sarma Atish Das, Loiwal Navneet, Mehta Vaibhav, Ramakrishnan Ganesh ve Bhattacharyya Pushpak (2004). "Generic Text Summarization Using WordNet." *LREC*.

Berry Micheal Q, Dumais Susan T. ve W. O'Brien Gavin (1995). "Using Linear Algebra for Intelligent Information Retrieval". *SIAM Review* 37 (4):573–595.

Binwahlan Mohammed S., Salim Naomie, ve Suanmali Ladda (2009). "Swarm Based Text Summarization". *2009 International Association of Computer Science and Information Technology - Spring Conference*, 145–150.

Bird, Steven (2017). *NLTK* (sürüm 3.2.5). 15 Ekim 2017 tarihinde *http://nltk.org/* adresinden erişildi.

Brandow Ronald, Mitze Karl ve Rau Lisa F. (1995). "Automatic condensation of electronic publications by sentence selection". *Information Processing & Management*, Summarizing Text, 31 (5):675–685.

Conroy John M. ve O'Leary Dianne P. (2001). "Text Summarization via Hidden Markov Models". *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 406–407. SIGIR '01. New York, NY, USA: ACM.

Dang Chenghua, Luo Xinjun ve Zhang Haibin (2008). "WordNet-based summarization of unstructured document". *WSEAS Transactions on Computers* 7 (Eylül):1467–1472.

Document Understanding Conferences. (2002). 11 Aralık 2017 tarihinde *http://duc.nist.gov/* sitesinden erişildi.

Dumais Susan T. (2004). "Latent semantic analysis". *Annual review of information science and technology* 38 (1):188–230.

Edmundson, Harold P. (1969). “New Methods in Automatic Extracting”. *J. ACM* 16 (2):264–285.

Edmundson Harold P. ve Wyllys Roland E. (1961). “Automatic Abstracting and Indexing—Survey and Recommendations”. *Commun. ACM* 4 (5):226–234.

ElementTree (sürüm 3.6) (2017) 05 Kasım 2017 tarihinde *https://docs.python.org/2/library/xml.etree.elementtree.html* adresinden erişildi.

Enríquez Fernando, Troyano José A. ve López-Solaz Tomás (2016). “An approach to the use of word embeddings in an opinion classification task”. *Expert Systems with Applications* 66 (Supplement C):1–6.

Erkan Günes ve Radev Dragomir R. (2004). “Lexrank: Graph-based lexical centrality as salience in text summarization”. *Journal of Artificial Intelligence Research* 22:457–479.

Eryigit, Gülsen (2014). “ITU Turkish NLP Web Service.” *EACL*, 1–4. Gothenburg, Sweden: Association for Computational Linguistics.

Fejer Hamzah Noori ve Omar Nazlia (2014). “Automatic Arabic text summarization using clustering and keyphrase extraction”. *Proceedings of the 6th International Conference on Information Technology and Multimedia*, 293–298.

Fellbaum, Christiane (1998). *WordNet: An Electronic Lexical Database*. MIT Press

Felzenszwalb Pedro F. ve Huttenlocher Daniel P. (2004). “Efficient Graph-Based Image Segmentation”. *International Journal of Computer Vision* 59 (2):167–181.

Foltz Peter W., Kintsch Walter ve Landauer Thomas K. (1998). “The measurement of textual coherence with latent semantic analysis”. *Discourse Processes* 25 (2–3):285–307.

Furnas George. W., Deerwester Scott, Dumais Susan T., Landauer Thomas K., Harshman Richard A., Streeter Lynn A. ve Lochbaum Karen E. (1988). “Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure”. *Proceedings of the 11th Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval, 465–480. SIGIR ’88. New York, NY, USA: ACM.

García-Hernández René Arnulfo, Montiel Romyna, Ledeneva Yulia, Rendón Eréndira, Gelbukh Alexander ve Cruz Rafael (2008). “Text Summarization by Sentence Extraction Using Unsupervised Learning”. *Proceedings of the 7th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence*, 133–143. MICAI ’08. Berlin.

Gong Yihong ve Liu Xin (2001). “Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis”. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 19–25. SIGIR ’01. New York, NY, USA: ACM.

