T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇİZGE TABANLI METİN ÖZETLEME

CAN YALKIN

YÜKSEK LİSANS TEZİ
MATEMATİK MÜHENDİSLİĞİ ANABİLİM DALI
MATEMATİK MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
YRD. DOÇ. DR. NİLGÜN GÜLER BAYAZIT

İSTANBUL, 2014

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ÇİZGE TABANLI METİN ÖZETLEME

Can YALKIN tarafından hazırlanan tez çalışması 03.07.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Nilgün Güler BAYAZIT

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Nilgün Güler BAYAZIT

Yıldız Teknik Üniversitesi _____

Doç. Dr. Hülya ŞAHİNTÜRK

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Mehmet Fatih AMASYALI

Yıldız Teknik Üniversitesi _____

ÖNSÖZ

Fikirleri ve rehberliğiyle çalışmam boyunca bana danışmanlık yapan Yrd. Doç. Dr. Nilgün Güler BAYAZIT'a ve bilgilerini benimle paylaşmış olan Aysun GÜRAN'a teşekkürlerimi sunarım.

Aynı zamanda tezi hazırlarken desteklerini her zaman hissetiğim annem Güneş YALKIN'a, babam Cengiz YALKIN'a ve Hande ŞENGÜL'e teşekkürlerimi sunarım.

Haziran, 2014

Can YALKIN

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

PR(A)	A sayfasının “PageRank”değeri
J(s1,s2)	s1 ve s2 cümlelerinin Jaccard benzerliği
d_{SL}	Tekli bağ kriterine göre uzaklık
d_{CL}	Tam bağ kriterine göre uzaklık
d_{GA}	Grup ortalamasına göre uzaklık
t_k	Terim sayısı
f_k	t_k terimi geçen cümle sayısı
f_{kl}	İki terimin beraber geçtiği cümle sayısı

KISALTMA LİSTESİ

CAST	Computer-Aided Summarization Tool
DUC	Document Understanding Conference
HBK	Hiyerarşik Birleştirici Kümeleme
HITS	Hyperlink-Induced Topic Search
NGD	Normalised Google Distance
NIST	National Institute of Standards and Technology
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
TREC	Text Retrieval Conference
UMLS	Unified Medical Language Systems

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

ÇİZGE TABANLI METİN ÖZETLEME

Can YALKIN

Matematik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Nilgün Güler BAYAZIT

Günümüzde teknolojinin katkısıyla veri miktarı çok artmıştır dolayısıyla doğru orantılı olarak doküman sayısındaki artış da ivme kazanmıştır. Bu denli bir artış bilgiye ulaşımı zorlaştırır veya bilginin gözden kaçmasına sebep olabilir. Bu tür problemleri çözmek için metin özetleme sistemleri kullanılabilir. Metin özetleme verilen metindeki ana fikri koruyarak onun kısaltılması işlemidir. Genellikle çıkarıma ya da soyutlamaya dayalı olmak üzere iki çeşit sistem üzerinde çalışma yapılır. Soyutlamaya dayalı özetleme derin bir doğal dil işleme gerektirdiğinden yapılan çalışmaların da çoğu çıkarıma dayalı sistemler içindir. Çıkarıma dayalı özetlemede ana metinden cümleler olduğu gibi seçilerek özet çıkartılır. Burada önemli olan en fazla bilgiyi içeren cümleyi özette olması için seçmektir. Çıkarıma dayalı özetlemelerde de anahtar olan nokta cümle seçim aşamasıdır. Cümle seçmek için önerilen birçok yöntem vardır; kelime frekansı kullanan yöntemler, cümle kümeleme, çizge tabanlı puanlama yöntemleri, makine öğrenmesi metotları vb. üstünde çalışılmış yöntemlerin arasındadır.

Çizge metotları metin özetleme sistemlerinde çokça kullanılan bir yöntemdir. Çünkü çizge olarak yapılan temsil verinin daha farklı bir şekilde yorumlanmasına yardımcı olduğundan diğer yöntemler ile kolay bir şekilde ortaya çıkamayacak özellikleri ortaya koyabilir. Bu çalışma kapsamında da çizge tabanlı metin özetleme üstünde araştırma yapılmıştır. Araştırma kapsamında performansı ispatlanmış olan “TextRank” yöntemi kullanılmıştır. Bu yöntem ağ sayfalarının puanlamasının yapıldığı “PageRank”

yönteminden esinlenilerek ortaya konulmuştur. Ağ sayfalarını önem derecesine göre puanlayabilmek için sayfaların birbirlerine vermiş olduğu linkleri kullanarak hesaplama yapar. Metin özetleme sisteminde de bu yöntemi kullanabilmek için cümleler arası ilişki tanımlanması gerekmektedir. Bu çalışma kapsamında 4 farklı ilişkilendirme yönteminin "TextRank" yöntemine olan etkisi araştırılmıştır. Deneysel çalışmalar DUC 2002 ve CAST veri seti kullanarak yapılmıştır. DUC veri setiyle yapılan testlerde en iyi sonucu içerik çakışması, CAST veri setinde ise NGD vermiştir. Bu çalışmaya ilave olarak daha önce yapılmamış bir sistem geliştirilmiştir. Bu sistem hiyerarşik birleştirici kümeleme ve "TextRank" yöntemleri kullanarak elde edilmiştir. Önerilen yeni yöntemde cümleler belli bir kritere göre kümelenmiştir, kümelerden cümle seçebilmek için "TextRank" uygulanmıştır. Yeni yöntemin deneysel çalışmaları, bir önceki çalışmadaki gibi DUC 2002 ve CAST veri setiyle yapılmıştır; böylece "TextRank" ile kıyaslama imkanı elde edilmiştir. Yapılan çalışmalara göre DUC 2002 kullanıldığında önerilen sistemin daha performanslı çalıştığı tespit edilmiştir. CAST veri setinde ise 4 farklı ilişkilendirme yönteminden 2 yöntemi geçtiği, diğer 2 yöntemle de arasındaki farkın az olduğu tespit edilmiştir. Dolayısıyla önerilen yeni yöntem farklı metin türlerine göre de başarılı performans gösterebilmektedir.

Anahtar Kelimeler: metin özetleme, cümle çıkartımı, TextRank, kümeleme

ABSTRACT

GRAPH BASED TEXT SUMMARIZATION

Can YALKIN

Department of Mathematics Engineering

MSc. Thesis

Adviser: Assistant Prof. Nilgün Güler BAYAZIT

Nowadays amount of data has increased very much as the techology improves. So that number of documents has also gained incredible acceleration. The huge amount of documents make harder for users to reach information or cause users to miss some information during search. These kind of problems can be solved by using text summarization systems. Text summarization is the process of extracting a shorter version of the given text by maintaining the main idea. Generally two kind of methods are examined for text summarization systems which are called extractive or abstractive. Since abstractive summarization needs deep knowledge of natural language processing, most of the studies cover extractive methods. In extractive based summarization, sentences are selected as it appears in the given text. The key point is to choose sentences which involve important information for being in the summary. There are a lot of methods proposed for selecting sentences such that methods that use word frequencies, sentence clustering, graph based ranking methods, machine learning methods and so on are some of the techniques that are worked on.

Graph methods are widely used for text summarization systems. Because graph representation enables different interpretation of data which helps to expose some other properties that cannot be easily observed by conventional methods. In this study, research is done on graph based text summarization. In the scope of research TextRank technique whose performance is proved has been utilized. This technique is inspired

from “PageRank” technique which is used for ranking web pages. Method uses links between pages for ranking web pages with respect to importance. In order to use this technique in text summarization system, a relation has to be defined between sentences. In the scope of study effect of four different relation methods on TextRank is analyzed. Experimental studies is done by using DUC 2002 and CAST corpus. According to experimental studies the best result is found with content overlap method by using DUC but for the CAST set NGD gives the best result. In addition to this study, a novel hybrid system which is not tried out before has been developed. The hybrid system has been achieved by combining hierarchical agglomerative clustering and TextRank methods. In the propsed study sentences are clustered according to a criteria, and then TextRank is applied for choosing sentences from clusters. Experimental studies for the novel system is done by using DUC 2002 and CAST corpus like in the previous study so that previous TextRank results can be compared with results of the novel system. According to research; it is observed that hybrid system works with better performance when DUC set is used. CAST set shows that two methods surpass the four different relation methods that are used in TextRank and for the other two method results are close to each other. Therefore proposed novel hybrid method is able to show its advance for different kind of texts.

Keywords: text summarization, sentence extraction, TextRank, clustering

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

İnternet ve bilgisayar kullanımının yaygınlaşması, iş ve akademik dünyada elektronik dokümanların her geçen gün artan sayıda kullanılmasına sebep olmuştur. Elektronik ortam kullanılmasının başlıca sebeplerinden bir kaçı; saklama alanının daha güvenli olması, dokümanların az yer kaplaması ve dokümanlara hızlı ulaşılmasıdır. Fakat bu denli sayıca fazla olan doküman arasından aranan bilginin bulunması kayda değer bir şekilde zaman alır veya önemli bilginin gözden kaçmasına sebep olabilir. Karşılaşılan bu tür problemler otomatik metin özetleme sistemlerinin kullanılmasıyla çözülebilir ve bundan dolayı araştırmacıların dikkatini çekmiştir. Otomatik metin özetleme, verilen bir metnin veya dokümanın bir bilgisayar ile önemli bilgiyi barındıracak şekilde kısaltılması işlemidir. Böylece özetlenmiş metin bir insanın yapmış olduğu özet gibi olamasa dahi, onun sayesinde kullanıcı daha kısa bir sürede dokümanın tamamını okuması gerektiğine karar verebilmiş olur.

Literatürde birçok metin özetleme tipi mevcuttur. Bunlardan en çok üstünde çalışılanlardan ikisi; çıkarıma dayalı ve soyutlamaya dayalı özetleme olarak geçmektedir. Çıkarıma dayalı özetler, metindeki istatistiksel verileri kullanarak; metinden cümleleri, terim ya da kelime gruplarını metinde geçtiği şekilde birleştirerek oluşturulur. Soyutlamaya dayalı özetlemede ise cümlenin anlamı alt/üst kavram ilişkisi ya da eş anlam kurularak başka kelimelerle ifade edilir. Fakat bu tip bir özetleme çok derin bir doğal dil işleme gerektirdiğinden, yapılması daha zordur. Akademik araştırmalar başlarda haber, bilimsel makale vb. olmak üzere tekli doküman özetleme üzerine

yapılmıştır. İlerleyen zamanlarda çoklu doküman özetleme üzerinde çalışmalar yapılmıştır. Bunların dışında içeriğe bağlı özetleme çeşitleri de mevcuttur. Örneğin bildirici özetler, okuyucunun okuduğu doküman hakkında uzunluk, yazı stili gibi bilgi veren özetlerdir. Bilgilendirici özetler dokümanda geçen olayları özetler. Dokümanla ilgili genel bilgi veren özetleme çeşitlerinin aksine sorgu tabanlı özetleme ise sorguya en yakın bilgiyi özet metninde içermesini sağlar [1].

Bir özetleme sisteminin çalışması için özetlenecek dokümanlar sisteme girdi olarak verilir. Ve çıktı olarak da girdide verilmiş olan önemli bilgileri harmanlanarak son özet hali oluşturulur. Fakat önemli bilgiyi bulmak dokümanı anlamak ile doğru orantılıdır. Birçok sistem metnin anlamıyla ilgili olan kısımlardan kaçınır ve çoğunlukla cümle seçimine dayalı bir özet oluşturulur. Cümle seçiminde karşılaşılan en büyük problem hangi cümlenin önemli olduğuna nasıl karar verileceğidir [1].

Çıkarıma dayalı özetleme, cümle bazında önemli içeriği bulmaya dayanır. Frekans ve benzerlik gibi özellikler kullanılarak önemli cümlelere karar verilir. İlk çalışmalar 1950 yıllarında Luhn tarafından yapılmıştır [2]. Bu çalışmalarda ağırlıklı olarak kelime frekansları kullanılmıştır. Araştırmaların sonucu göstermiştir ki; herhangi bir dokümanda en yaygın kelimeler içeriği açıklamaz ve de zamir edat gibi kelimeler içerikle ilgili bilgi vermezler. Araştırmalar, makine öğrenmesi teknikleri, cümle pozisyonları, cümle ve başlıkta geçen kelimeler de dikkate alınarak geliştirilmiştir [1].

1.2 Tezin Amacı

Bu çalışmada verilen metni özetlemek için çıkarıma dayalı bir yöntem kullanılmıştır. Metnin özetinin bulunabilmesi için cümlelerin çizge temsilinden yararlanılmıştır. Literatürde çizge metotlarını kullanan birçok özetleme sistemi bulunmaktadır [3], [4], [5]. Çizge metotları arasından en başarılı olanlardan biri Google'ın kullanmış olduğu ağ sayfa puanlama yöntemidir. Bu puanlama yönteminde sayfaların birbirlerine verdikleri referanslar temel alınır ve bu doğrultuda sayfalar önem derecelerine göre puanlanır. Bu yöntemi metin özetleme sisteminde uygulamak için cümleler arası ilişki tanımlanması gerekmektedir. Cümleler arası ilişki birçok yöntem ile sağlanabilir. Bu çalışma

kapsamında değişik ilişkilendirme yöntemleri kullanılarak uzun ve kısa dokümanlarda sayfa puanlama yönteminin metin özetine nasıl katkı sağladığı araştırılmıştır. Bunun yanında hiyerarşik birleştirici kümeleme yöntemi ile çizge metodu olan "TextRank" birleştirilerek yeni bir sistem elde edilmiştir. Bu çalışmada bu sistemin hangi durumlarda başarılı olduğu ve bir önceki çalışmadaki değişik ilişkilendirme metotlarıyla arasındaki başarı farkı gözlemlenmiştir.

