Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 29(2), ss. 115-129, Aralık 2014
Çukurova University Journal of the Faculty of Engineering and Architecture, 29(2), pp.115-129, Aralık 2014

Türkçe Dokümanlardaki Benzerliklerin Tespiti İçin Mevcut Yazılımların Karşılaştırılması ve Türkçe Karakter Kullanımı ile Kök Almanın Etkisinin İncelenmesi

Mümine KAYA[*1], Selma Ayşe ÖZEL[2]

[1]Adana Bilim ve Teknoloji Üniversitesi, Bilgisayar Mühendisliği Bölümü, Adana
[2]Çukurova Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana

Geliş tarihi: 24.10.2014 Kabul tarihi:12.11.2014

Özet

Web ortamındaki bilginin çoğalıp, Internet ve bilgi teknolojilerinin yaygın kullanılması hemen her alanda intihal vakalarının artmasına neden olmuştur. Örneğin, akademik ortamda bazı öğrenciler kendilerine eğitmenleri tarafından verilen ödevler üzerinde çeşitli intihal yöntemlerini uygulamaktadırlar. Bazı öğrenciler başkalarının çalışmasını herhangi bir değişiklik yapmadan ve sahibine atıfta bulunmadan kendi çalışması gibi gösterirken, bazı öğrenciler de diğerlerinin çalışmasını sadece bazı küçük değişiklikler yaparak sunmaktadır. Bu çalışmada amacımız intihal tespit yazılımlarından CopyCatchGold, Sherlock, SIM, WCopyFind, JPlag, YTÜ Kemik Grubu tarafından hazırlanan Metin Eşleştirme Sistemi ve Doküman Benzerliği programları ile kendi kodladığımız Kosinüs, Dice ve Jaccard metin benzerlik ölçütlerinin Türkçe örnek veri kümeleri üzerinde performanslarını karşılaştırmaktır. Buna ek olarak Türkçe karakter ve kelime köklerinin kullanımının intihal tespiti üzerindeki etkisi incelenmiştir. Sonuç olarak, Türkçe karakter kullanımının benzerlik tespitini azalttığı, ancak kelime köklerinin kullanımının ise intihal tespit araçlarının performansını arttırdığı gözlenmiştir.

Anahtar Kelimeler: İntihal, Türkçe, Kök alma, İntihal tespit yazılımları.

A Comparison of Text Similarity Detection Software for Turkish Documents and Investigating the Effects of Stemming and Turkish Character Usage

Abstract

The increase in the amount of available information on the Web and widespread usage of the Internet and information technologies have caused to rise in occurrence of plagiarism in almost everywhere. As an example, in academia some students have performed a variety of plagiarism methods on their assignments given by the instructors. While some students show others' work by their own without making any changes and giving any reference to owner, some other students submit others' studies by making some small changes. In this study, our aim is to compare the performance of plagiarism detection software that

[*] Yazışmaların yapılacağı yazar: Mümine KAYA, *Adana Bilim ve Teknoloji Üniversitesi, Bilgisayar Mühendisliği Bölümü, Adana, mkaya @adanabtu.edu.tr.*

are CopyCatchGold, Sherlock, SIM, WCopyFind, JPlag, two other software that are Text Matching System and Document Similarity developed by YTÜ Kemik Group, as well as our implemented Cosine, Dice, and Jaccard text similarity measures on Turkish sample datasets. In addition, we have investigated the effects of using Turkish character set and Turkish stemmer on plagiarism detection. Consequently, it was observed that using Turkish characters decreases similarity detection, using stemmed words on the other hands, increases the performance of plagiarism detection tools.

Keywords: Plagiarism, Turkish, Stemming, Plagiarism detection softwares.

1. GİRİŞ

Bir kişinin eserini, çalışmasını ya da düşüncesini kendisininmiş gibi gösterme olarak tanımlanan intihal akademik ortamda büyüyen bir problem olmaktadır. İnternet ve bilgisayarların yaygın kullanımı; elektronik ortamdaki metinlerin kolayca kopyalanabilmesi bu problemi arttıran faktörlerdir.

İntihalin kesin bir tanımı olmamakla birlikte birçok yazarın farklı tanımları bulunmaktadır. Orijinal çalışma ya da kaynağa atıfta bulunmadan herhangi bir kaynak belgeden kopyalanan kelimeler olarak da ifade edilebilen intihal [1], bir başka tanıma göre ise başka bir yazarın düşünce ve dilinin taklidi veya kullanımı ve onları kendi özgün eseri olarak sunmaktır [2].

Bilginin ve bilgi kaynaklarının hızla artması, bilginin araştırılması ve bulunabilirliğini etkilemektedir. Bilgiye ulaşımın kolaylaşması aynı bilginin birçok kişi tarafından izinsiz ve atıf verilmeksizin kullanılmasını da arttırmıştır. İntihal, benzerlik ölçümü ve tespiti yapan programlarla günümüzde daha rahat belirlenebilmektedir.

Eğitim ortamlarında intihalin öğrenciler arasında giderek artmasından dolayı verilen ödevlerdeki benzerliklerin tespitini kolaylaştıran yazılımların kullanılmasının faydalı olacağı düşünüldüğünden bu çalışmada intihal tespiti yöntemlerinin karşılaştırılması yapılmıştır.

Araştırmada öncelikle, literatür çalışmaları incelenmiş, intihal sorununa çözüm olacak yöntemler ve yazılımlar belirlenmiş ve bu yazılım ve yöntemler örnek veri kümeleri üzerinde karşılaştırılarak sonuçları bu çalışmada sunulmuştur. Analiz sonucu en başarılı yöntemler ve yazılımlar tespit edilmiştir. Aynı zamanda yapılan testler ile benzerlik hesaplanmasında metinlerde Türkçe karakter kullanımının ve kelimelerin köklerinin alınmasının etkisi ölçülerek, sonucu nasıl etkilediği gözlenmiştir. Makalenin geri kalanı şu şekildedir: Bölüm 2'de Türkçe metin benzerliğinde yapılmış olan çalışmalar özetlenmiştir. Bölüm 3'te mevcut intihal tespit yazılımları ve yöntemlerinden bahsedilmiştir. Bölüm 4'te Araştırma Yöntemi, Bölüm 5'te Bulgular açıklanmış olup, Bölüm 6'da çalışmanın sonucu sunulmuştur.