Graesser Arthur C., VanLehn Kurt, Rosé Carolyn P., Jordan Pamela W. ve Harter Derek (2001). “Intelligent tutoring systems with conversational dialogue”. *AI magazine* 22 (4):39.

Güran Aysun, Bayazit Nilgün G. ve Bekar Eren (2011). “Automatic summarization of Turkish documents using non-negative matrix factorization”. *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 480–484.

Hirao Tsutomu, Sasaki Yutaka, Isozaki Hideki ve Maeda Eisaku (2002). “NTT’s Text Summarization System for DUC-2002”. *In Proceedings of the Document Understanding Conference*, 104–107.

Holland, John H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press.

Hovy Eduard ve Lin Chin-Yew (1998). “Automated Text Summarization and the SUMMARIST System”. *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, 197–214. TIPSTER ’98. Stroudsburg, PA, USA: Association for Computational Linguistics.

Inc, Diffeo (2017). *many-stop-words: stop words lists in many languages* (sürüm 0.2.2). 25 Kasım 2017 tarihinde *https://pypi.python.org/pypi/many-stop-words* adresinden erişildi.

Ansamma John ve Wilscy M. (2013). “Random forest classifier based multi-document summarization system”. *2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 31–36.

Kalman, Dan (2002). “A Singularly Valuable Decomposition: The SVD of a Matrix”. The American University.

Kavzoğlu Taşkın ve Çölkesen İsmail (2010). “Karar Ağaçları İle Uydu Görüntülerinin Sınıflandırılması”: *Harita Teknolojileri Elektronik Dergisi* 2 (1):36–45.

Kaynar Oğuz, Işık Yunus Emre, Görmez Yasin Görmez ve Demirkoparan Ferhan (2017). “OTOMATİK METİN ÖZETLEME İÇİN GENETİK ALGORİTMA TABANLI CÜMLE ÇIKARIMI”. *Yönetim Bilişim Sistemleri Dergisi* 3 (2):62–75.

Kipper Karin, Dang Hoa Trang ve Palmer Martha (2000). “Class-based construction of a verb lexicon”. *AAAI/IAAI* 691:696.

Klema Virginia C. ve Laub Alan J. (1980). “The singular value decomposition: Its computation and some applications”. *IEEE Transactions on Automatic Control* 25 (2):164–176.

Kokash, Natallia (2005). “An introduction to heuristic algorithms”. *Department of Informatics and Telecommunications*, 1–8.

Konstantinides Konstantinos, Natarajan Balas ve Yovanof Gregory S. (1997). “Noise estimation and filtering using block-based singular value decomposition”. *IEEE Transactions on Image Processing* 6 (3):479–483.

Kupiec Julian, Pedersen Jan ve Chen Francine (1995). “A Trainable Document Summarizer”. *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 68–73. SIGIR ’95. Seattle, Washington, USA: ACM.

Landauer Thomas K. ve Dumais Susan T. (1997). “A solution to Plato’s problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge”. *Psychological Review* 104.

Latinier, Benoît (2014). *pyHunspell*. 01 Aralık 2017 tarihinde *https://github.com/blatinier/pyhunspell* adresinden erişildi.

Lee John A ve Verleysen Micheal (2012). "Graph-Based Dimensionality Reduction". *Image Processing and Analysis with Graphs: Theory and Practice*, 570. CRC Press.

Lee Ju-Hong, Park Sun, Ahn Chan-Min ve Kim Daeho (2009). "Automatic generic document summarization based on non-negative matrix factorization". *Information Processing & Management* 45 (1):20–34.

Lin, Chin-Yew (1999). "Training a Selection Function for Extraction". *Proceedings of the Eighth International Conference on Information and Knowledge Management*, 55–62. CIKM '99. New York, NY, USA: ACM.

Lin, Chin-yew (2004). "Rouge: a package for automatic evaluation of summaries". *Text summarization branches out: Proceedings of the ACL-04 workshop.* 25–26.

Lin Chin-Yew ve Hovy Eduard (2003). "Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics". *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, 71–78.

Liu Yanchi, Li Zhongmou, Xiong Hui, Gao Xuedong ve Wu Junjie (2010). "Understanding of Internal Clustering Validation Measures". *2010 IEEE International Conference on Data Mining*, 911–916.