1.3 Hipotez

Çalışmanın ölçümlerini yapabilmek için standartlaşmış olan DUC 2002 referans dokümanları ve CAST veri seti kullanılmıştır [6], [7]. DUC sonuçları analiz edildiğinde; "TextRank" için en başarılı yöntemin içerik çakışması fonksiyonu ile kullanıldığında ortaya çıktığı gözlemlenmiştir. Bununla birlikte DUC veri setinde Jaccard yönteminin NGD'ye göre daha başarılı sonuçlar verdiği gözlemlenmiştir. CAST veri seti kullanılarak yapılan ölçümlerde ise NGD benzerlik yönteminin orijinal "TextRank" yönteminin kullanmış olduğu içerik çakışma metodundan daha iyi performans gösterdiği sonucuna varılmıştır. Bu durum da "TextRank" metodunda kullanılacak ilişkilendirme yöntemlerinin veri setlerinin yapısına ve özetin hazırlanma şekline göre değişik başarılar gösterdiğini ortaya çıkartmıştır. Dolayısıyla dokümanların özeti çıkartılacağı zaman özetlenme biçimlerine göre ilişkilendirme metodları seçilmelidir.

Önerilen sistem ile ilgili yapılan testler sonucunda; DUC verisi kullanıldığı zaman başarı oranının sadece "TextRank" kullanılan çalışmadaki 4 ilişkilendirme yöntemine göre daha iyileştiği görülmüştür. CAST veri seti kullanıldığında önerilen sistem bu 4 ölçütün ikisinde başarılı olmuştur. Fakat diğer iki yöntemin sonuçlarına yakın olduğu gözlemlenmiştir.

BÖLÜM 2

LİTERATÜR TARAMASI

Otomatik metin özetleme geçtiğimiz yıllarda birçok değişik metot ile üstünde bir hayli çalışılan bir alan olmuştur. Çalışmaların çoğu yoruma dayalı özetleme yerine çıkarıma dayalı özetlemeler üzerinde yapılmıştır. Evrimsel algoritmalar ve kümeleme yöntemleri, metin özetleme sistemlerinde son yıllarda çalışılan diğer algoritmalardandır. Bu algoritmalar daha çok kümeleme yapılırken fonksiyon optimizasyonu aşamasında veya yöntemde bulunması gereken bazı kat sayı hesaplamalarında kullanılmaktadır. Kümeleme kullanılan yöntemlerde benzer olan cümleler bir kümede toplanılır ve özetlere cümle seçilirken genellikle merkezilik özelliğinden yararlanılır [8], [9]. Buna ilaveten metin özetleme alanında başarılı sonuçlarından dolayı çizge teorisi de yaygın bir şekilde kullanılmaktadır. Metin özetleme çalışmalarında, çalışmanın başarısını ölçmek amacıyla genellikle "Metin Anlama Konferansı" (Document Undertsanding Conference-DUC) ve CAST veri setleri kullanılmaktadır [6], [7].

2.1 Çıkarıma Dayalı Özetleme

Otomatik metin özetlemede en çok uygulanan yöntem çıkarıma dayalı metotlardır. Bu amaca hizmet eden bugüne kadar geliştirilmiş birçok yöntem vardır. Bu yöntemlerin amacı en dikkat çeken cümleleri bulmaktır. Birçok çalışma özette bulunacak en belirgin cümleleri belirlemek için puanlama yöntemi üzerine yapılmıştır [3], [10], [11]. Örneğin You Ouyang vd. çalışmasında yeni bir cümle seçim yöntemi sunmuştur ve kademeli cümle seçim stratejisi izlemiştir [12]. Yöntemin ortaya çıkışında cümlenin farklı kavramlar içermesi ve böylece dikkat çekici cümlelerin özete seçilmesi düşüncesi vardır.

Bunun için çalışmada, o cümlenin seçilmesi gerektiğine karar vermek için bahsedilmemiş kavramların önemi incelenir. Buradaki ana fikir genel kavramların yanında onların içerdiği destekleyici kavramların da beraberinde seçilmesi gerektiğidir. Denemeler DUC veri setiyle yapılmış ve ortaya çıkan sonuca göre önerilen yöntemdeki dikkat çekme ve kavram kapsamının daha iyi olduğu ortaya konmuştur.

Aditi Sharan vd. makalesinde cümle seçimini gerçekleştirmek için Luhn'un anahtar kelime kümeleme tekniğini kullanmıştır [10]. Algoritma ilk etapta önemli kelimeleri seçer. Daha sonra üç fonksiyonun birleşiminden elde edilen skor değerlerine göre özette olacak cümleler seçilir.

Çıkarıma dayalı özetlemelerde cümle kümeleme yöntemi sıkça tercih edilebilen bir yöntemdir. Bunun başlıca sebebi benzer cümleleri bir arada toplayıp özette farklı alanlardan cümlelerin bulunmasını sağlamaktır. Zhang Pei Ying vd. çalışmasında cümle kümelemeye dayalı özetleme yöntemi önermiştir [8]. Yöntemin çalışması üç aşamada gerçekleştirilir. İlk olarak cümleler anlamsal uzaklıklarına göre kümelenir. Daha sonra her küme için birkaç özelliğin kullanılmasıyla toplam benzerlik hesaplaması yapılır. Son aşamada bazı çıkartım kurallarına bağlı olarak kümelerden konuyla ilgili cümleler seçilir. Deneysel çalışmalar DUC 2003 ve ölçümler de ROUGE kullanılarak yapılmıştır. DUC 2003 üzerinde yapılan sonuçlarda önerilen yöntemin diğer özetleme yöntemlerine göre daha performanslı çalıştığı ortaya konmuştur.

Özette, dokümandaki cümlelerin direkt olarak kullanılması yerine çoğu zaman insanlar, seçilmiş cümleleri de kısaltarak anlam bütünlüğünü bozmayacak şekilde özeti oluştururlar. Jin makalesinde yaptığı çalışmasında önerilen sistem, özet için çıkartılmış cümlelerden konu dışı olan kelime gruplarını siler [13]. Kelime gruplarının konu dışında olduğuna karar verebilmek için birçok dilbilimsel, kapsam bilgisi ve istatistik hesaplamalar içeren kaynaklar kullanılır. Metot ilk başta sözdizimsel çözümleme yapar ve çözümleme ağacındaki her bir düğüm ile ilgili bilgiyi, konuyla ilgili olduğuna karar vermek adına saklar. Daha sonra cümleden hangi kısımların atılmaması gerektiğini belirlemek için gramer analizi yapılır. Üçüncü aşamada önemli kelime gruplarına karar verebilmek için kelimeler arasındaki sözlüksel ilişkiler çıkartılır. Buna bağlı olarak da kelimelerin önem dereceleri hesaplanır. Tüm bu analizler göz önünde bulundurularak

hangi kelime gruplarının atılacağına karar verilir. Çalışmalar 100 cümle üzerinde yapılmış ve ortalama %81 başarı sağlandığı saptanmıştır.

2.2 Metin Özetlemede Evrimsel Algoritmalar

Evrimsel algoritma kullanılarak yapılan metin özetlemeler çoğunlukla kümeleme yapmak için kullanılan fonksiyonun optimizasyonu için veya cümle puanlaması için kullanılan fonksiyonun kat sayılarının belirlenmesi amacıyla kullanılmıştır.

Aliguliev vd. yaptığı çalışmada cümle çıkarımına dayalı bir sistem önermiştir [9]. Metinden en önemli cümleleri seçerek özeti oluşturur. Önerilen yöntemde normalize edilmiş Google uzaklığı kullanılarak cümleler kümelere ayrılır. Daha sonra belirli yöntemlere göre cümleler kümelerden seçilir. Kümeleme yapılabilmesi için iki kriter fonksiyonu belirlenir. Kriter fonksiyonlarından biri kümeler arasındaki benzerliğin azaltılması için diğeri ise aynı kümedeki cümlelerin birbirine yakın olmasını sağlamak için oluşturulmuştur. Bu iki fonksiyonun kombinasyonu kullanılarak amaç fonksiyonu belirlenmiş olur. Bu fonksiyonun optimizasyonu diferansiyel evrim algoritması ile bulunur. Dolayısıyla cümleler kümelere ayrılmış olur. Kümelerden cümle seçimi yapılabilmesi için kümede en çok kelimeyi bulunduran cümle merkez kabul edilir ve o cümle seçilir.

Binwahlan vd. çalışmasında cümle çıkarımına dayalı bir özetleme sistemi önerilmiştir [11]. Bu sistem önemli cümleleri puanlama yöntemi ile tayin eder. Cümleleri puanlama yönteminde kullanılan özelliklerin hepsinin aynı önem derecesinde olması çıkan özetin kalitesinin düşük olmasına sebep olduğu iddia edilmiştir. Fakat bu çalışmada amaç parçacık sürü optimizasyon yöntemi kullanılarak hangi özelliğin daha baskın olduğunu ortaya çıkartıp, cümle puanlamada bu özelliklerin etkisini arttırarak daha iyi bir puanlama elde etmektir. Hangi özelliğin daha etkin olduğunu bulmak için eğitim seti kullanılarak metin özelliklerinin kat sayısı parçacık sürü optimizasyonu algoritması ile hesaplanmıştır. Daha sonra bulunan katsayılar ile veri kümesindeki cümleler özetlenmiştir. Analiz MS Word özetleri ile kıyaslanarak yapılmış ve ondan daha iyi olduğu gözlenmiştir.

Diğer bir çalışmada ise Nguyen Quang vd. otomatik metin özetleme yapabilmek için genetik programlama kullanmıştır [14]. Genetik programlama kullanılmasındaki amaç cümleyi puanlayan fonksiyonu oluşturmaktır. Böylece oluşturulan fonksiyon en önemli cümleleri seçer. Fonksiyonu oluşturmak için genetik programlama dört birim elemanı ve dört farklı operatör kullanmıştır. Birim elemanları paragrafın sırası, cümlenin sırası, cümle uzunluğu, içerik-kelime frekansıdır. Bu fonksiyonların hangi kombinasyon ile kullanılacağı genetik programlama sayesinde bulunur. Ve cümle özetlemek için oluşturulan fonksiyon cümleyi puanlamak amacıyla kullanılır. Denemeler Vietnamca dokümanlar üzerinde yapılmıştır. Diğer istatistiksel yöntemlere göre daha etkin olduğu gözlemlenmiştir.

Abuobieda vd. çalışmasında tekli doküman özetleme için kullanılan [9] çalışmasındaki metodu referans alarak geliştirmişlerdir [15]. Önerilen bu yeni yöntemde üç ana değişiklik yapılmıştır. Bunlardan ilki; yöntemde, daha önceden normalize edilmiş Google uzaklığı kullanılmak yerine Jaccard benzerlik ölçütü kullanılmıştır. İkinci olarak oluşturulan kümelerden cümleleri seçebilmek adına özellik tabanlı bir yaklaşım eklenmiştir. Son değişiklik olarak da cümle kümeleme sürecinde kullanılmak üzere gerçek değerli sayıdan tam değerli sayı dönüşümü yapmak için bir kipleyici önerilmiştir. Çalışma DUC 2002 veri seti kullanılarak ölçülmüştür. Ve çalışmanın sonucunda NGD benzerliğinin bu yöntem için uygun olmadığı sonucuna varılmıştır.

2.3 Metin Özetlemede Çizge Tabanlı Yaklaşımlar

Metin özetleme sistemlerinde çizge tabanlı yaklaşım kullanılması verinin daha esnek bir şekilde ifade edilmesini sağlar. Genellikle cümleler köşe noktaları, aralarındaki bağlar cümlelerin birbirleri ile arasındaki yakınlığı ifade eder. Yakınlık ölçütü anlamsal ya da kelime frekanslarına bağlı yakınlık ölçütleri olabilir. Plaza vd. çalışmasında medikal alanda bir özetleme sistemi yapmıştır [16]. Bu alanda çok fazla kaynak olduğundan aranılan konuya ulaşımın geliştirilmesi hedeflenmektedir. Bunu sağlayabilmek için kavramlar ve UMLS sistemi kullanılarak mantıksal bir çizge elde edilmiştir. Kümeleme yöntemi kullanarak da dokümanda geçen farklı başlıklar ortaya çıkartılmıştır. Bu çalışma alana özel bilginin kullanılmasının özete ne kadar yararlı olduğunu ve çıkan özetin ne kadar kaliteli olduğunu göstermek amacıyla yapılmıştır. Sistem özeti yedi aşamada

tamamlar. İlk olarak dokümandan referanslar bilgilendirmeler gibi ilgisiz kısımlar, kısaltmalar çıkartılır sonrasında gereksiz kelimeler de özete katkısı bulunmayacağından çıkartılır. İkinci aşamada UMLS kullanılarak metindeki kavramlar bulunur. Kavramların hiyerarşik yapısı kullanılarak cümlenin çizgesi çıkartılır. Oluşturulan tüm çizgeler tek bir doküman çizgesi oluşturması amacıyla birleştirilir ve farklı kavramlar çıkartılması amacıyla kümeleme işlemi yapılır. Altıncı aşamada cümleler uygun kümelere mantıksal yakınlığa göre atanır. En son aşamada ise cümle seçim işlemi gerçekleştirilir. Çalışmanın sonucunda özetlemede alan bilgisinin kullanılmasının özetleme işlemi için çok kullanışlı olduğu ortaya çıkmıştır. Biyomedikal alanında bu çalışmaya benzer kümeleme ve çizge kullanılarak farklı bir çalışma yapılmıştır [17]. Bu çalışmada da çizgeye uygulanan kümeleme tekniği kullanılarak önemli cümleler çizge oluşturulması yöntemiyle seçilmiştir. Bu çalışmada farklı olarak kümeleme için merkezilik yerine kavram ilişkilerinin sürekliliği temel alınmıştır. Denemeler biyomedikal makaleleri üzerinde ve sonuç ölçümleri ROUGE kullanılarak yapılmıştır.