2. ÖNCEKİ ÇALIŞMALAR

Metin benzerliği hesaplama ve benzer metinleri kümeleme konusunda pek çok çalışma mevcuttur.

Huang [3], internetin kullanımının yaygınlaşmasıyla metin dokümanlarının gün geçtikçe çoğaldığını ve bu nedenle metin dokümanlarının etkili bir biçimde düzenlenmesindeki zorlukları vurgulamıştır. Bu sebeple çalışmasında metin dokümanlarının etkili bir şekilde kümelenmesinde kullanılabilecek uzaklık fonksiyonları ve benzerlik ölçümlerinden Öklit Uzaklığı, Kosinüs Benzerliği, Jaccard Katsayısı, Pearson Korelasyon Katsayısı ve Ortalama Kullback-Leibler Iraksaklığı yöntemlerini karşılaştırmıştır. Kümeleme Algoritması olarak da K-Means algoritmasını seçmiştir. Elde edilen test sonuçlarına göre, Öklit uzaklığı metin kümelemede pek etkili olamamıştır. Pearson Korelasyon Katsayısı ve Ortalama Kullback-Leibler Iraksaklığı yöntemleri kümelemede daha başarılı olup daha dengeli sonuçlar üretirken, Jaccard Katsayısı daha mantıklı

kümeler bulmuştur. Dursun ve Sönmez [4] ise, Türkçe metinlerin benzerliklerinin hesaplanması için yeni bir benzerlik yöntemi geliştirdiklerini belirtmişlerdir. Bu yöntemi geliştirirken de Türkçe metinlerde sık karşılaşılan hata durumları tespit edilmiş ve bu durumlar modellenmiştir. Geliştirdikleri yeni yöntemin test edilmesi için farklı özelliklerdeki kullanıcılardan farklı şekillerde metin girilmesi istenmiş ve bu girilen metinler arasından hatalı olanlar seçilerek test verisi olarak kullanılmıştır. Bu test verisi üzerinde de kendi geliştirdikleri yöntemle Levenshtein Edit Distance Benzerliği ve Jaro-Winkler Benzerliği yöntemleri karşılaştırılmıştır. Elde edilen test sonuçlarına göre geliştirilen yöntemin karşılaştırılan diğer iki yönteme göre daha başarılı olduğu gözlenmiştir.

Amasyalı ve Beken [5], çalışmalarında kelimelerin hangi tür boyutlu uzaylarda temsil edildiği konusu üzerinde durmuştur. Bu amaçla, Türkçe metinlerde geçen kelimeleri anlamsal bir uzayda konumlandırdıkları bir yaklaşım geliştirmişlerdir. Önerdikleri yaklaşımda Harris'in Hipotezi yöntemini esas alıp Türkçe haber metinleri üzerinde uygulamışlardır. Kelimelerin anlamsal uzaydaki konumlarından faydalanarak bu kelimeleri içeren metinlerin konumu bulunmuş ve metinler sınıflandırılmıştır. Elde edilen sonuçlara göre uygulanan yöntem geleneksel kelime grubu (bag-of-words) metotlarına göre daha başarılı bulunmuştur.

Işık ve Çamurcu [6], boyutları her geçen gün artan web sayfalarının içerisinde istenen belgeye erişimi kolaylaştırmak için kullanılan web belgelerini kümeleme konusu üzerinde çalışmıştır. Web belgelerini kümelemede Öklid, Pearson, Kosinüs ve Genişletilmiş Jaccard teknikleri Milliyet Gazetesi ve YahooNews (indirgenmiş) adlı iki ayrı veri kümesi üzerinde test edilmiştir. Kümelemenin değerlendirilmesi için kullanılan saflık, entropi ve ortak bilgi ölçütleri, kümelerin sonucuna uygulanmıştır. Yapılan testler sonucu, benzerlik ölçütleri arasında en iyi performansı Kosinüs ve Genişletilmiş Jaccard benzerlikleri sağlarken, Öklit uzaklığı yüksek hata oranlarına neden olmuştur. Bu sonuçlar doğrultusunda, web belgelerini kümelemede Kosinüs ölçütünün kullanılmasını uygun bulmuşlardır.

3. BENZERLİK YAZILIMLARI VE ÖLÇÜTLERİ

İntihal tespiti için birbirine benzeyen dokümanların karşılaştırılması gerekmektedir. Karşılaştırma sırasında Pencereleme (Winnowing) Algoritması [7], Karp-Rabin Algoritması [8], Hırslı Metin Eşleme Algoritması (Greedy String Tiling Algoritması) [9], Running-Karp-Rabin-Greedy-String-Tiling Algoritması [10], En Uzun Ortak Alt Sıra (The Longest Common Subsequence) Algoritması [11] gibi metin eşleme algoritmalarından ve Kosinüs Benzerliği [12], Öklit Uzaklığı [13], Levenshtein Uzaklığı [14], Dice Katsayısı [15], Jaccard Katsayısı [16], Jaro–Winkler Uzaklığı [17], Tanimoto Benzerliği [18], Hamming Uzaklığı [19], Manhattan Uzaklığı [20], Minkowksi Uzaklığı [21] gibi benzerlik ölçütlerinden faydalanılarak hazırlanan SIM [22], Sherlock [25], WCopyfind [27], CopyCatch Gold [28], JPlag [29], Metin Eşleştirme Sistemi [30] ve Doküman Benzerliği [31] yazılımlarından yararlanılmıştır.

3.1. SIM

SIM [22]; C, Java, Pascal, Modula-2, Lisp, Miranda ve metin dosyaları için kullanılabilen bir benzerlik tespit programıdır. 1989 yılında Grune tarafından Amsterdam'daki Vrije Üniversitesi'nde geliştirilmiştir. Huntenjens ise SIM'in çıktısını alıp, intihal raporuna dönüştürmüştür [23].

SIM, büyük yazılım projelerindeki, program metinlerindeki, kabuk betiklerindeki (shell script) potansiyel çoğaltılmış kod parçalarını tespit etmek ve eğitim alanındaki yazılım projelerinde intihali tespit etmek için kullanılır. SIM, hem Windows hem de UNIX/Linux üzerinde çalışabilmektedir [22]. Konsoldan parametre alarak çalışmaktadır. ftp://ftp.cs.vu.nl/pub/dick/similarity_tester/ adresinden indirilebilen ve ücretsiz bir yazılım olan SIM isim değişikliklerini ve program bloklarının yer değişimini fark edebilmekte, boşlukları ise dikkate almamaktadır [24].