Lloret Elena ve Palomar Manuel (2012). "Text Summarisation in Progress: A Literature Review". *Artificial Intelligence Review* 37 (1):1–41.

Luhn, Hans P. (1958). "The Automatic Creation of Literature Abstracts". *IBM J. Res. Dev.* 2 (2):159–165.

MacQueen, James (1967). "Some methods for classification and analysis of multivariate observations". *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1:281–297.

Mani, Inderjeet (2001). *Automatic Summarization*. John Benjamins Publishing Company.

McCallum Andrew ve Nigam Kamal (1998). "A comparison of event models for naive bayes text classification". *AAAI-98 workshop on learning for text categorization*, 752:41–48.

Meena Yogesh Kumar ve Gopalani Dinesh (2015). "Evolutionary Algorithms for Extractive Automatic Text Summarization". *Procedia Computer Science*, International Conference on Computer, Communication and Convergence (ICCC 2015), 48: 244–249.

Mihalcea Rada F. ve Radev Dragomir R. (2011). *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press.

Mihalcea Rada ve Tarau Paul (2004). "TextRank: Bringing Order into Texts". *Proceedings of the 2004 conference on empirical methods in natural language processing.* 404–411.

MultiLing Community Site: MultiLing 2015. (2015). 11 Kasım 2017 tarihinde *http://multiling.iit.demokritos.gr/pages/view/1516/multiling-2015* adresinden erişildi.

Murray Gabriel, Renals Steve ve Carletta Jean (2005). "Extractive Summarization of Meeting Recordings." *Proceedings of the 9th European Conference on Speech Communication and Technology. 593-596*

Oufaida Houda, Nouali Omar ve Blache Philippe (2014). "Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization". *Journal of King Saud University - Computer and Information Sciences*, 450–61.

Ozsoy, Makbule Gulcin (2011). *Text summarization using Latent Semantic Analysis*. Doktora Tezi, Middle East Technical University, Ankara.

Ozsoy Makbule Gulcin, Alpaslan Ferda Nur ve Cicekli Ilyas (2011). "Text Summarization Using Latent Semantic Analysis". *J. Inf. Sci.* 37 (4):405–417.

Page Lawrance, Brin Sergey, Motwani Rajeev ve Winograd Terry (1998). "The PageRank citation ranking: Bringing order to the Web". *Proceedings of the 7th International World Wide Web Conference*, 161–172.

Pardo Thiago A. S., Rino Lucia H.M. ve Nunes Maria G.V. (2003). "GistSumm: A Summarization Tool Based on a New Extractive Method". *Computational Processing of the Portuguese Language*, 210–18.

Paredaens Jan, Peelman Peter ve Tanca Letizia (1995). "G-Log: a graph-based query language". *IEEE Transactions on Knowledge and Data Engineering* 7 (3):436–53.

Patil Kaustubh ve Brazdil Pavel (2007). "Text summarization: Using centrality in the pathfinder network". *International Journal Computional Science Information System* 2:18–32.

Paul, Mark J. (2007). *The Potential Of Latent Semantic Analysis To Identify Themes In Online Forums*. University of Washington.

Phillips Cynthia ve Swiler Laura P. (1998). "A Graph-based System for Network-vulnerability Analysis". *Proceedings of the 1998 Workshop on New Security Paradigms*, 71–79. NSPW '98. New York, NY, USA: ACM.

Powers, David M W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". *Journal of Machine Learning Technologies* 2 (1):37–63.

Quesada José, Kintsch Walter ve Gomez Emilio (2002). "A Computational Theory of Complex Problem Solving Using Latent Semantic Analysis". *Proceedings of the Cognitive Science Society* 24 (24).

Rehurek, Radim (2013). *Sparsesvd* (sürüm 0.2.2). 15 Ekim 2017 tarihinde *http://pypi.python.org/pypi/sparsesvd* adresinden erişildi.

Ruohonen, Keijo (2013). "Graph Theory by Keijo Ruohonen (English Edition) Student Loose Leaf Edition". Keijo Ruohonen.

Saggion Horacio, Teufel Simone, Radev Dragomir R. ve Lam Wai (2002). "Meta-evaluation of Summaries in a Cross-lingual Environment Using Content-based

Metrics". *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, 1–7.