Mihalcea vd. çalışmasında ağ arama teknolojisinde kullanılan ağ sayfası puanlama yönteminden esinlenmiştir [3]. Ağ sayfasını puanlama yöntemi aslında çizge tabanlı, köşe noktasının önemini belirleyen bir puanlama algoritmasıdır. Algoritma çalışması esnasında sadece yerel bilgiyi değil tüm çizgeyi değerlendirir. Bu çalışma iki alt çalışmadan oluşmuştur; bunlardan ilki anahtar kelime seçimi, diğeri ise önemli cümle çıkarımıdır. Bu yöntemin cümle çıkarımında kullanılabilmesi için ilk olarak dokümanın çizge şeklinde ifade edilmesi gerekir. Cümlelerin puanlanması söz konusu olduğundan cümleler düğüm noktaları olarak belirlenmiştir. Cümleler arasındaki bağların katsayısı ise birbirleri ile olan yakınlıklarına göre belirlenir. Çalışmada yakınlık içerik örtüşmesi yöntemiyle belirlenmiştir. Önerilen yeni yöntem DUC 2002 kullanılarak sonuçları en iyi 5 sistem ile kıyaslanmıştır ve en önemli cümleleri bulabildiği gözlemlenmiştir.

Çalışma [3], [18] çalışmasının bir devamı niteliğinde yapılmıştır. Çalışma [3] de ek olarak çoklu doküman özetleme ve de “Google” sayfa işaretleme algoritmasının yanında HITS algoritması da eklenerek iki çizge tabanlı yöntemin analizi gerçekleştirilmiştir. Bunlara ilave olarak çizge oluşturulurken düğümler arasındaki bağlantılar yönsüz, ileri ve geri yönlü olacak şekilde üç farklı bağlantı şekli için de analizi gerçekleştirilmiştir. Çalışmanın sonucuna göre çizge tabanlı algoritmalar cümle çıkartımı yöntemli özetleme metotları

için başarılı sonuçlar elde ettiği görüşmüştür. Bunun yanında Portekizce doküman seti üzerinde yapılan deneme ile de dilden bağımsız bir şekilde çalışabileceği gözlemlenmiştir.

Chatterjee vd. çalışmasında cümle çıkarımı bazlı tekli doküman özetleme sistemi önermiştir [4]. Sistemin çalışması dokümanın yönlendirilmiş çevrimsiz çizge oluşturulması ile yapılır. Çizge oluşturmak için cümlelerin benzerlikleri, terim ve ters cümle frekansı kullanılarak bulunan ağırlık katsayıları, kullanılarak bulunur. Doküman özeti; başlık ilişkisi, uyum ve okunabilirlik olmak üzere üç farklı kritere göre yapılır. Bu kriterler belli katsayılar dahilinde matematiksel uyumluluk fonksiyonu ile ifade edilir. Genetik algoritma kullanılarak bu fonksiyonu maksimum yapacak cümlelerin seçilmesi sağlanır. Deneysel çalışmalar DUC 2002 dokümanları ile yapılmıştır ve sonuçlar duyarlılık ölçütü üzerinden değerlendirilmiştir. Bu metodun da birçok metot kadar etkin olduğu çalışma sonucuna göre ortaya konmuştur. Qazvinian ve diğerlerinin çalışması da çalışma [4] e benzer bir şekilde genetik algoritma kullanarak en önemli cümleleri seçer [5]. Bu çalışmada farklı olarak uyumluluk faktörü ve en uzun yol hesaplaması daha farklı bir şekilde gerçekleştirilmiştir. Çalışma [19] kümeleme ve "TextRank" kullanarak toplantı özeti çıkartan bir sistem önerir. Normal kullanılan yöntemler toplantılar için de uygulanabilir fakat çok fazla miktardaki gürültü ve gereksiz kelime gruplarıyla uğraşmak performansın artmasını sağlar. Çalışmada "TextRank"'in geliştirilmiş bir algoritması kullanılır. "ClusterRank" yazıyı bölümlere ayırır daha sonra bu kümeler, yazının küme çizgesini oluşturmak adına kullanılır. Daha sonra kümedeki merkez cümle kullanılarak cümleler puanlanır. Ve puanı yüksek olan özette olacak biçimde seçilir. Değerlendirmeler AMI toplantı kitaplığı üzerinden yapılır. Önerilen yöntemin normal "TextRank"'e göre kayda değer bir şekilde ilerleme kaydettiği gözlemlenmiştir.

2.4 Tez Kapsamında Kullanılan Veri Setleri

Bugüne kadar metin özetleme sistemleri ile ilgili birçok bilimsel çalışma yapılmıştır. Bu yüzden yapılan çalışmaların doğru bir şekilde ölçülmesi ve değerlendirilmesi gerekmektedir. Bunun yanında yapılan çalışmaların bir önceki çalışmalara göre ne kadar gelişme sağlandığının takibi de çok önem arz etmektedir. Bunu sağlayabilmek adına da standart olarak kullanılan veri setlerinin kullanılması gerekmektedir. Bu konuda yapılan

çalışmaların büyük bir çoğunluğu İngilizce dili üzerine olduğundan metin özetleme ile ilgili oluşturulmuş geniş çaplı veri setleri mevcuttur. Bu çalışma kapsamında en yaygın olan DUC ve CAST olmak üzere 2 çeşit veri seti kullanılmaktadır. Bu verilerle ilgili detaylar aşağıda verilmektedir.

2.4.1 DUC Veri Seti

Bu veri seti 2002 yılında düzenlenen "Metin Anlama Konferansı" (Document Undertsanding Conference - DUC 2002) kapsamında oluşturulmuştur ve NIST (National Institute of Standards and Technology) tarafından desteklenmiştir [7], [20]. Doküman kümeleri TREC verisi kullanılarak oluşturulmuştur. Her kümede ortalaması 10 olacak şekilde 5-15 arasında doküman bulunur. Dokümanlarda en az 10 cümle bulunur ve maksimum sınırı yoktur. Her doküman için ortalama kelime sayısı 100 olan tekli doküman özeti bulunur. Her veri kümesi için 4 adet çoklu doküman özeti vardır. Her doküman kümesi için 4 tür mevcuttur. Bu tipler doğal afetler, biyografi, tek olay veya çoklu ayrık olaylar içermektedir. Veri seti farklı konuları içeren 567 adet haber dokümanını ve bu dokümanlara ait olan özet dokümanlarını kapsamaktadır. Fakat bu dokümanlar içerisinden 34 tane aynı olan doküman bulunduğundan çalışma boyunca 533 doküman kullanılmıştır [20]. Çıkarılan özetler yoruma dayalı özetlerdir.

Tez çalışması kapsamında her bir haber metni her satırda bir cümle olacak şekilde parçalara ayrılmıştır. Çıkarılan ideal özet gruplarına da aynı işlem uygulanmıştır. Bu şartlar altında elde edilen veri setine ait olan istatistikler aşağıdaki çizelgeden görülebilir:

Çizelge 2.1 DUC Veri Seti İstatistikleri

DUC Veri Seti İstatistikleri	
Veri Setindeki Toplam Doküman Sayısı	533
Veri Setindeki Toplam Cümle Sayısı	16432
Ortalama Cümle Sayısı : (Toplam Cümle Sayısı/Toplam Doküman Sayısı)	28.98
Veri Seti İçindeki Dokümanlardan en az Cümleye Sahip olan Dokümanın Cümle Sayısı	10
Veri Seti İçindeki Dokümanlardan en fazla Cümleye Sahip olan Dokümanın Cümle Sayısı	176

2.4.2 CAST

CAST veri setini oluşturan kitaplık Reuters haber dokümanlarından oluşmaktadır. Bunun yanında İngiliz ulusal kitaplığından alınan bazı ünlü bilimsel metinleri de içerir [6]. CAST veri seti diğer kütüphanelerden farklı olarak açıklamalı bir şekilde oluşturulmuştur. Diğer bir deyişle cümlelerin önemiyle ilgili bilgiler verilmiştir. Dokümanlarda önemli cümlelerin yanı sıra cümlelerden silinebilecek kısımlar da işaretlenmiştir. DUC 2002 veri setinden farklı olarak özetler cümle çıkarımına dayalı olan özetlerdir. Başka bir deyişle doküman özetleri haber metinlerinden direk alınan cümleler ile oluşturulmuştur. Bu kitaplıkta özet, dokümanlar ile aynı dosya içinde bulunmaktadır ve orijinal haber dokümanlarında "önemli", "orta seviyede önemli" ve "önemsiz" şeklinde etiketlendirilmiştir. Etiketleme işlemi bazı dokümanlar için birden fazla kişi tarafından yapılmıştır. Bu sayede farklı etiketlemelerin birbirleriyle ne kadar uyuştuğu gözlemlenebilmiştir. Veri ile ilgili istatistikler Çizelge 2.2'de gösterilmiştir.

Çizelge 2.2 CAST Veri Seti ile ilgili istatistikler

CAST Veri Setine ait olan istatistikler	
Veri Setindeki Toplam Doküman Sayısı	100
Veri Setindeki Toplam Cümle Sayısı	2154
Ortalama Cümle Sayısı : (Toplam Cümle Sayısı/Toplam Doküman Sayısı)	23,4
Veri Seti İçindeki Dokümanlardan en az Cümleye Sahip olan Dokümanın Cümle Sayısı	7
Veri Seti İçindeki Dokümanlardan en fazla Cümleye Sahip olan Dokümanın Cümle Sayısı	47

2.4.3 Metin Özetlemede Değerlendirme Ölçütleri

Otomatik özetleme sistemlerinin kullanılan veri setleri üzerindeki başarımlarının değerlendirilmesi amacıyla kullanılan çok sayıda ölçüm yöntemleri bulunmaktadır. Bu çalışma kapsamında değerlendirmeler keskinlik, anma, f-ölçüm değeri ve ROUGE kullanılarak yapılmıştır.

2.4.3.1 Keskinlik, Anma, F-Ölçüm Değeri

Bu üç ölçüm değeri sistemin performansını ölçmek için kullanılır. Keskinlik değeri sistemin ve referans özetin ortak cümle sayısının, sistemin bulduğu özetteki cümle sayısına oranıdır. Anma değeri ise referans ve özetteki ortak cümle sayısının, referans özetteki cümle sayısına oranı olarak tanımlanır. F ölçüm değeri keskinlik ve anma değerlerinin kombinasyonudur. Denklem 2.1'de keskinlik değeri (K), anma değeri (A) ve F-ölçüm değeri (F) ile gösterilmektedir. Sistem özetleri (S) ve referans özetler ise (R) ile gösterilir.

$$K = \frac{|S \cap R|}{|S|} \quad A = \frac{|S \cap R|}{|R|} \quad F = \frac{2KA}{K + A} \tag{2.1}$$

2.4.3.2 ROUGE

ROUGE (Recall - Oriented Understudy for Gisting Evaluation) Chin-Yew Lin tarafından özetleri değerlendirmek için oluşturulmuş bir yazılım paketidir [21]. Bu ölçüm çakışan n-gram, kelime sıralarını sayarak sistemin bulduğu özet ile referans özeti karşılaştırır. Yazılım paketi Perl dili ile tasarlanmıştır ve ilk olarak doküman anlama konferansında (DUC) kullanılmıştır [7]. Günümüzde metin özetleme alanında kullanılan en güncel özet değerlendirme paketidir. ROUGE paketi toplam beş farklı ölçüm değerine sahiptir: ROUGE-N, ROUGE-L, ROUGE-S, ROUGE-W, ROUGE-SU. ROUGE-N denklem 2.2'de verilmiştir.

$$ROUGE - N = \frac{\sum_{S \epsilon \{\text{İnsan Özetler}\}} \sum_{gram_n \epsilon\, S} hesapla_{\text{çakışan}}(gram_n)}{\sum_{S \epsilon \{\text{İnsan Özetler}\}} \sum_{gram_n \epsilon\, S} hesapla(gram_n)} \tag{2.2}$$

Denklemde bulunan n, n-gramın uzunluğunu belirtir, $hesapla_{\text{çakışan}}(gram_n)$ referans ve otomatik sistem özetlerinin ortaklaşa sahip olduğu maksimum n-gram sayısıdır ve $hesapla(gram_n)$ ideal özetteki toplam n-gram sayısıdır. Bu ölçüm yönteminin anma tabanlı olmasının sebebi; paydada, referans özetteki n-gramların toplamını

kullanmasıdır. Buna ek olarak değerlendirmeye daha fazla referans doküman eklendiğinde paydadaki değer artar. Referans özetlerin artması bir yerde daha iyi özetlerin olabileceğini gösterir [21]. Paketin içinde ROUGE-N yanında ROUGE-L, ROUGE-S gibi farklı değerlendirme yöntemleri de vardır [21].