Karşılaştırma sonuçlarını metin dosyası gibi dosya türlerine aktarabilme özelliğine sahiptir [22]. Bir eşik değeri belirlenerek istenilen orandan daha yüksekteki benzerlikleri görme olanağı sağlamaktadır. İstenildiği ya da ihtiyaç duyulduğu takdirde benzerlik tespit edilen satırlar gösterilerek ya da sadece hangi satırlarda olduğu özetlenerek sonuçlar gösterilebilir. SIM'in benzerlikleri tespit etmek için kullandığı işlem öncelikle kaynak kodu sembol (token) haline getirmektir. Sonra kıyaslanması gerekeni ve yeni gönderilmiş dosyalar arasındaki en iyi eşleşmeleri tespit edebilen ileri referans tablosu oluşturulur [23]. SIM artık aktif olarak sürdürülüp desteklenmese de, kaynak kodu kamuya açıktır.

3.2. Sherlock

Sherlock [25], metin dokümanları arasındaki benzerlikleri bulan bir programdır. Programın orijinal versiyonu Pike tarafından geliştirilmiştir. Sherlock makale, program kaynak kodları ve dijital platformlardaki ödevler gibi metin dosyalarını karşılaştırabilir. Sherlock hem Unix/Linux üzerinde ve hem de Windows üzerinde çalışır. Sherlock, hem tar hem de zip dosyalarını kabul etse de zipli dosyalarda daha başarılıdır fakat zipli arşivlerde çalışması için önce arşivden çıkartılması gerekmektedir Sherlock, komut satırı programıdır, bir grafiksel kullanıcı ara yüzü yoktur [25]. Sherlock yapı temelli yaklaşım kullanır [26]. Metinlerin benzer parçalarını bulmak için dijital imzayı kullanır. Dijital imza birtakım kelimeleri bit serilerine dönüştüren ve bu bitleri de birleştirerek numaraya dönüştüren bir sayı formudur [25].

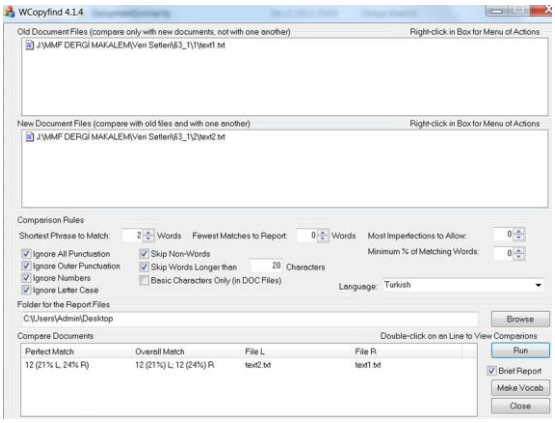
Sherlock uygulamasından önce sig ve comp adında iki adet program vardı. Sig programı dijital imzaları üretip, bir dosya içinde depo ederken, comp programı da imza dosyalarını kıyaslayıp benzerlikleri rapor etmekteydi. Disk alanı ve yönetiminden dolayı bu iki program Sherlock adı altında bir programda birleştirildi [25].

3.3. Wcopyfind

WcopyFind [27], 2002 yılında Virginia Üniversitesi tarafından geliştirilmiş bir yazılımdır. Dokümanları yüklemek, çırpı (hash) yapısına almak ve karşılaştırmak için kullanılan açık kaynak kodlu ve ücretsiz WCopyfind programı, Windows-stili grafik kullanıcı ara yüzü sağlamaktadır. Bilgisayarda bulunan dokümanları kullanarak karşılaştırma yapmaktadır. Çevrimiçi olarak edinilen ve yararlanılan çalışmalardan alıntı yapıldıysa ve kaynak gösterme unutulduysa bu konuda yardımcı olabilecek bir programdır.

İki aşamada işlemlerini gerçekleştirir. İlk aşamada bütün dokümanları yükler ve çırpı kodlarını oluşturur. İkinci aşamada çırpı kodu oluşturulmuş doküman çiftlerini karşılaştırır ve rapor oluşturur. Şekil 1'de de gösterildiği gibi programın ara yüzünde kullanıcıya bazı ayarlar sunulmaktadır. Karşılaştırma ayarlarında büyük/küçük harf, noktalama işaretleri, sayılar ve diğer karakterlerin kaldırılması, eşleşecek en kısa sözcük grubu, eşleşen kelimelerin minimum yüzdesi ve dil ayar durumu mevcuttur. Türkçe dilini desteklemektedir.

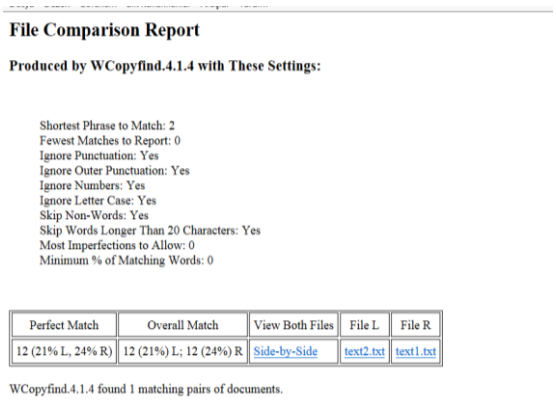
Her doküman bir kez okunduktan sonra 32-bit çırpı kodlarına dönüştürülür. Bu 32-bit çırpı kodları daha sonra doküman-sıralı listelere dönüştürülür. Karşılaştırma işlemi bu çırpı kodlar üzerinden gerçekleştirilir. Eşleştirme sırasında 4 karakterden kısa kelimeler incelenmemektedir.



Şekil 1. WCopyfind kullanıcı ara yüzü

Karşılaştırma işlemi bittikten sonra Şekil 2'deki gibi html uzantılı bir rapor oluşturulmaktadır. Program oluşturulan raporda iki dosyayı aynı sayfa

içerisinde yan yana görmeyi mümkün kılmaktadır. Aynı zamanda metinler arasında ortak sözcükleri de .txt dosyası şeklinde kullanıcıya rapor etmektedir.