Savas Berkant ve Eldén Lars (2007). "Handwritten digit classification using higher order singular value decomposition". *Pattern Recognition* 40 (3):993–1003.

Saziyabegum Saiyed ve Sajja Priti S. (2016). "Literature Review on Extractive Text Summarization Approaches". *International Journal of Computer Applications* 156 (12):28–36.

Scharf, Louis L. (1991). "The SVD and reduced rank signal processing". *Signal Processing* 25 (2):113–33.

Shibukawa, Yoshiki (2013). *SnowBall Stemmer*. 19 Kasım 2017 tarihinde *https://github.com/shibukawa/snowball_py* adresinden erişildi.

Shnayderman Aleksandr, Gusev Alexander ve Eskicioglu Ahmet M. (2006). "An SVD-based grayscale image quality measure for local and global assessment". *IEEE Transactions on Image Processing* 15 (2):422–29.

Silva Gabriel, Ferreira Rafael, Lins Rafael D., Cabral Luciano, Oliveira Hilário, Simske Steven J. ve Riss Marcelo (2015). "Automatic Text Document Summarization Based on Machine Learning". *Proceedings of the 2015 ACM Symposium on Document Engineering*, 191–194.

Steinberger Josef ve Jezek Karel (2004). "Using latent semantic analysis in text summarization and summary evaluation". *Proceedings of ISIM'04,* 93–100.

Stenström, Emil (2016). *RippleTagger*. 5 Kasım 2017 tarihinde *https://github.com/EmilStenstrom/rippletagger* adresinden erişildi.

Stewart, G. W. (1993). "On the Early History of the Singular Value Decomposition". *SIAM Review* 35 (4):551–66.

Strait Michal J, Haynes Jacqueline A. ve Foltz Peter W. (2000). "Applications of Latent Semantic Analysis To Lessons Learned Systems". *Technical Report WS-00-03. Intelligent lessons learned systems: Papers from the AAAI Workshop.*

Svore Krysta Marie, Vanderwende Lucy ve Burges Christopher J. C. (2007). "Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources." *EMNLP-CoNLL*, 448–57.

Teufel, Simone (1997). "Sentence extraction as a classification task". *Intelligent Scalable Text Summarization*.

Torres-Moreno, Juan-Manuel (2014). *Automatic Text Summarization*. John Wiley & Sons.

Vector space model - Wikipedia (2013) 22 Kasım 2017 tarihinde *https://en.wikipedia.org/wiki/Vector_space_model* adresinden erişildi.

Wang Xiaolong, Wei Furu, Liu Xiaohua, Zhou Ming ve Zhang Ming (2011). "Topic Sentiment Analysis in Twitter: A Graph-based Hashtag Sentiment Classification Approach". *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 1031–1040.

Weisstein, Eric W. (2013). "Matrix Decomposition". 29 Ekim 2017 tarihinde *http://mathworld.wolfram.com/MatrixDecomposition.html* adresinden erişildi.

Wiemer-Hastings, Peter (2004). "Latent semantic analysis". *Proceedings of the 16th international joint conference on Artificial intelligence*, 1–14.

Wong Kam-fai, Wu Mingli ve Li Wenjie (2008). "Extractive Summarization using Supervised and Semi-supervised learning". *Proceeding COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1,*985-992

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı:	Yunus Emre IŞIK
Uyruğu:	T.C.
Doğum Tarihi ve Yeri:	07.01.1991 KADIKÖY
e-posta:	yeisik@cumhuriyet.edu.tr

EĞİTİM

Derece	**Kurum**	**Mezuniyet Yılı**
Yüksek Lisans	Cumhuriyet Üniversitesi	2018
Lisans	Mehmet Akif Ersoy Üniversitesi	2013

İŞ TECRÜBESİ

Tarih	**Kurum**	**Görev**
01.2015 - Devam Ediyor	Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Fak.	Araştırma Görevlisi
03.2013 – 06.2013	Platin Group A.Ş.	Web Geliştirici
09.2011 - 06.2012	Mehmet Akif Ersoy Üniversitesi	Yarı zamanlı Lab. Sorumlusu

YABANCI DİL BİLGİSİ

Yabancı Dilin Adı YDS (78.75)