BÖLÜM 3

BENZERLİK YÖNTEMLERİNİN TEXTRANK ALGORİTMASINA ETKİSİ

3.1 TextRank Algoritması

Bu algoritma "Google" arama motorunun da kullanmış olduğu ağ sayfası puanlama "PageRank" yönteminden esinlenerek ortaya konulmuş bir yöntemdir. Yöntem Brin ve Page tarafından 1998 yılında bulunmuştur [22] ve en çok alıntı analizi, sosyal ağlar ve ağ ortamındaki bağlantıları analiz etmek amacıyla kullanılır. Çalışmasındaki temel prensip sayfaya yapılan alıntıların veya o sayfaya gelen bağların sayılmasıdır. Böylece bu istatistik sayesinde sayfanın önem değeri yaklaşık olarak hesaplanmış olur. Algoritmanın en önemli yanlarından biri; sayfanın önemini içeriği üzerinden puanlamak yerine, ağ mimarisinin oluşturmuş olduğu bütünsellikten yararlanmasıdır. Diğer bir deyişle ağlar arasındaki bağların rağbetinden yararlanılır; yani bir sayfanın önemi aldığı bağ sayısı ile ölçülebilir. Bunun yanında o sayfanın bağ alması tek ölçüt değildir, bağ aldığı sayfaların da puanının yüksek olması ona önem katan başka bir ölçüt olur. Dolayısıyla bir sayfa sadece tek başına değerlendirilmemiş olur ve diğer sayfaların da o sayfaya etkisi hesaba katılır. PageRank hesaplaması için kullanılan denklem (3.1)'deki gibidir.

$$PR(A) = (1-d) + d\left(\frac{PR(T1)}{C(T1)} + \cdots + \frac{PR(Tn)}{C(Tn)}\right) \tag{3.1}$$

(3.1) denkleminde “A” PageRank’ı hesaplanacak sayfayı, T1…Tn ise A sayfasını işaret eden sayfaları ifade etmektedir. “d” parametresi bir kullanıcının bir sayfadan diğerine geçmesi ihtimalidir, 0 ile 1 arasında belirlenen bir değerdir ve genellikle 0.85 olarak kullanılır. C(A) değeri A sayfasından çıkan bağların sayısıdır [22]. Diğer bir deyişle bir sayfanın puanlanmasının yapılması için o sayfaya gelen ve giden bağların özyinelemeli olarak olasılık dağılımından yararlanılır. Örneğin A ve B iki sayfa birbirlerine bağ vermiş olsunlar. Ve d parametresi önceden belirtildiği gibi 0.85 olarak kullanılırsa; PR(A)=PR(B)=0.15+0.85*1=1 olarak hesaplanır. Fakat bu hesapta PR(B)=1 olarak alınmıştır. Eğer ilk durumda PR(B)=0 olarak alınırsa PR(A)=0.15+0.85*0=0.15 ve PR(B)=0.15+0.85*0.15=0.2775 bulunur. Ve buradan çıkan değerler tekrar kullanılarak hesaplanırsa bir sonraki adımda PR(A)=0.15+0.85*0.2775=0.385 çıkar. PR(B) ise buradan hesaplanan PR(A) kullanılarak tekrar hesaplanabilir; PR(B)=0.15+0.85*0.385=0.4779.

Çizge tabanlı önceliklendirme yöntemleri, çizgedeki düğüm noktalarının önemini hesaplamak için kullanılırlar. TextRank algoritması da PageRank’te olduğu gibi düğümlerden çıkan ve düğümlere gelen bağlardan yararlanır. Fakat bu yöntemde çizgeler metin veya metin gruplarından oluştuğu için, düğümler arasındaki bağın kuvvetini gösteren ağırlıklandırma eklenmiştir. Dolayısıyla eklenen ağırlıklandırma değerleri kullanılarak, belli kriterdeki düğümlerin öncelikleri arttırılabilir. Çizge tabanlı algoritmalar bu yöntemi geleneksel olarak yönlendirilmiş çizgeler için kullanır, fakat bu yöntemi yönlendirilmemiş çizgeler için de kullanmak mümkündür. Buna ek olarak yönlendirilmemiş çizge kullanımının daha iyi yakınsama eğilimi olduğu belirtilmiştir [3]. TextRank için kullanılan denklem (3.2)’deki gibi verilmiştir [3].

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \qquad (3.2)$$

Metin özetleme kapsamında TextRank algoritmasından yararlanabilmek için, metnin çizge şeklinde temsil edilmesi gerekir. Bu temsili yapmak için belirli metin gruplarının düğüm noktaları ve aralarındaki bağın kuvvetini gösteren ağırlık değerlerinin verilmesi gerekir. Cümle çıkartımı metodu kullanarak özetleme yapan sistemlerde özete hangi cümlenin gireceğine karar verilebilmesi için cümle puanlama yöntemlerinin kritik bir önemi vardır. Bu yüzden özetleme yapmak için düğüm noktaları cümleler olarak alınır ve

tüm cümlelerin birbiriyle bağı olduğu esasına göre çizge oluşturulur. Cümleler arasındaki ağırlık değerleri bir benzerlik yöntemiyle hesaplanabilir. Tüm cümleler birbirlerine yönsüz bir şekilde tam bağlı olduğu için algoritmayı esas yönlendiren özellik cümleler arasındaki ağırlık değerleridir. Dolayısıyla bu ağırlık değerlerinin hesaplanması özetin ne kadar iyi yapılacağına direkt olarak etki eder. Çalışmanın bu kısmında kosinüs, Jaccard ve normalize edilmiş Google uzaklığı (NGD) benzerlik yöntemlerinin TextRank algoritmasına, kısa ve uzun dokümanlar olmak üzere iki ayrı kategoride, nasıl etki ettiği gözlemlenmiştir.

3.2 Cümle Benzerlik Yöntemleri

3.2.1 Kosinüs Benzerlik Yöntemi

Kosinüs benzerliği; iki vektörün nokta çarpımı ile elde edilir ve değeri 0 ile 1 arasındadır. Değerin bire yakın olması iki vektör arasındaki açının küçük olduğunu gösterir. İki vektör arasındaki açının küçük olması vektörlerin birbirine benzer diye yorumlanması sonucunu doğurur. İki cümlenin kosinüs benzerliği aşağıdaki denklemdeki gibi hesaplanır.

$$
\begin{aligned}
&w_b = \langle w_1, \dots, w_n \rangle \\
&s_1 = \langle c(w_1, s_1), \dots c(w_n, s_1) \rangle \\
&s_2 = \langle c(w_1, s_2), \dots c(w_n, s_2) \rangle \\
&Cos\ \theta = \frac{s_1 \,.\, s_2}{\|s_1\| \|s_2\|}
\end{aligned}
\tag{3.3}
$$

İki cümlenin kosinüs benzerliğini bulabilmek için ilk olarak iki cümleden alınan kelimelerle temel vektör (w_b) oluşturulur. Her iki cümle için de bu temel vektör baz alınır. "c" fonksiyonu verilen kelimenin o cümlede kaç kere geçtiğini bulur. Sırasıyla temel vektördeki kelimelerin her iki cümlede de geçme sayısına göre cümle vektörleri oluşturulur. Ve kosinüs benzerliği bu iki vektörün nokta çarpımı ile hesaplanmış olur.

3.2.2 Jaccard Benzerlik Yöntemi

Jaccard benzerliği iki öğenin kesişiminin, birleşimine oranı olarak ifade edilir ve değeri 0 ile 1 arasındadır. Bu durumda iki cümlenin Jaccard benzerliği cümlelerdeki kesişen kelime sayısının birleşimdeki kelime sayısına oranı olarak hesaplanır.

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{3.4}$$

3.2.3 Normalize Edilmiş Google Uzaklığı (NGD) Benzerlik Yöntemi

Bu yöntem terimler arasındaki uzaklığı hesaplamak için terimlerin kullanılış biçimlerini ele alır. Terimlerin birlikte kullanılma oranlarının yüksek olması bu iki terimin yakın olduğu yorumunun yapılabilmesine sebep olur. Orijinal olarak çalışma Google'ın depolamış olduğu ağ sayfalarından yararlanarak iki terim arasındaki anlamsal benzerliği hesaplayarak gerçekleştirilmiştir. Yöntemde, veri tabanı ve arama motoru olarak Google'ı kullanmak yerine başka veri tabanları ve arama motorları da kullanılabilir. Anlamsal benzerliği hesaplamak için terimlerin dokümanda tek başlarına ve birlikte geçtiği sayfa sayıları kullanılır. Eğer iki terim dokümanlar da beraber geçmiyor fakat ayrı ayrı bulunuyorlarsa sonuç sonsuz yani anlamsal açıdan uzak olarak çıkar. Eğer iki terim sürekli beraber geçiyorsa bu durumda da NGD değeri 0 çıkar. Bu da iki terimin çok kuvvetli bir şekilde birbirleriyle ilgili olduğunu gösterir. Tabi bu yöntem kullanılarak gerçekte hiç benzer olmayan iki terimin birbirine benzer anlamlarda çıkması olasılığı da vardır. Ama bu durum kullanılan veri tabanının büyümesiyle en aza iner [23].

Tekli doküman özetlemede bu yöntemi kullanmak için doküman, veri tabanı olarak kullanılır. Cümleler de sayfalar gibi işleme alınır. Diğer bir deyişle terimin cümlede kaç defa geçtiği sayılır. Yöntemi kullanabilmek için ilk olarak terimler arasındaki benzerliğin hesaplanması gerekir. Terimler arası benzerlik aşağıdaki gibi tanımlanır:

$$sim_{NGD}(t_k, t_l) = \exp(-NGD(t_k, t_l)) \tag{3.5}$$

İki terimin NGD değeri ise aşağıdaki gibi hesaplanır:

$$NGD(t_k, t_l) = \frac{\max\{\log(f_k), \log(f_l)\} - \log(f_{kl})}{\log(n) - min\{\log(f_k), \log(f_l)\}} \tag{3.6}$$

Denklem (3.6)'daki f_k değeri t_k terimi geçen cümlelerin sayısını ifade eder. f_{kl} değeri ise her iki terimin de beraber geçtiği cümle sayısıdır. Denklem (3.5)'deki formülden yararlanarak iki cümle arasındaki benzerlik aşağıdaki şekilde tanımlanmıştır[9]:

$$sim_{NGD}(S_i, S_j) = \frac{\sum_{t_k \in S_i} \sum_{t_l \in S_j} sim_{NGD}(t_k, t_l)}{m_i m_j} \quad (3.7)$$

Bu yöntemi kullanmak için aşağıdaki özelliklerin dikkate alınması gerekir:

- Eğer $t_k = t_l$ veya $t_k \neq t_l$ fakat $f_k = f_l = f_{kl} > 0$ ise $sim_{NGD}(t_k, t_l) = 1$ alınır.
- Eğer $f_k > 0, f_l > 0 \; ve \; f_{kl} = 0$ ise $NGD(t_k, t_l) = 1$ alınır. Böylece $0 < sim_{NGD}(t_k, t_l) < 1$ olması sağlanır.
- Eğer $0 < f_{kl} < f_l < f_k < n$ ve $f_k . f_l > n \cdot f_{kl}$ ise $0 < sim_{NGD}(t_k, t_l) < 1$ olur.
- $sim_{NGD}(t_k, t_k) = 1 \; ve \; sim_{NGD}(t_k, t_l) = sim_{NGD}(t_l, t_k)$

3.3 Metin Özetleme Algoritması

TextRank algoritmasını gerçeklemek amacıyla ilk olarak düğüm noktaları cümleler olan yönsüz çizge oluşturulmuştur. Metin tek bir konu hakkında olduğundan tüm cümleler birbiriyle ilgilidir. Bundan dolayı düğümler tam bağlı olacak şekilde çizge oluşturulmuştur. Yani tüm düğümler arasında bir bağ yapısı kurulmuştur. Çizge yapısını sağlamak için açık kaynak kodlu JUNG sistemi kullanılmıştır [24]. JUNG veriyi ağ ya da çizge şeklinde oluşturulup analiz, modelleme gibi işlemlerin yapılmasını sağlayan bir kütüphanedir. Kütüphane içinde veri madenciliği, istatistiksel analiz, optimizasyon, sosyal ağ analizi vb. yapmayı sağlayan birçok algoritma içerir. Daha sonra düğümler arasındaki ağırlık katsayıları yukarıda formülleri verilen üç farklı yöntem ile hesaplanmıştır. Ağırlık hesaplaması yapıldıktan sonra PageRank algoritması çalıştırılıp her düğüm için puan verilmesi sağlanmıştır. Puanlama işleminin bitmesinden sonra cümleler puanlarına göre büyükten küçüğe sıralanmıştır. Ve özette olması gereken kelime sayısına ulaşana kadar sıralanmış cümlelerden sırasıyla seçim işlemi gerçekleştirilir.

3.4 Benzerlik Yöntemlerinin Değerlendirilmesi

Yöntemlerin analizi DUC 2002 veri setinin kullanılmasıyla yapılmıştır. Veri setinde 533 doküman ve bu dokümanların insanlar tarafından çıkartılmış özetleri mevcuttur. TextRank'te kullanılan 3 yöntemin ölçülmesi referans ve sistem özetinde bulunan ortak

n-gram'ların sayısı kullanılarak, ROUGE-N hesaplanarak yapılır [21]. Bu çalışma kapsamında kelime bazlı ROUGE-1 kullanılmıştır. Ve özet dokümanlarda 100 kelime olacak şekilde sistem özeti çıkartılmıştır. Buna ek olarak daha detaylı bir analiz gerçekleştirmek adına dokümanlar kısa ve uzun olmak üzere 2 ayrı grupta incelenmiştir. DUC veri setinde toplam kelime sayısı 500'den az olanlar kısa, diğerleri uzun olarak alınmıştır. Buna ek olarak dokümanlardaki kelimeler olduğu gibi, kökleri bulunarak, hem kökleri hem de gereksiz kelimelerin çıkartılmasıyla 3 ayrı ön işleme kategorisine göre de incelenmiştir. Buna ilave olarak, metotların farklı metinlerde nasıl sonuç verdiğini araştırmak adına CAST veri seti üzerinde de analizler yapılmıştır. Bu veri seti üzerinde analizi gerçekleştirebilmek için hassasiyet, anma ve f-ölçütü kullanılmıştır.