File Comparison Report

Produced by WCopyfind.4.1.4 with These Settings:

Shortest Phrase to Match: 2
Fewest Matches to Report: 0
Ignore Punctuation: Yes
Ignore Outer Punctuation: Yes
Ignore Numbers: Yes
Ignore Letter Case: Yes
Skip Non-Words: Yes
Skip Words Longer Than 20 Characters: Yes
Most Imperfections to Allow: 0
Minimum % of Matching Words: 0

Perfect Match	Overall Match	View Both Files	File L	File R
12 (21% L, 24% R)	12 (21%) L; 12 (24%) R	Side-by-Side	text2.txt	text1.txt

WCopyfind.4.1.4 found 1 matching pairs of documents.

Şekil 2. WCopyfind sonuç sayfası

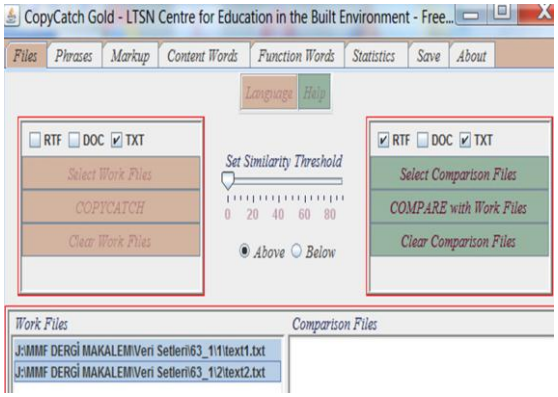
3.4. Copycatch Gold

Uzun yıllardan beri eğitim enstitülerinde kampüs ve bölüm bazlı kullanılan CopyCatch Gold [28] yazılımı gün geçtikçe ticari kullanıcı sayısında artış gösteren bir yazılım olmaktadır.

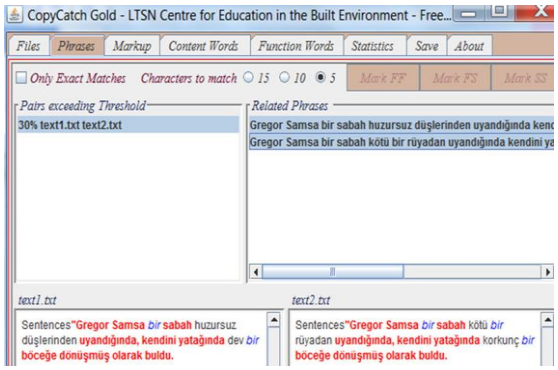
CopyCatch Gold yazılımı on yılın üstünde bir zamandır bireysel üniversite eğitmenleri tarafından intihal ve hileleri tespit etmek amaçlı öğrenci ödevlerini izlemek için kullanılmaktadır. Hızlı, ölçeklenebilir, kendine ait bir ara yüzü olan, çoklu-işlem kapasiteli, başlangıçta Java dilinde yazılmış ancak 2012 yılından itibaren Java, Hadoop, Jena (Rdf ve Sparql) ve Chapel ile geliştirilmiş, birçok dili destekleyen çoklu-platform yazılımıdır. Rtf, doc, htm ve txt uzantılı dosyalar üzerinde çalışmaktadır. Şekil 3'te de gösterildiği gibi ara yüzünde dil ve benzerlik eşik değerinin de bulunduğu çeşitli ayarları mevcuttur.

Büyük boyutlu dosyalarla hızlı bir şekilde çalışabilmekte ve geri bildirimde bulunabilmektedir. Sınıf ya da bire-bir kullanım için de uygun bir yazılımdır.

Sonuçların .html ve .rtf olmak üzere iki çeşitte kaydedilmesine olanak sağlamaktadır. Aynı zamanda Şekil 4'te de gösterildiği gibi uygulama ara yüzünde de sonuçlar, dokümanlar yan yana olacak şekilde gösterilmektedir.



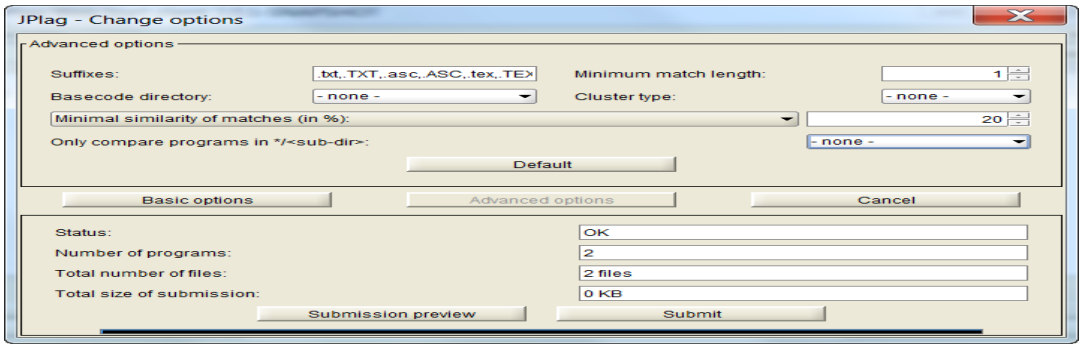
Şekil 3. CopyCatch gold kullanıcı ara yüzü



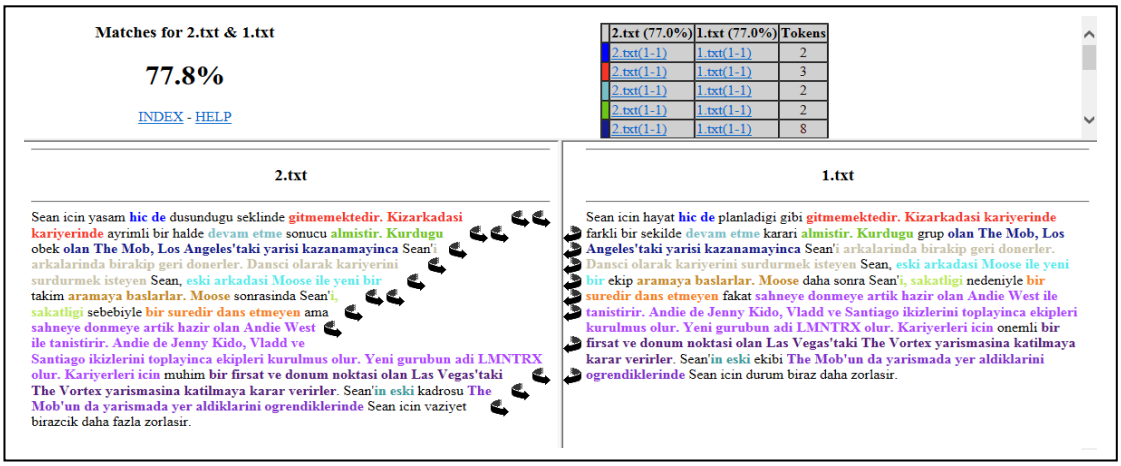
Şekil 4. CopyCatch gold sonuç sayfası

3.5. Jplag

Karlsruhe Üniversitesi ve Michael Philippsen Üniversitesi'nden Prechelt ve Malpohl tarafından Java dilinde yazılmış, açık-kaynak kodlu JPlag uygulaması özellikle kaynak kodlardaki intihallerin tespiti için hazırlanmış bir yazılımdır [11]. Sadece C, C++, Java, C#, Scheme gibi programlama dilleriyle yazılmış olan kaynak kodlar üzerinde değil doğal dillerde de benzerlik tespiti yapabilen JPlag, öğrenci ödevlerindeki benzerliklerin bulunmasında oldukça başarılıdır.



Şekil 5. JPlag kullanıcı ara yüzü



Şekil 6. JPlag sonuç sayfası

JPlag verilen dokümanlar kümesi içerisinde benzer doküman çiftlerini bulabilen bir sistemdir [29].

JPlag bir web servisi olarak çalışmaktadır. Ücretsiz bir yazılım olan JPlag'ı kullanmak için e-mail yoluyla hesap açtırılması gerekmektedir. Gönderilecek olan kullanıcı adı ve şifreyle bu web servisine her an ulaşılmaktadır. JPlag, sunucuya gönderilen her bir kaynak kod için ait olduğu programlama dilinin gramerini ya da her bir metin parçası için metin türünün gramerini dikkate alarak inceler ve simge dizilerine dönüştürür. Bu simge dizileri de Wise tarafından önerilen Greedy String Tiling Algoritması [10] kullanılarak birbirleriyle karşılaştırılır ve benzerlik ölçümü hesaplanır [24].

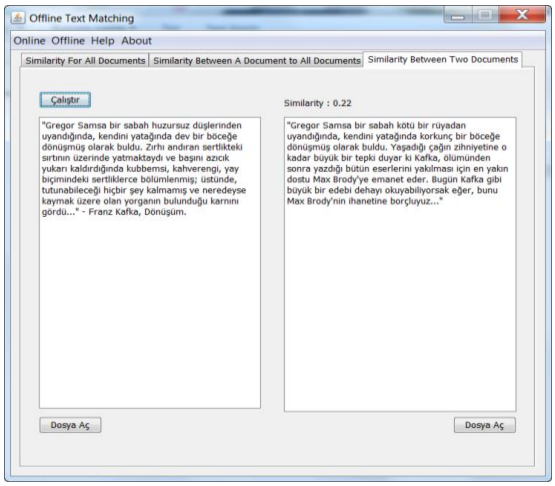
JPlag'ın Şekil 5'te gösterildiği gibi kendine ait bir grafik ara yüzü vardır. Kendi sunucusunda çevrimiçi olarak karşılaştırma yapmaktadır. Karşılaştırılacak dosyalar JPlag'ın sunucusuna gönderilmektedir. Sonuçlar ise kullanıcının bilgisayarına HTML sayfası olarak gelmektedir ve kullanıcıya bu şekilde sunulmaktadır. Karşılaştırılan kaynak kodlar arasındaki benzerlik yüzde olarak verilmekte ve benzerlik bulunan kısımlar Şekil 6'daki gibi yan yana listelenmektedir [24].

3.6. Metin Eşleme Sistemi

Metin Eşleme Sistemi [30], Yıldız Teknik

Üniversitesi Kemik Grubu projesi olarak 2009 yılında Aydoğan ve Diri tarafından hazırlanmış bir projedir. Türkçe dokümanların (.txt, .pdf, .doc, .html formatlarındaki) birbirlerine benzerliklerini bulan bir yazılımdır. Java dilinde Windows ortamında yazılmıştır. Veritabanı olarak JavaDB tercih edilmiştir. Yazılım "Çevrimdışı" ve "Çevrimiçi" olmak üzere iki modda çalışmaktadır. Çevrimiçi mod, genel olarak arşivlemede ve arama motorlarında kullanılarak zaman ve kaynaktan tasarruf edilmesini sağlar. Arşivlemede aranılan dokümanlara daha hızlı ulaşılmasını sağladığı gibi arama motorlarında da kullanılarak birçok doküman yerine bunların anahtar sözcüklerinde arama yapar. Şekil 7'de de gösterildiği gibi çevrimdışı mod, "Bütün Dokümanlar İçin Benzerlik", "Bir Dokümanın Tüm Dokümanlara Olan Benzerliği" ve "İki Doküman Arasındaki Benzerlik" isimli üç adet sekmeden oluşmaktadır.

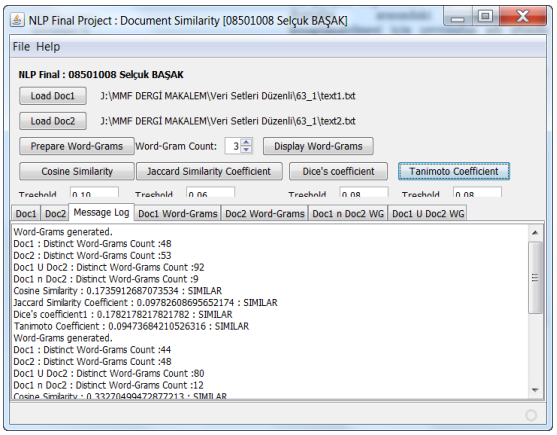
Bu çalışma kapsamında bilgisayarda bulunan dosyalar arasındaki benzerliklerin hesaplanabilmesi için çevrimdışı adı altında gözüken arşivleme özelliği kullanılmıştır.