3.5 Sonuçlar

Çizelge 3.1 DUC 2002 Temel ve Ön İşleme Yapılmış Sonuçlar

	ROUGE-1		
	Temel	Kelime Kökleri	Kelime kökleri ve Gereksiz Kelimelerin Çıkarılması
TextRank İçerik Çakışması	0,4518	0,4752	0,3998
TextRank NGD	0,4372	0,4607	0,3894
TextRank Kosinüs Benz.	0,4111	0,4350	0,3546
TextRank Jaccard Benz.	0,4470	0,4680	0,3910
Baz çizgisi	0,4599	0,4779	0,4162

Çizelge 3.1 DUC veri setine göre bulunmuştur ve 533 dokümanın ortalama sonuçlarını içerir. Baz çizgisi, dokümanlardaki ilk 100 kelimeyi içeren cümlelere göre hesaplanmıştır. Sonuçlar analiz edildiğinde içerik çakışma yöntemiyle yapılan özetlerin sonuçları diğer üç sonuca göre yaklaşık %2 daha iyi olduğu ve baz çizgisine %o8 farkla yakın olduğu görülmektedir. Biz bu çalışmada benzerliği ölçmek için TextRank'ın kullandığı içerik çakışması dışında bu çalışmada NGD, kosinüs ve Jaccard benzerlik yöntemlerini kullandık. Bu üç yöntem karşılaştırıldığında içlerinde başarım açısından en kötü sonucu veren kosinüs benzerliğidir. Bunun sebebi kosinüs benzerliğinde hesaplama yapılırken baz vektöründe kelimelerin iki cümlede de geçmesine göre sonuç bulunmasıdır. Yani iki cümleden birinde bu kelime geçmiyorsa o bileşenden 0 değeri gelmektedir. NGD özelliği kelimelerin doküman içinde beraber geçmesine bağlı olarak iki kelimenin yakınlığını tayin eder. Daha önceden de belirtildiği üzere bu yöntemde aslında benzer olmayan iki kelimenin beraber geçmesi onları benzer olarak bulunmasını sağlayabilir. Bununla birlikte kosinüs benzerliğine göre daha iyi sonuç verse de Jaccard benzerliğini geçemediği görülmüştür. Jaccard benzerliği ve içerik çakışmasının sonuçları birbirine çok yakındır. Bunun sebebi ikisi de pay değerlerini kesişim olarak almaktadır. Fakat payda da Jaccard birleşimin büyüklüğünü kullanırken, diğer yöntem 2 cümle uzunluklarının logaritmalarının toplamını kullanarak normalize eder.

Çizelge 3.2 DUC veri seti kısa ve uzun doküman sonuçları

	Temel		Kelime Kökleri	
	Kısa	Uzun	Kısa	Uzun
TextRank İçerik Çakışması	0,4974	0,4097	0,5235	0,4307
TextRank NGD	0,4874	0,3909	0,5103	0,4148
TextRank Kosinüs Benz.	0,4679	0,3586	0,4892	0,3849
TextRank Jaccard Benz.	0,4898	0,4074	0,5099	0,4294

Çizelge 3.3 DUC veri seti kısa ve uzun ön işlemden geçirilmiş sonuçlar

	Kelime kök ve Gereksiz Kelime Çıkarımı	
	Kısa	Uzun
TextRank İçerik Çakışması	0,4503	0,3534
TextRank NGD	0,4564	0,3269
TextRank Kosinüs Benz.	0,4175	0,2969
TextRank Jaccard Benz.	0,4395	0,3458

Çizelge 3.2 ve 3.3'te benzerlik yöntemlerinin kısa ve uzun dokümanlardaki etkisi görülmektedir. Uzun ve kısa dokümanlarda içerik çakışması yönteminin yine en iyi sonucu verdiği gözlemlenir. Kısa makalelerde ROUGE değerinin yüksek olması çakışan kelime sayılarının daha fazla bulunabildiğini gösterir. Bu üç yöntem arasında kosinüs benzerliği kısa dokümanlarda %41'lik bir başarı yakalasa bile diğer 2 yöntem ile aralarındaki fark hala %2-%4 olacak şekilde çok büyüktür. Bu sonuçlar içinde en dikkat çeken; kelime kökleri ve gereksiz kelimelerin atılmasıyla NGD yönteminin kısa dokümanlarda en iyi sonucu yakaladığıdır. Bunun sebebi sezgisel olarak kısa dokümanlarda beraber geçen kelimelerin oranının daha fazla olması şeklinde yorumlanabilir.

Çizelge 3.4 ve 3.5'te CAST dokümanlarının sonuçları veri setinin 3 farklı önişleme biçimi için verilmektedir. CAST veri setinde cümleler olduğu gibi bütünüyle seçildiğinden, değerlendirmeler ROUGE yerine hassasiyet ve anımsama ölçütleriyle yapılmaktadır. Tabloda görüldüğü üzere DUC veri setinde genellikle iyi sonuç veren içerik çakışması yöntemi bu sette 2. en iyi sonuca sahiptir. Araştırma sonucuna göre en iyi sonuç NGD benzerlik ölçütünün kullanılmasıyla elde edilmiştir. Kelime kökleri bulunarak yapılan özet kıyaslamasında DUC'tan farklı olarak hassasiyette yaklaşık binde 7'lik bir azalmaya sebep olmuştur. Hâlbuki DUC verisinde bu ön işleme biçimi değerlerin yükselmesini

sağlıyor. Buna ilaveten kelime köklerinin bulunmasının yanında gereksiz kelimelerin de atılması hassasiyeti %4 - %5 ve anımsama ölçütünü ise %2 - %3 seviyelerinde arttırılmasını sağlar.

Çizelge 3.4 CAST sonuçları

	Temel			Kelime Köklerinin Bulunması		
	Hassasiyet	Anımsama	F	Hassasiyet	Anımsama	F
TextRank İçerik Çakışması	0,4400	0,4217	0,4261	0,4473	0,4295	0,4336
TextRank Kosinüs Benz.	0,3181	0,2933	0,3018	0,3136	0,2853	0,2956
TextRank Jaccard Benz.	0,4201	0,3978	0,4041	0,4130	0,3910	0,3975
TextRank NGD	0,4640	0,4477	0,4511	0,4638	0,4477	0,4510

Çizelge 3.5 CAST sonuçları

	Kelime Kökleri ve Gereksiz Kelime Çıkarımı		
	Hassasiyet	Anımsama	F
TextRank İçerik Çakışması	0,4775	0,4543	0,4612
TextRank Kosinüs Benz.	0,3361	0,3053	0,3171
TextRank Jaccard Benz.	0,4479	0,4202	0,4292
TextRank NGD	0,4903	0,4718	0,4760

Sonuç olarak iki veri setinde de kosinüs yönteminin çok iyi olmadığı gözlemlenmiştir. Benzerlik yöntemlerinden NGD ya da içerik çakışması kullanılması veri setine ve özet çıkartma yöntemine göre farklılıklar gösterebildiği tespit edilmiştir. NGD kelimelerin benzerliğini yerel (lokal) olarak değerlendirip bulduğundan CAST'te daha başarılı olmuştur. Bunun sebebi DUC veri setinde referans özetteki cümleler yoruma dayalı oluşturulurken; CAST'te çıkartılan referans özetteki cümlelerin tamamen orijinal makaleden seçilmiş olmasıdır.

BÖLÜM 4

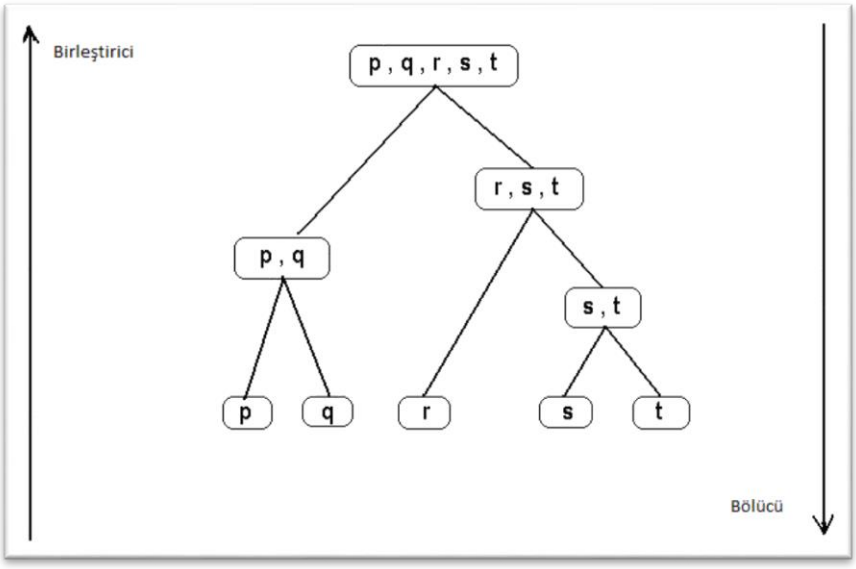
ÖNERİLEN ALGORİTMA

4.1 Hiyerarşik Birleştirici Kümeleme (HBK)

Veri kümeleme, çok boyutlu verideki doğal grupları ya da kümeleri belirli bir benzerlik ölçütü yardımıyla ortaya çıkartma işlemidir. Kümeleme yapılarak bir grupta toplanan verilerin diğer gruptaki verilere göre birbirlerine daha benzer olmaları beklenir. Veri madenciliği, örüntü tanıma, istatistiksel veri analizi, makine öğrenmesi vb. alanlarda çokça kullanılır. Dolayısıyla farklı disiplinlerdeki araştırmacılar kümeleme problemi üzerinde çalışmalarını sürdürürler.

Metin madenciliğinde üzerinde çalışılan konulardan biri doküman kümelemedir. Doküman kümeleme; dokümanları içeriğine ya da konularına göre gruplara ayırmaktır. Bu tür gruplamanın birçok sebebi vardır, bunlar; arama alanının genişletilmesi, özet çıkartmak, konu bulmak şeklinde örneklendirilebilir. Kümeleme yöntemleri genellikle veri setinin temsili, veriler arasındaki uzaklık ölçütü, kümelemeyi oluşturacak kriter fonksiyonu ve bu fonksiyonu optimize edecek yöntem olmak üzere dört bileşenden oluşur. Literatürde kullanılan birçok kümeleme modelleri bulunur ve hiyerarşik kümeleme (bağlantılı kümeleme) bunlardan biridir. Bağlantılı olarak adlandırılmalarındaki en etkin öğe birbirine yakın olan nesneleri uzaklığına göre bağlayarak kümeleme oluşturmasıdır. Dolayısıyla bu da farklı uzaklıklarda farklı kümelerin oluşmasına sebep olur. Ve her iterasyonda uzaklık kriterine bağlı olarak iki küme birleştirilir böylece hiyerarşik bir yapı oluşur. Bu hiyerarşik yapının gösterimi

dendrogram adı verilen bir ağaç diyagramında tutulur. Hiyerarşik kümeleme metotları oluşturulmalarına göre birleştirici ve bölücü olmak üzere iki gruba ayrılır. Birleştirici kümeleme metodunda başlangıçta her veri bir kümeye atanır ve kümeler her iterasyonda birleştirilerek hiyerarşi oluşturulur. Bölücü olan metotta ise bu işlemin tam tersi yapılır; ilk olarak tüm veriler bir kümede bulunur ve her iterasyonda kümeler bölünür. Hiyerarşik kümeleme ile ilgili örnek Şekil 4.1'de verilmiştir.



Şekil 4.1 Birleştirici ve Bölücü Hiyerarşik Kümeleme

4.1.1 HBK'nin Özellikleri

Hiyerarşik birleştirici kümeleme veri analizinde çokça kullanılan bir yöntemdir. Benzer grupların birleşmesi ile dendrogram denilen ikili ağaç oluşturularak verinin görsel özeti sunulmuş olunur. Klasik k-ortalama kümesinin kullanmış olduğu küme sayısı, ilk küme atanması ve uzaklık ölçütü gibi birçok kavram yerine sadece gruplar arasındaki benzerliğin tanımlanması gerekir. Kaç küme üzerinde çalışılmak isteniyorsa ilgili seviyeden dendrogram kesilir. Bu algoritmalar açgözlü algoritmaların karakterine sahiptirler [25]. Her bir adımda iki farklı küme birleştirilerek yeni küme oluşturulur. Bu durumda ilk başta n adet veri var ise n tane küme vardır ve birleştirme işlemlerinin en sonunda 1 küme elde edilir, böylelikle dendrogramda oluşturulmuş olur. Birleştirme işleminin seviyesi arttıkça birleşmiş kümelerin arasındaki benzerlik monoton olarak

azalır. Kümelerin birleştirilmesi için grup benzerliği diye adlandırılan birçok yöntem bulunmaktaysa da aşağıdaki üç yaygın yöntem kullanılır:

- Tekli Bağ: Denklem (4.1)'deki gibi kümeler arasındaki uzaklık farklı kümelerdeki en yakın objelere göre belirlenir.

$$d_{SL}(G,H) = \min_{i\in G, j\in H} d_{i,j} \tag{4.1}$$

- Tam Bağ: Bu metot ile kümeler arasındaki uzaklık, tekli bağın tam tersi bir şekilde farklı kümelerdeki en uzak objelere göre belirlenir.

$$d_{CL}(G,H) = \max_{i\in G, j\in H} d_{i,j} \tag{4.2}$$

- Grup ortalaması: Bu yöntem ile iki küme arasındaki mesafe iki farklı kümedeki tüm objelerin uzaklıklarının ortalaması olarak denklem (4.3)'deki gibi hesaplanır.

$$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i\in G} \sum_{i\in H} d_{i,j} \tag{4.3}$$

Tekli bağ kullanılması sonucu yakın olan verilerin erkenden bir araya gelmesiyle zincir etkisi oluşabilir. Diğer bir deyişle verilerin bir veya iki kümede birikmesi ihtimali vardır. Ya da tam bağ yöntemi kullanılarak bunun tam tersi bir etki görülebilir. Birleştirme işlemine önderlik eden grup benzerlik algoritmasının farklı seçilmesi dendrogramın çok farklı çıkmasına sebep olur. Buna ek olarak HBK kullanılması hiyerarşik bir yapı içermeyen veri kümelerinin de hiyerarşik gibi davranılmasına sebep olur. Bu sebeplerden dolayı hiyerarşik kümeleme dikkatli bir biçimde kullanılmalıdır.