Şekil 7. Metin eşleme sistemi kullanıcı ara yüzü

3.7. Doküman Benzerliği

Doküman Benzerliği, Yıldız Teknik Üniversitesi Kemik Grubu projesi olarak Başak [31] tarafından hazırlanmış, iki Türkçe doküman arasındaki benzerliğin Jacard, Cosine, Dice ve Tanimoto benzerlikleri kullanılarak ölçülmesini sağlayan bir yazılımdır. Java platformunda Netbeans 6.7.1 ile geliştirilmiştir. Bu uygulamada zemberek kütüphanesi kullanılarak dokümanda geçen kelimelerin kökleri elde edilmiştir. Zemberek tarafından kökü bulunamayan kelimeler bütün olarak kullanılmıştır. Özellik olarak 1, 2, 3 ve 4'lü word-gram'lar kullanılmıştır. Programda word gram uzunluğu Şekil 8'deki gibi interaktif olarak değiştirilebilmektedir.



Şekil 8. Doküman benzerliği kullanıcı ara yüzü

3.8. Kosinüs Benzerliği

Kosinüs Benzerliği, iki doküman arasındaki benzerliği karşılaştırmakta kullanılır. "n" boyutlu iki vektör arasındaki açının bulunmasıyla elde edilir [32]. Dokümanların benzerlikleri vektörler arasında kalan açının kosinüsü ile hesaplanır. Metin eşleştirme için, öznitelik vektörleri A ve B genellikle belgelerin terim-frekans vektörleridir. A_i n boyutlu A vektörünün, B_i ise n boyutlu B vektörünün i. elemanlarıdır. Doküman benzerliği (S) hesaplamasında A ve B vektörlerinin elemanları dokümanlarda geçen terimlerin frekanslarıdır. Bir dokümanda bulunan diğerinde bulunmayan özellik için bulunmayan dokümanda o özellik 0 olarak alınır. Kosinüs Benzerliği, karşılaştırma sırasında belge uzunluğunu normalizasyon yöntemi olarak kullanır. Niteliklerin iki vektörü, A ve B, göz önüne alındığında kosinüs benzerliği, Eşitlik 1'deki gibi

bir nokta çarpımı ve vektör büyüklüğü kullanılarak temsil edilebilir:

$$S = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2 \times \sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

Sonuç benzerliği, tam olarak aynı olan belgeler için 1, tamamen birbiri ile ilişkisiz olanlar için ise 0 olacaktır.

3.9. Jaccard Benzerliği

Jaccard Benzerlik Katsayısı, kümeler arasındaki benzerliği istatistiksel olarak değerlendirir. Jaccard Benzerliği, iki dokümandaki kelimelerin kesişiminin iki dokümandaki kelime sayısının birleşimine bölünmesi ile elde edilir [33]. Eşitlik 2'de Jaccard Benzerlik formülü yer almaktadır.

$$S = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sum_{i=1}^{n} (A_i)^2 + \sum_{i=1}^{n} (B_i)^2 - \sum_{i=1}^{n} A_i \times B_i} \quad (2)$$

3.10. Dice Benzerliği

Kelimeler arasındaki mesafenin ölçülmesi için de kullanılan benzerlik ölçümlerinden biri olan Dice Katsayısı, Jaccard Benzerliği ile ilişkilidir [34] ve Eşitlik 3'teki gibi hesaplanmaktadır.

$$S = \frac{2 | A \cdot B |}{| A |^2 + | B |^2} = \frac{2 \sum_{i=1}^{n} A_i \times B_i}{\sum_{i=1}^{n} (A_i)^2 + \sum_{i=1}^{n} (B_i)^2} \quad (3)$$

4. ARAŞTIRMANIN YÖNTEMİ

Çalışmayla ilgili analizlerin yapılabilmesi için öncelikle internetten birbiriyle ilgili sitelerden veri kümesi hazırlanmıştır. Franz Kafka'nın "Dönüşüm" [35] adlı kitabının ilk sayfasından ve arka kapak sayfasından alınan iki küçük metin Çizelge 1'deki gibi ilk veri kümesini oluşturmaktadır.

Metin1.txt ve Metin2.txt'deki Türkçe karakterlerin Latin karakterlere dönüştürülmesiyle ikinci veri kümesi, bu metinlerdeki Türkçe kelimelerin köklerinin alınmasıyla üçüncü veri kümesi oluşturulmuştur. Kelime köklerini alma işlemi Zemberek [36] yazılımıyla gerçekleştirildi. Hem Türkçe karakterlerin arındırılıp hem de kelimelerin köklerinin alınmasıyla dördüncü veri kümesi oluşturulmuştur. Bu veri kümesinde Latin karakterlere dönüştürülen kelimeler Zemberek yazılımıyla köklerine ayrılmıştır. Kökü bulunamayan kelimeler Zemberek yazılımının "Kelime Öner" metoduyla düzeltilip, daha sonra tekrar çözümlenmiştir.

"Step Up All In" adlı filmin tanıtım özetleri iki ayrı film web sitesinden [37, 38] alınarak iki küçük metin Çizelge 2'deki gibi beşinci veri kümesini oluşturmaktadır. İlk veri kümesine uygulanan işlemleri aynı sırayla Step Up All In adlı film özetlerinden oluşan veri kümesine de uygulayarak altıncı, yedinci ve sekizinci veri kümeleri de oluşturulmuştur. Veri kümeleri hazır edildikten sonra bu çalışma kapsamında karşılaştırılması yapılacak olan yazılımlarla – CopyCatchGold, Sherlock, SIM, WCopyFind, JPlag, YTÜ Metin Eşleştirme Sistemi, YTÜ Doküman Benzerliği ve bizim kodladığımız Kosinüs, Dice ve Jaccard Benzerlikleri - analiz edilmiştir.

5. BULGULAR

Bu çalışma kapsamında verilen örnek veri kümeleriyle yazılımların ve yöntemlerin karşılaştırılması yapılmıştır. M1, birinci metini; M2 ikinci metini; (T), Türkçe karakterlerin Latin karakterlere dönüşümü yapıldığını; (K) kelimelerin köklerinin alındığını ifade etmektedir.