4.1.2 Algoritmanın Uygulaması

Bu çalışmada hiyerarşik birleştirici kümeleme algoritması Java yazılım dili kullanılarak yazıldı. Grup benzerliği yöntemi olarak tek ve tam bağ kullanıldı. Algoritma ilk olarak toplam veri sayısını ve veriler arasındaki benzerlik matrisini alır. Kümeleme algoritması metin özetleme için çalıştırılacağından, ilk adımda her cümle bir kümeye gelecek şekilde atanır. Böylece dendromun ilk seviyesi oluşturulmuş olur. Sonra algoritma sırasıyla önce birbirine en yakın iki kümeyi bulur ve bunları birleştirir. Yeni birleşmiş küme ile diğer

kümeler birlikte dendromun yeni bir seviyesini oluşturur. Birbirine en yakın kümeleri bulmak için tüm kümeler birbirleri ile uzaklık hesabına tabi tutulur; bu çalışmada hesap tek ve tam bağ yöntemi ile yapılır. En kısa mesafeli olan kümelerin indis değerleri fonksiyonun çalışması sonucu döndürülür. Birleştirme işleminde ise hesaplanan indis değerlerinden büyük olan indisteki kümenin elemanları indisi küçük olana doğru taşınır ve büyük indisli küme silinir. Algoritmanın çalışması aşağıdaki örnekle açıklanabilir:

1. C(1)=1, c(2)=2, c(3)=3, c(4)=4, c(5)=5, c(6)=6. İkinci ve üçüncü küme birbirine en yakın olduğunu var sayalım. Bu durumda birleştirme işleminin sonucu adım 2'deki gibi olur.
2. C(1)=1, c(2)={2,3}, c(3)=4, c(4)=5, c(5)=6. Eğer bu adımda 4. ve 5. Küme birbirlerine en yakın ise bundan sonraki seviye 3. adımda gösterildiği gibi olur.
3. C(1)=1, c(2)={2,3}, c(3)=4, c(4)={5,6}. 3. seviyede ise 2. ve 4. kümeler yakın olsun.
4. Bu durumda C(1)=1, c(2)={2, 3, 5, 6}, c(3)=4, şeklinde kümeleme yapılmış olur.

4.2 Önerilen Algoritmanın Çalışması

Çalışma kapsamında yeni bir metin özetleme algoritması ortaya konulmuştur. Algoritma ilk etapta cümleler için benzerlik matrisi oluşturur. Çalışma kapsamında benzerlik matrisi denklem (3.3)'teki gibi cümlelerin kosinüs benzerliklerine göre oluşturulur.

$$Cümle\ Benzerlik\ Matrisi = \begin{bmatrix} 0 & \cdots & S_{1,n} \\ \vdots & \ddots & \vdots \\ S_{n,1} & \cdots & 0 \end{bmatrix} \tag{4.4}$$

Dokümanda n adet cümle var ise denklem (4.4)'teki gibi n x n'lik kare bir matris elde edilir. Matrisin satır ve sütunları dokümandaki her bir cümleye karşılık gelir ve $S_{n,1}$ ile gösterilen hücre 1. ve n. cümlelerin kosinüs benzerliğidir. Cümlelerin kendisi ile olan benzerliği önemsiz olduğundan değerlendirilmeye alınmamıştır. Doküman için oluşturulan benzerlik matrisi hiyerarşik kümeleme yapılabilmesi için kullanılır. Hiyerarşik birleştirici kümelemenin özelliğine göre ilk durumda her cümle bir kümeyi temsil eder. Grup benzerliği olarak tek bağ kullanılırsa, kümelerden en yakın olan iki cümle birleştirilir ve böylece yeni bir seviye eklenmiş olur. Kümeleri ikili olarak birleştirme işlemi sadece bir küme kalıncaya kadar devam eder ve sonuçta dendrogram oluşmuş olur. Böylece

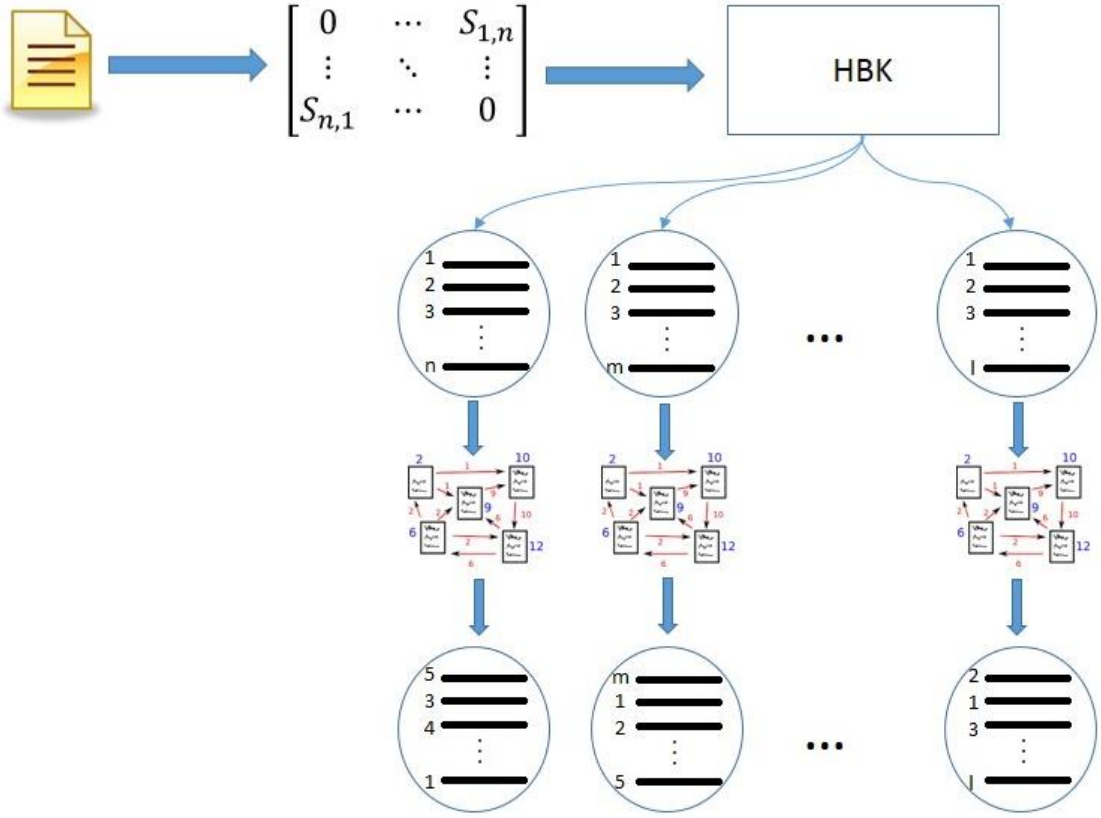
dendrogramdan seçilen bir seviyede, farklı kümelerde birbirlerine daha yakın cümlelerin bir arada olması sağlanmış olur. Sonraki adımda kümeler eleman sayılarına göre büyükten küçüğe doğru sıralanır. Çünkü konular eleman sayısı fazla olan kümelere doğru yoğunlaşma gösterir. Bu da dokümanın o konu çerçevesinde daha fazla bilgi içerdiğini gösterir. Sıralama sonrasında her bir kümeden bir cümle seçilir. Seçim işlemi maksimum kelime sayısı veya olması gereken cümle sayısı tamamlanana kadar yapılır. Burada önemli noktalardan biri kümelerden cümlelerin nasıl seçileceğidir. Çünkü kümeleme işleminden sonra bir kümede bulunan cümlelerin aynı konu üzerinde olacağı düşüncesiyle birbirlerine benzer olması beklenir. Bundan dolayı özette, kümelerde bulunan cümlelerden en önemli olanların seçilmesi gerekir. Bu noktada genellikle çalışmalarda belli kriterlere göre kümelerdeki merkez cümle bulunarak seçim yapılmaktadır [8], [9]. Bu çalışma kapsamında ise kümeden seçilecek cümleyi bulabilmek için her küme içinde TextRank algoritması kullanılmıştır. Böylelikle her küme içerisinde alt çizgeler oluşturularak o küme içindeki cümlelerin puanlandırılması sağlanmıştır. Bu puanlama ile en yüksek puanlı cümle özette bulunmak üzere seçilir. Eğer küme tek elemanlı ise cümle direk seçilir. Algoritmanın detayları aşağıdaki şekil ve çizelgelerde verilmiştir.

Çizelge 4.1 Hibrit Algoritmanın Sahte Kodu

```
Fonksiyon özet çıkar
  Cümleler için benzerlik matrisi hesapla
  Benzerlik matrisi kullanarak hiyerarşik küme oluştur
  Küme sayısı belirleyip o seviyeden kümeleri al.
  Kümeleri boyutuna göre büyükten küçüğe sırala.
   For i=1 to küme sayısı
      Kümeye TextRank uygula.
      En yüksek puanlı indisi özete ekle
End
```

Çizelge 4.1'de önerilen algoritmanın sahte kodu verilmiştir. Bu koda göre kümeleme yapıldıktan sonra dendrogramdan kaç küme olması isteniyorsa o seviyeden kesilir ve kümeler seçilmiş olur. Daha sonra kümeler büyükten küçüğe doğru sıralanır. Her bir küme için TextRank uygulanır ve puanlar büyükten küçüğe sıralanmış bir biçimde ilgili indisler döndürülür. Şekil 4.2'de önerilen algoritmayı anlatan bir diyagram verilmiştir. Temsili diyagramda dokümanda bulunan n cümle için matris oluşturulur. Bu matris

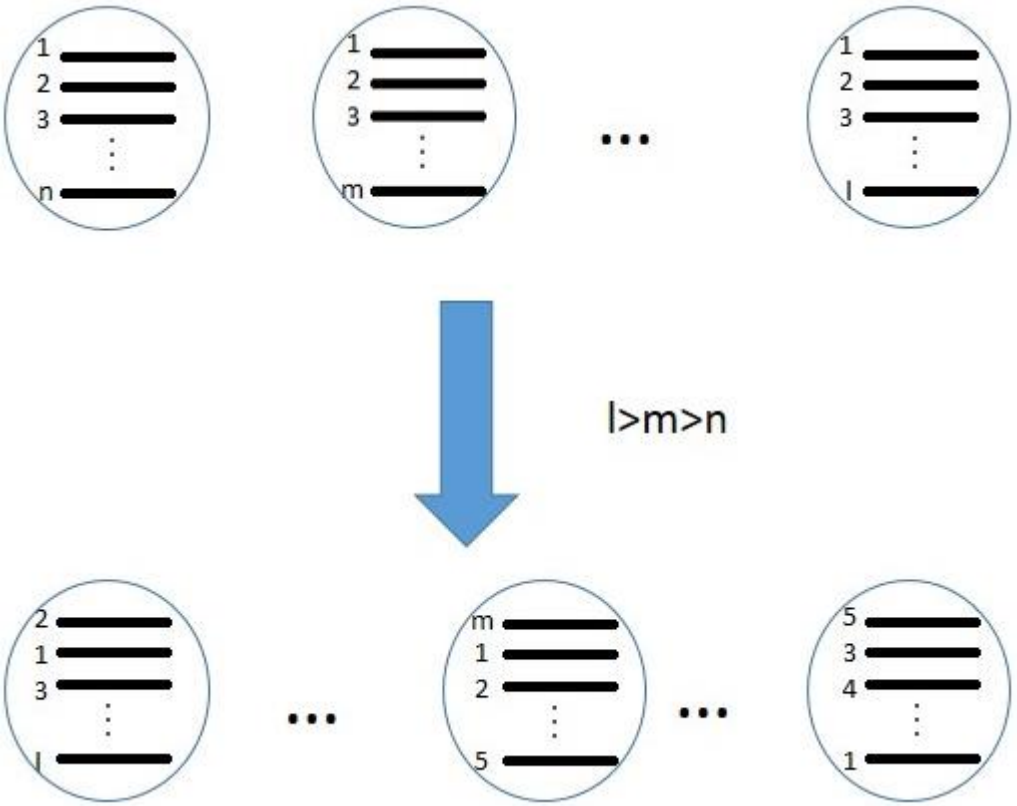
kullanılarak HBK yardımıyla kümeleme yapılır. Kümelemenin ardından her bir kümede bulunan cümlelerle bir çizge oluşturulur. Eğer kümedeki cümle sayısı bir ise onun indeksi dönüş değeri, çizge oluşturulmadan direk, olarak verilir. Fakat eleman sayıları birden fazla ise bu durumda bu kümeler için ilk olarak çizgenin düğümleri oluşturulur. Ardından düğümler arasındaki ağırlık değerleri içerik çakışması yöntemine göre hesaplanır. "TextRank" algoritması kullanılarak puanlama hesaplanır. Puanlamadan sonra indisler puana göre büyükten küçüğe sıralanır.



Şekil 4.2 Hibrit Algoritma

Kümelerin büyüklükleri l>m>n ise sıralama şekil 4.3'teki gibi değişir. Sıralamanın ardından her bir kümeden en yüksek puanlı bir cümle alınarak özet oluşturulur. Şekle göre 2, … , m, … , 5 indisli cümleler özete girmek üzere seçilir. Fakat bu cümlelerin hepsi özette bulunmak zorunda değildir. Bu sıralama bozulmayacak şekilde özette bulunması gereken kelime sayısı veya oran kadar cümle seçilir. Eğer seçilen cümleler özeti

oluşturmaya yetmezse seçim işlemi şekil 4.3 alt kısmındaki kümelerin ikinci indislerden yani 1, … , 1, … 3 sırasından seçilir. Farklı kümelerden cümleler seçilerek özete farklı alanlardan kelimelerin girmesi sağlanır.



Şekil 4.3 Kümelerin Büyüklüğüne göre Sıralama

Detayları verilmiş olan önerilen sistemin yukarıda verildiği gibi önce kümeleme yapılıp sonrasında her küme için "TextRank" uygulanmıştır. Hiyerarşik kümeleme yapılarak cümlelerin yapısal olmayan kümeleme yöntemlerine göre daha fazla bilgi verici olması beklenir. Yani bu yöntemin sağlamış olduğu bazı özellikler metinler için de faydalı olabilir. Örnek olarak, değişik boyuttaki kümeler daha önceden ortaya çıkmamış bir bilgi içerebilir. Böylece diğer yöntemlerde özete girmesi mümkün olmayan bir cümlenin de özete girmesi ihtimali olmuş olur. Kümeleme sonrası verilerin sıralamaları değişir, yeni bilgi ortaya çıkarıcı küçük kümeler oluşabilir. Bu özelliklerin ışığında kümeleme işlemi sonrasında elde edilen kümelerdeki cümlelerin birbirine yakın olması beklenir. Böylece özete seçilen cümlelerin farklı konularda olması sağlanır ve doküman bütünlüğünü destekleyen bir seçim işlemi yapılmış olur.

4.3 Algoritmanın Değerlendirilmesi

Algoritmayı değerlendirmek için bir önceki bölümde olduğu gibi DUC 2002 ve CAST veri setleri kullanılmıştır. DUC 2002 üzerinde değerlendirme yapabilmek için Rouge-1 ölçütü; CAST setinde değerlendirme yapabilmek için de keskinlik, anma ve f-ölçütü kullanılmıştır. Önerilen sistemin çalışmasındaki önemli noktalardan biri küme sayısının seçilmesidir. İdeal küme sayısının seçilmesi verinin gösterimine ve benzerlik ölçütü gibi kriterlere bağlı olduğundan kolay değildir. Bu sebeple Aliguliyev'in yapmış olduğu çalışmada kullandığı yönteme göre küme sayıları belirlenir [26]. Bu yöntem dokümandaki kelime dağılımlarını kullanarak ideal küme sayısını bulmaya çalışır. Başka bir deyişle dokümandaki konu sayılarını tahmin etmeye çalışır. Yöntem aşağıdaki denklem 4.5'te verildiği gibidir.

$$k = n\frac{|D|}{\sum_{i=1}^{n}|S_i|} = n\frac{|\cup_{i=1}^{n} S_i|}{\sum_{i=1}^{n}|S_i|} \tag{4.5}$$

Denklem 4.5'e göre küme sayısı toplam terim sayısının birikmiş terim sayısına oranının, cümle sayısı olan n ile çarpımı olarak belirlenir. $|S_i|$ cümle i'deki terim sayısını, $|D|$ ise dokümandaki terim sayısını gösterir. Denklem analiz edildiğinde eğer her cümle aynı kelimelerden oluşursa küme sayısı 1 çıkar ki aslında bu durumda dokümanda 1 cümle var demektir. Eğer cümleler arasında hiç ortak kelime bulunmuyorsa bu durumda küme sayısı cümle sayısıyla eşit olur. Bu iki özel durum dışında ise küme sayısı 1 ile n arasındadır.

Kelime dağılımlarının yanı sıra küme sayısının özete etkisini analiz etmek amacıyla değişik küme sayıları ile de deneysel çalışmalar yapılmıştır. Denemelerde küme sayısı o doküman için sağlanamıyorsa onun için küme sayısı cümle sayısının yarısı olarak verilmiştir.

Sonuçlar dokümanda hiçbir ön işlem yapmadan (temel), kelimelerin kökleri bulunarak ve de hem kelime kökleri bulunarak hem de gereksiz kelimelerin çıkartılmasıyla olmak üzere 3 farklı kategoride analiz yapılmıştır.

4.3.1 Sonuçlar

Tekli bağla yapılan DUC 2002 sonuçları çizelge 4.2'de, tam bağ ile yapılan sonuçlar çizelge 4.3'te gösterilmiştir.

Çizelge 4.2 Önerilen Sistem DUC 2002 Sonuçları (Tek Bağ)

Küme Sayıları	Temel	Kökleri Bulunmuş	Kök Bulunmuş ve Gereksiz Kelime Çıkarımı
Ort. 14	0,4756	0,4951	0,4043
Baz çizgisi	0,4599	0,4779	0,4162

Çizelge 4.3 Önerilen Sistem DUC 2002 Sonuçları (Tam Bağ)

Küme Sayıları	Temel	Kökleri Bulunmuş	Kök Bulunmuş ve Gereksiz Kelime Çıkarımı
Ort. 14	0,4408	0,4664	0,3806
Baz çizgisi	0,4599	0,4779	0,4162

Hiyerarşik kümelemede tek ve tam bağlı olmak üzere iki tip gruplama sonuçlarına bakıldığında oluşan küme yapılarının tamamen farklı olduğu sonuçlar üzerinden görülmektedir. Tam bağ ile elde edilen sonuç tek bağa göre genelde %2-%3 daha düşük çıkmıştır. DUC sonucu üzerinden önerilen sistem ile "TextRank"'ın değişik benzerlik metodları kıyaslanacak olursa, tek bağ yönteminin sonuçları ortalama %2 oranında arttırdığı görülmektedir.

Çizelge 4.4'te farklı küme sayıları ile yapılan deneylerin sonuçları verilmiştir. Küme sayısı 4 olduğunda %44 ile en düşük sonuç elde edilmiştir. Küme sayısı 8 ile 24 arasında olduğu durumlarda %47'lik başarı sağlanmıştır. 8 ile 24 arasındaki küme sayılarındaki değişiklik, sonuçları %o1 - %o6 arasında etkilediği gözlemlenmiştir.

TextRank yönteminin kümelerden cümle seçimine olan katkısını kıyaslamak için daha farklı bir cümle seçim yöntemi kullanılarak deneme yapılmıştır. Bu yönteme göre kümelerden en uzun cümleler seçilerek özet oluşturulmuştur. Deney sonuçları çizelge 4.5'te verildiği gibidir. Sonuçlar temel kriterinde analiz edildiğinde çizelge 4.2'deki %47'lik sonuca %o6 farkla çok yakın olduğu gözlemlenmiştir.

Önerilen yöntemde kümeleme aşamasında benzerlik ölçütü olarak kosinüs benzerliğinden yararlanılmıştır. Çalışmada hiyerarşik birleştirici kümelemenin

performansını Jaccard, NGD ve içerik çakışması yöntemlerinin ne kadar etkilediği de analiz edilmiştir. Diğer yakınlık kriterlerinin sonuçlara etkisi çizelge 4.6'da verilmiştir. Sonuçlara göre TextRank'te en başarılı olan benzerlik yöntemlerinin hiyerarşik birleştirici kümelemede performansını kaybettiği görülmüştür. Bunun en önemli sebebi TextRank yönteminde puanlamanın sadece 2 cümlenin birbirine olan benzerliğine bağlı kalmadan ona bağlı olan diğer cümlelerin de benzerliğini özyinelemeli olarak hesaba katmasıdır. Fakat hiyerarşik kümelemede, açgözlü algoritmaların çalıştığı şekilde, iki grubu birleştirmek için sadece iki cümlenin birbirine yakın olması yeterlidir.

Tam bağda ise çizelge 4.3'teki temel ve kelime kökler bulunmuş olan kriterler göz önünde bulundurulduğunda; NGD ve kosinüs benzerliğinden daha iyi sonuçlar verdiği gözlemlenmiştir. Kelime kökleri bulunup gereksiz kelimelerin çıkarıldığı sonuçlar kıyaslanırsa 4 yöntemden sadece kosinüs benzerliğini geçtiği görülür.

Çizelge 4.4 Farklı Küme Sayıları Sonuçları

Küme Sayısı	Temel	Kelime Köklerinin Bulunması	Kelime Kökleri ve Gereksiz Kelime Çıkartımı
4	0.448	0.475	0.377
8	0.472	0.495	0.393
12	0.472	0.493	0.406
16	0.478	0.499	0.403
20	0.475	0.5	0.404
24	0.474	0.498	0.402

Çizelge 4.5 TextRank Yerine En Uzun Cümle Seçimi ile Alınan Sonuçlar

	Temel	Kelime Köklerinin Bulunması	Kelime Kökleri ve Gereksiz Kelime Çıkartımı
En Uzun Cümle	0.469	0.489	0.42

Çizelge 4.6 Diğer yakınlık kriterlerinin sonuca etkisi

	Temel	Kelime Köklerinin Bulunması	Kelime Kökleri ve Gereksiz Kelime Çıkartımı
Jaccard	0.440	0.456	0.394
NGD	0.408	0.430	0.359
İçerik Çakışması	0.416	0.434	0.382

Çizelge 4.7, 4.8, 4.9 ve 4.10'de CAST veri setleri ile bulunmuş sonuçlar gösterilmektedir.

Çizelge 4.7 Önerilen Sistem CAST Sonuçları (Tek Bağ)

Küme sayısı	Temel			Kökleri Bulunmuş		
	Hassasiyet	Anma	F	Hassasiyet	Anma	F
Ort. 14	0,4382	0,4165	0,42229	0,4293	0,4086	0,4143

Çizelge 4.8 Önerilen Sistem CAST Sonuçları (Tek Bağ)

Küme sayısı	Kök Bulunmuş ve Gereksiz Kelime Çıkarımı		
	Hassasiyet	Anma	F
Ort. 14	0,4387	0,4086	0,4188

Çizelge 4.7 ve 4.8'deki tek bağ grup benzerlik yöntemi kullanılarak CAST veri setiyle elde edilen sonuçlar verilmiştir. Sonuçları hassasiyet, anma ve F ölçütü üzerinden TextRank sonuçları ile kıyaslayınca; temel ve kökleri bulunmuş kriterde TextRank'taki 4 sonuçtan kosinüs ve Jaccard benzerliğini geçtiği görülmektedir. Son olarak kökleri bulunmuş ve gereksiz kelimeler çıkartılmış sonuçlar bulunduğunda ise sadece kosinüs benzerliğini geçtiği görülür.

Çizelge 4.9 Önerilen Sistem CAST Sonuçları (Tam Bağ)

Küme sayısı	Temel			Kökleri Bulunmuş		
	Hassasiyet	Anımsama	F	Hassasiyet	Anma	F
Ort. 14	0,3998	0,3769	0,3837	0,4065	0,3793	0,3883

Çizelge 4.10 Önerilen Sistem CAST Sonuçları (Tam Bağ)

Küme sayısı	Kök ve Gereksiz Kelime Çıkarımı		
	Hassasiyet	Anımsama	F
Ort. 14	0,3993	0,3725	0,3815

DUC sonuçlarına benzer bir şekilde CAST'in çizelge 4.9 ve 4.10'daki gibi tam bağ ile elde edilen sonuçlarında tek bağa göre ortalama %3 - %4'lük bir düşüş meydana gelmiştir. "TextRank" sonuçlarıyla kıyaslandığı 4 metottan sadece kosinüs benzerliğini geçtiği

görülmüştür. Önerilen sistemde tam bağ kullanıldığında CAST sonuçlarına göre NGD, içerik çakışması ve Jaccard benzerlik ölçütleri karşısında başarılı olmadığı görülmüştür.

Özet olarak; yapılan çalışmalar sonucu önerilen sistemin tek bağ kullandığı durumda DUC veri seti için %2'lik bir geliştirme gösterdiği gözlemlenmiştir. Tam bağ ile yapılan ölçümlerde yüzde değerleri düşse de "TextRank" sonuçlarına yakın bir performans gösterdiği saptanmıştır. CAST veri setinde de tek bağ ile yapılan çalışmanın tam bağa göre daha performanslı sonuç verdiği saptanmıştır. Fakat önerilen sistem genel olarak "TextRank"'te kullanılan 2 yöntemi geçebilmiştir. Bu da CAST veri seti için NGD veya içerik çakışması kullanmanın daha avantajlı olacağı yorumunu doğurur. Bunun yanında yapılan gözlemlere göre metin özetleme sistemleri için hiyerarşik kümeleme kullanırken tek bağ kullanmanın tam bağa göre daha avantajlı olduğu sonucuna varılır.

4.3.1.1 Diğer Çalışmalar ile Kıyaslama

Rasim Alguliev ve Ramiz Aliguliev çalışmasında kümeleme ve cümle çıkarımı yapan gözetimsiz doküman özetleme metodu önerir [9]. Yazar çalışmasında kümeleme yapabilmek için kriter fonksiyonları tanımlar. Daha sonra bu fonksiyonları kullanarak uyumluluk fonksiyonlarını oluşturur. Uyumluluk fonksiyonunu kesikli diferansiyel evrim algoritması ile optimize eder. Daha sonra kümedeki merkezilik özelliğine göre cümleler seçilir. Yapılan çalışmada alınan sonuçlar aşağıdaki çizelgede verilmiştir.

Çizelge 4.11 Çalışma [9] ile kıyaslama

Yöntem	ROUGE-1
F	0.451
F1	0.449
F2	0.454
CRF	0.441
QCS	0.448
HITS	0.428
SVM	0.434
Önerilen Yöntem	0.475

Çalışma [9]'da alınan sonuçlar farklı kriter fonksiyonları olan F, F1 ve F2 için verilmiştir ve seçilen diğer yöntemler ile kıyaslanmıştır. Yazarın önerdiği çalışmasında aldığı

sonuçlar diğer yöntemleri geçmiştir. Bizim önerdiğimiz yöntemin yazarın önerdiği yönteme göre %2 daha performanslı sonuç ürettiği gözlemlenmiştir.