Çizelge 1'de de görüldüğü üzere Metin 1 ve Metin 2'den oluşan veri kümeleri incelendiğinde genel olarak; en başarısız yazılımların Sherlock, JPlag, Metin Eşleme Sistemi, Doküman Benzerliği- Jaccard Benzerliği, Doküman Benzerliği- Tanimoto ve kendi kodladığımız

Jaccard Benzerliği olduğu belirlenmiştir. CopyCatch Gold, Wcopyfind ve Kosinüs Benzerliğinin ise daha uygun sonuçlar ürettikleri gözlenmiştir. Çizelge 2'de de gösterildiği gibi Türkçe karakter kullanılmayan ikinci veri kümesinde sonucun, hiçbir değişiklik yapılmadan aynen alınan birinci veri kümesine göre biraz daha iyileştirildiği gözlenmiştir. Sonuçlar ilk veri kümesindeki sonuçlara göre daha başarılı ve gerçekçi bulunmuştur. Kelimelerin köklerinin kullanıldığı üçüncü veri kümesi ile yapılan testlerden elde edilen sonuçlara göre, kelime köklerinin alınmasının benzerlik tespitinde pozitif etkisi olduğu gözlenmiştir. Metinlerin ham hallerinin kullanıldığı birinci veri kümesine göre biraz daha iyileştirildiği gözlenmiştir. Sonuçlar ilk veri kümesindeki sonuçlara göre daha başarılı ve gerçekçi bulunmuştur.

Kelimelerin köklerinin kullanıldığı üçüncü veri kümesi ile yapılan testlerden elde edilen sonuçlara göre, kelime köklerinin alınmasının benzerlik tespitinde pozitif etkisi olduğu gözlenmiştir. Metinlerin ham hallerinin kullanıldığı birinci veri kümesine göre Çizelge 3'te de görüldüğü üzere daha başarılı sonuçlar üretilmiştir. Çizelge 4'te de görüldüğü üzere, kelimelerdeki Türkçe karakterlerin Latin karakterlere

Çizelge 1. Metin1 ve Metin2

Metin1.txt	**Metin2.txt**
"Gregor Samsa bir sabah huzursuz düşlerinden uyandığında, kendini yatağında dev bir böceğe dönüşmüş olarak buldu. Zırhı andıran sertlikteki sırtının üzerinde yatmaktaydı ve başını azıcık yukarı kaldırdığında kubbemsi, kahverengi, yay biçimindeki sertliklerce bölümlenmiş; üstünde, tutunabileceği hiçbir şey kalmamış ve neredeyse kaymak üzere olan yorganın bulunduğu karnını gördü..." - Franz Kafka, Dönüşüm.	"Gregor Samsa bir sabah kötü bir rüyadan uyandığında, kendini yatağında korkunç bir böceğe dönüşmüş olarak buldu. Yaşadığı çağın zihniyetine o kadar büyük bir tepki duyar ki Kafka, ölümünden sonra yazdığı bütün eserlerini yakılması için en yakın dostu Max Brody'ye emanet eder. Bugün Kafka gibi büyük bir edebi dehayı okuyabiliyorsak eğer, bunu Max Brody'nin ihanetine borçluyuz..."

Çizelge 2. Metin3 ve Metin4

Metin3.txt	**Metin4.txt**
Sean için hayat hiç de planladığı gibi gitmemektedir. Kızarkadaşı kariyerinde farklı bir şekilde devam etme kararı almıştır. Kurduğu grup olan The Mob, Los Angeles'taki yarısı kazanamayınca Sean'ı arkalarında bırakıp geri dönerler. Dansçı olarak kariyerini sürdürmek isteyen Sean, eski arkadaşı Moose ile yeni bir ekip aramaya başlarlar. Moose daha sonra Sean'ı, sakatlığı nedeniyle bir süredir dans etmeyen fakat sahneye dönmeye artık hazır olan Andie West ile tanıştırır. Andie de Jenny Kido, Vladd ve Santiago ikizlerini toplayınca ekipleri kurulmuş olur. Yeni gurubun adı LMNTRX olur. Kariyerleri için önemli bir fırsat ve dönüm noktası olan Las Vegas'taki The Vortex yarışmasına katılmaya karar verirler. Sean'ın eski ekibi The Mob'un da yarışmada yer aldıklarını öğrendiklerinde Sean için durum biraz daha zorlaşır.	Sean için yaşam hiç de düşündüğü şeklinde gitmemektedir. Kızarkadaşı kariyerinde ayrımlı bir halde devam etme sonucu almıştır. Kurduğu öbek olan The Mob, Los Angeles'taki yarısı kazanamayınca Sean'ı arkalarında bırakıp geri dönerler. Dansçı olarak kariyerini sürdürmek isteyen Sean, eski arkadaşı Moose ile yeni bir takım aramaya başlarlar. Moose sonrasında Sean'ı, sakatlığı sebebiyle bir süredir dans etmeyen ama sahneye dönmeye artık hazır olan Andie West ile tanıştırır. Andie de Jenny Kido, Vladd ve Santiago ikizlerini toplayınca ekipleri kurulmuş olur. Yeni gurubun adı LMNTRX olur. Kariyerleri için mühim bir fırsat ve dönüm noktası olan Las Vegas'taki The Vortex yarışmasına katılmaya karar verirler. Sean'ın eski kadrosu The Mob'un da yarışmada yer aldıklarını öğrendiklerinde Sean için vaziyet birazcık daha fazla zorlaşır.

Çizelge 3. Yazılımların birinci veri kümesi için karşılaştırma sonuçları (% benzerlik)

Yöntem	M1-M2	M2-M1	M1-M1
CopyCatch Gold	26	-	100
WCopyfind	22	20	100
Sim	22	26	100
Sherlock	0	-	100
JPlag	9,4	-	97,9
Metin Eşleme Sistemi	17	-	100
Doküman Benzerliği - Cosine	31,30	-	100
Doküman Benzerliği - Jaccard	12,35	-	100
Doküman Benzerliği - Dice	21,98	-	100
Doküman Benzerliği - Tanimoto	18,33	-	100
Kosinüs	29,09	-	100
Jaccard	16,67	-	100
Dice	28,57	-	100

Çizelge 4. Yazılımların ikinci veri kümesi için karşılaştırma sonuçları (% benzerlik)