Ramiz M. Aliguliyev çalışmasında [9]'daki çalışmasına benzer cümle kümeleme ve çıkartımı yapan bir çalışma daha önermiştir [26]. Bu çalışmasında sonuçların özetlemenin sadece optimize edilen fonksiyonlara değil benzerlik metotlarına da bağlı olduğunu belirtmiştir. Yapılan çalışmanın DUC 2002 sonuçlarına göre kıyasları aşağıdaki çizelge 4.9'daki gibidir.

Çizelge 4.12 Çalışma [26] ile kıyaslama

Yöntem	ROUGE-1
F	0.466
F1	0.456
F2	0.442
NetSum	0.449
CRF	0.440
SVM	0.432
Manifold-Ranking	0.423
Önerilen Yöntem	0.475

[26]'de önerilen çalışmanın sonuçları F, F1, F2 ile çizelge 4.12'da gösterilmiştir. Yazar yaptığı çalışmasında [9]'daki sonuca göre daha performanslı bir sonuç elde etmiştir. Ve diğer yöntemlerden daha iyi sonuçlar bulmuştur. Burada önerdiğimiz sistem F ile kıyaslandığında yaklaşık %1, F1 ile kıyaslandığında %2 ve F2 ile kıyaslandığında %3 daha iyi bir performans sergilediği saptanmıştır.

Zhang Pei-ying ve Li Cun-he çalışmasında cümle kümelemeye dayalı bir özetleme yöntemi önermiştir [8]. Önerdikleri yöntem üç aşamadan oluşur; ilk olarak dokümandaki cümleler anlamsal yakınlıklarına göre K ortalama yöntemi ile kümelenir. Sonra her küme için toplayıcı cümle benzerliği hesaplanır. Son aşamada her küme için merkez cümle toplayıcı benzerlik fonksiyonu ile bulunur. Daha sonra da merkez cümleler fonksiyonda bulunan değerlerine göre büyükten küçüğe sıralanır. Ve en fazla ağırlık değeri alan cümlelerin özette bulunması sağlanır. Bu çalışmada sonuçlar 3 yöntem ile kıyaslanmıştır. Sonuçlar çizelge 4.13'daki gibi verilmiştir.

Çizelge 4.13 Çalışma [8] ile kıyaslama

Yöntem	ROUGE-1
MMR	0.348
WAA	0.380
Çalışma [4]'teki yöntem	0.435
Önerilen Yöntem	0.475

Çalışmada yazarın önerdiği yöntem MMR ve WAA yöntemini geçtiği görülmüştür. Bizim önerdiğimiz yöntem yazarın yöntemine göre %4 daha iyi bir performans sergilemiştir.

Miner Berker ve Tunga Güngör çalışmasında cümle çıkarımına dayalı bir yöntem üzerinde çalışmıştır [27]. Bu yöntemde cümleleri puanlamak için anlamsal zincir özelliğini de içeren toplam 12 farklı özellik kullanılmıştır. Hangi özelliğin daha önemli olduğunu bulmak için kat sayı hesaplaması genetik algoritma kullanılarak yapılmıştır. Bulunan kat sayılar kullanılarak özet çıkarılmış ve denemeler bu şekilde yapılmıştır. Yazar çalışmasında denemeler için CAST veri setini kullanmıştır. Çalışmada bulunan ortalama hassasiyet değeri 0.46 olarak verilmiştir. Bizim önerdiğimiz algoritmanın ortalama hassasiyeti 0.438'dir. Fakat bir önceki bölümde verilen TextRank sonuçları ele alınırsa NGD benzerlik yöntemiyle her üç sonucunda 0.46'yı geçtiği gözlemlenir.

4.3.1.2 Çoklu Doküman Sonuçları

Önerilen sistem kümeleme yaptıktan sonra her küme içinden cümle seçer, bu şekildeki bir yaklaşım çoklu doküman özetleme için de kullanılabilir. Önerilen yöntemi çoklu özetlemede değerlendirmek için DUC 2002'deki 59 ayrı konuda bulunan dokümanlardan yararlanılmıştır. Değerlendirmelerin yapılabilmesi için [18] ve [28] çalışmasında verilen çoklu özet çalışma sonuçları göz önünde bulundurulmuştur.

Çizelge 4.14 Çoklu Doküman Sonuçları ile Kıyaslama

Sistem	ROUGE-1
ClusterCMRW(Agglomerative)	0.385
ClusterHITS (Agglomerative)	0.377
ClusterHITS (Kmeans)	0.376
S 29	0.3264
S 25	0.3056
S 20	0.3047
Baz Çizgisi	0.2932
Önerilen Yöntem	0.3232

Çizelgede verilen çoklu doküman sonuçlarına göre ClusterCMRW ve ClusterHITS yöntemlerinin daha performanslı çalıştığı gözlemlenmiştir. Bunun yanında önerilen yöntem baz çizgisini, sistem 20 ve 25'i %2'lik bir fark ile geçtiği saptanmıştır. Dolayısıyla önerilen sistem sadece tekli doküman üzerinde değil çoklu dokümanlar üzerinde de çalışabileceğini ispatlamıştır.

BÖLÜM 5

SONUÇ ve ÖNERİLER

Bu çalışmada çizge tabanlı özetleme için kullanılan TextRank algoritmasının değişik benzerlik yöntemleri kullanılarak, çıkan özete nasıl katkı sağlayabileceği incelenmiştir. Benzerlik yöntemleri olarak içerik çakışması, NGD, kosinüs ve Jaccard benzerlikleri kullanılmıştır ve sonuçlar DUC 2002 ve CAST kütüphanelerinde analiz edilmiştir. Veri setleri hazırlanışları bakımından farklı olduğu için DUC sonuçları ROUGE-1 ölçütü ve CAST sonuçları keskinlik ve anma ölçütleriyle incelenmiştir. Deneysel çalışmalara göre DUC kütüphanesi kullanıldığında en iyi sonucun içerik çakışmasıyla; CAST kütüphanesi kullanıldığında ise en iyi sonucun NGD kullanılarak bulunduğu gözlemlenmiştir. Bunun yanında benzerlik ölçütü olarak anlamsal benzerlik de kullanılmıştır fakat performans problemlerinden dolayı faydalı netice alınamamıştır. İleriki çalışmalarda daha performanslı bir anlamsal benzerlik üzerinde çalışma yapıp, diğer benzerlik ölçütleriyle kıyaslanması; TextRank algoritmasının daha iyi sonuç verip vermeyeceğini analiz etmek açısından faydalı olacaktır.

Bu çalışma ayrıca hiyerarşik birleştirici kümeleme ve TextRank algoritmasına dayalı seçim gerçekleştiren yeni bir sistem önerilmiştir. Kümeleme yöntemi kullanılarak yapılan çalışmalar genelde cümle seçimini, merkezi cümleyi bularak yaparlar. Fakat bu çalışmada kümelerden cümle seçimi yapabilmek için her kümede kendi içerisinde TextRank algoritması uygulanmıştır. Bu algoritmada elde edilen sıralamaya göre kümelerden cümleler seçilmiştir. DUC 2002 sonuçlarına bakıldığında sadece TextRank algoritması kullanılan sonuçları ortalamada %2 geçtiği görülür. CAST veri setinde ise 4 benzerlik metodundan 2'sini geçtiği gözlemlenmiştir. Bu çalışmada hiyerarşik birleştirici küme

oluşturulması için kosinüs benzerlik yöntemi kullanılmıştır; ileriki çalışmalarda hiyerarşik birleştirici kümeleme yapılırken kullanılan farklı benzerlik yöntemlerinin kümelemeye olan etkisi analiz edilecektir. Buna ilave olarak yapılacak çalışmalarda anlamsal benzerlik yöntemi veya belli kavramların gruplanması kullanılarak da analiz yapılacaktır. Bu şekildeki bir yaklaşım, özete daha farklı kavramların girmesini sağlayacağından sonuçlara etkisi analiz edilecektir.

KAYNAKLAR

[1] Nenkova, A. ve McKeown, K. (2011). “Automatic Summarization”, Foundations and Trends in Information Retrieval, 5(2): 103-233.

[2] Luhn, H. P. (1958). “The automatic creation of literature abstracts”. IBM Journal of Research Development, 2(2):159–165.

[3] Mihalcea, R. ve Tarau, P., (2004). “TextRank: Bringing Order into Texts”, Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing.

[4] Chatterjee, N., Mittal, A. ve Goyal, S., (2012) "Single document extractive text summarization using Genetic Algorithms", Emerging Applications of Information Technology (EAIT), 2012 Third International Conference, doi: 10.1109/EAIT.2012.6407852.

[5] Qazvinian, V., Hassanabadi, L.S. ve Halavati, R., (2008). “Summarising text with a genetic algorithm-based sentence extraction”, International Journal of Knowledge Management Studies, 426-444, 10. 1504/IJKMS.

[6] CAST projesi, http://clg.wlv.ac.uk/projects/CAST/corpus/index.php., 30/03/2014.

[7] DUC konferansı, http://duc.nist.gov/.,30/03/2014.

[8] Pei-ying, Z. ve Cun-he, L. (2009). “Automatic text summarization based on sentences clustering and extraction”, Computer Science and Information Technology, ICCSIT 2009. 2nd IEEE International Conference, 167-170.

[9] Aliguliev, R. ve Aliguliyev, R., (2009) "Evolutionary Algorithm for Extractive Text Summarization," Intelligent Information Management, 1(2):128-138. doi: 10.4236/iim.

[10] Sharan, A., Jain, N. ve Imran,H. (2008). “Automatic Text Summarization: by Extracting Significant Sentences”, In Proceedings of the 6th Annual CISTM Conference.

[11] Binwahlan, M.S., Salim, N. ve Suanmali, L., (2009) "Swarm Based Text Summarization", Computer Science and Information Technology - Spring Conference. IACSITSC '09. 145,150, doi: 10.1109/IACSIT-SC.2009.61.

[12] Ouyang, Y., Li, W., Zhang, R., Li, S. ve Lu, Q., (2012) “A progressive sentence selection strategy for document summarization Information“, Processing & Management doi:10.1016/j.ipm.2012.05.002.

[13] Jin, H., (2000). “Sentence Reduction for Automatic Text Summarization”, ANLC '00 Proceedings of the sixth conference on Applied natural language processing, 310-315.

[14] Uy, N., Q., Anh, P.T., Doan, T.C., ve Hoai, N.X., 2012. “A Study on the Use of Genetic Programming for Automatic Text Summarization”. In Proceedings of the 2012 Fourth International Conference on Knowledge and Systems Engineering (KSE '12). IEEE Computer Society, Washington, DC, USA, 93-98. DOI=10.1109/KSE.2012.10.

[15] Abuobieda, A., Salim, N., Kumar, Y.J., ve Osman, A.H., (2013). “An improved evolutionary algorithm for extractive text summarization”. In Proceedings of the 5th Asian conference on Intelligent Information and Database Systems, DOI=10.1007/978-3-642-36543-0_9.

[16] Plaza, L., Díaz, A., ve Gervás, P., (2011). “A semantic graph-based approach to biomedical summarisation”. Artif. Intell. Med. 53(1):1-14.

[17] Menéndez, H. D., Plaza, L., ve Camacho, D. (2013). “A genetic graph-based clustering approach to biomedical summarization”. In Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics. DOI=10.1145/2479787.2479807.

[18] Mihalcea, R. ve Tarau, P. (2005). “A language independent algorithm for single and multiple document summarization”, Proceedings of IJCNLP’2005.

[19] Garg, N., Favre, B., Riedhammer ve K., Hakkani-Tür, D. (2009). “ClusterRank: A Graph Based Method for Meeting Summarization”.

[20] Alexander, G., (2012). “Computational Linguistics and Intelligent Text Processing” 13th International Conference.

[21] Lin, C.Y., (2004). “ROUGE: A Package for Automatic Evaluation of summaries”, in 'Proc. ACL Workshop on Text Summarization Branches Out'.

[22] Brin S. ve Page L., (1998). “The anatomy of a large-scale hypertextual Web search engine”. In Proceedings of the seventh international conference on World Wide Web 7 (WWW7), Philip H. Enslow, Jr. and Allen Ellis (Eds.). Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 107-117.

[23] Cilibrasi, R. L. ve Vitanyi, P., (2007). "The Google Similarity Distance", IEEE Transactions on Knowledge & Data Engineering, 19(3):370-383.

[24] Madadhain, J., Fisher, D., Smyth, P., White, S. ve Boey, Y. B. (2005). "Analysis and visualization of network data using JUNG". Journal of Statistical Software: 1–25.

[25] Murtagh, F. ve Contreras, P. (2011). “Methods of Hierarchical Clustering”, CoRR,http://arxiv.org/abs/1105.0121.

[26] Aliguliyev, R.M. (2009). “A new sentence similarity measure and sentence based extractive technique for automatic summarization”. Expert System with Applications 7764-7772.

[27] Berker, M. ve Güngör, T. (2012).”Using Genetic Algorithms With Lexical Chains For Automatic Text Summarization”. International Conference on Agents and Artificial Intelligence

[28] Wan, X. Ve Yang, J. (2008). “Multi-Document Summarization Using Cluster-Based Link Analysis”.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Can YALKIN

Doğum Tarihi ve Yeri : 02.01.1987 - Monterey

Yabancı Dili : İngilizce

E-posta : canyalkin@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Bilgisayar Müh.	Yeditepe Üniv.	2010
Lise	Fen-Matematik	Bilfen Anadolu Lisesi	2005

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2013-…	Avea İletişim Hizmetleri A.Ş.	Yazılım Müh.
2010-2013	Yaltes Elektronik A.Ş.	Yazılım Müh.
2010	Cybersoft	Yazılım Müh.

YAYINLARI

Makale

1. Can YALKIN, Emin Erkan KORKMAZ (December, 2012), “A neural network based selection method for genetic algorithms”, Neural Network World – International Journal on Neural and Mass-Parallel Computing and Information Systems