Yöntem	M1(T)-M2(T)	M2(T)-M1(T)	M1(T)-M1(T)
CopyCatch Gold	28	-	100
WCopyfind	26	21	100
Sim	26	27	100
Sherlock	0	-	100
JPlag	9,9	-	97,8
Metin Eşleme Sistemi	18	-	100
Doküman Benzerliği - Cosine	31,75	-	100
Doküman Benzerliği - Jaccard	13,92	-	100
Doküman Benzerliği - Dice	24,44	-	100
Doküman Benzerliği - Tanimoto	18,42	-	100
Kosinüs	31,43	-	100
Jaccard	18,02	-	100
Dice	30,53	-	100

Çizelge 5. Yazılımların üçüncü veri kümesi için karşılaştırma sonuçları (% Benzerlik)

Yöntem	M1(K)-M2(K)	M2(K)-M1(K)	M1(K)-M1(K)
CopyCatch Gold	30	-	100
WCopyfind	28	25	100
Sim	30	30	100
Sherlock	0	-	0
JPlag	10,9	-	98,5
Metin Eşleme Sistemi	25	-	100
Doküman Benzerliği - Cosine	34,72	-	100
Doküman Benzerliği - Jaccard	15	-	100
Doküman Benzerliği - Dice	26,09	-	100
Doküman Benzerliği - Tanimoto	20,69	-	100
Kosinüs	34,72	-	100
Jaccard	20,69	-	100
Dice	34,29	-	100

Çizelge 6. Yazılımların dördüncü veri kümesi için karşılaştırma sonuçları (% Benzerlik)

Yöntem	M1(T,K)-M2(T,K)	M2(T,K)-M1(T,K)	M1(T,K)-M1(T,K)
CopyCatch Gold	28	-	100
WCopyfind	28	23	100
Sim	28	29	100
Sherlock	33	-	100
JPlag	12,6	-	98,3
Metin Eşleme Sistemi	12	-	100
Doküman Benzerliği - Cosine	32,81	-	100
Doküman Benzerliği - Jaccard	14,67	-	100
Doküman Benzerliği - Dice	25,58	-	100
Doküman Benzerliği - Tanimoto	18,85	-	100
Kosinüs	32,81	-	100
Jaccard	18,85	-	100
Dice	31,72	-	100

dönüştürülüp, kelime köklerinin kullanıldığı dördüncü veri kümesi, metinlerin ham halinin kullanıldığı birinci veri kümesine ve sadece Türkçe karakter dönüşümünün yapıldığı ikinci veri kümesine göre daha iyi bir sonuç verse de sadece kelime köklerinin kullanıldığı üçüncü veri kümesine göre daha düşük bir sonuç vermiştir. Türkçe karakter kullanımında hatalar olan "Step Up All In" metinlerinden oluşan veri kümelerinde Çizelge 7'de de görüldüğü gibi en başarılı yazılımlar Sim, WCopyfind, CopyCatch Gold, Metin Eşleme Sistemi, Doküman Benzerliği – Cosine Benzerliği, kendi kodladığımız Kosinüs ve Dice Benzerlikleri olmuştur. En başarısız olanlar ise Kafka veri kümelerinde olduğu gibi Sherlock ve JPlag yazılımları olmuştur.

Çizelge 8'de de gösterildiği gibi Latin karakter dönüşümü yapılan altıncı veri kümesinde sonucun, hiçbir değişiklik yapılmadan aynen alınan beşinci veri kümesine göre daha iyileştirildiği gözlenmiştir. Sonuçlar beşinci veri kümesindeki sonuçlara göre daha başarılı ve gerçekçi bulunmuştur. Kelimelerin köklerinin kullanıldığı yedinci veri kümesi ile yapılan testlerden elde

edilen sonuçlara göre, kelime köklerinin alınmasının benzerlik tespitinde etkisi olduğu gözlenmiştir. Metinlerin ham hallerinin kullanıldığı beşinci veri kümesine göre Çizelge 9'da da görüldüğü üzere daha başarılı sonuçlar üretilmiştir. Çizelge 10'da da görüldüğü üzere, kelimelerdeki Türkçe karakterlerin dönüştürülüp, kelime köklerinin kullanıldığı sekizinci veri kümesi, metinlerin ham halinin kullanıldığı beşinci veri kümesine ve sadece

Çizelge 7. Yazılımların beşinci veri kümesi için karşılaştırma sonuçları (% benzerlik)

Yöntem	M3-M4	M4-M3	M3-M3
CopyCatch Gold	83	-	100
WCopyfind	84	84	100
Sim	85	84	100
Sherlock	60	-	100
JPlag	52,4	-	95
Metin Eşleme Sistemi	86	-	100
Doküman Benzerliği - Cosine	90,33	-	100
Doküman Benzerliği - Jaccard	68,58	-	100
Doküman Benzerliği - Dice	81,36	-	100
Doküman Benzerliği - Tanimoto	82,30	-	100
Kosinüs	88,35	-	100
Jaccard	79,12	-	100
Dice	88,34	-	100

Çizelge 8. Yazılımların altıncı veri kümesi için karşılaştırma sonuçları (% Benzerlik)

Yöntem	M3(T)-M4(T)	M4(T)-M3(T)	M3(T)-M3(T)
CopyCatch Gold	85	-	100
WCopyfind	87	87	100
Sim	87	86	100
Sherlock	100	-	100
JPlag	61,4	-	95
Metin Eşleme Sistemi	84	-	100
Doküman Benzerliği - Cosine	91,68	-	100
Doküman Benzerliği - Jaccard	72,12	-	100
Doküman Benzerliği - Dice	83,80	-	100
Doküman Benzerliği - Tanimoto	84,62	-	100
Kosinüs	90,19	-	100
Jaccard	82,12	-	100
Dice	90,18	-	100

Çizelge 9. Yazılımların yedinci veri kümesi için karşılaştırma sonuçları (% Benzerlik)

Yöntem	M3(K)-M4(K)	M4(K)-M3(K)	M3(K)-M3(K)
CopyCatch Gold	88	-	100
WCopyfind	88	88	100
Sim	91	87	100
Sherlock	100	-	100
JPlag	73,1	-	99,3
Metin Eşleme Sistemi	90	-	100
Doküman Benzerliği - Cosine	93,27	-	100
Doküman Benzerliği - Jaccard	75	-	100
Doküman Benzerliği - Dice	85,71	-	100
Doküman Benzerliği - Tanimoto	87,34	-	100
Kosinüs	93,02	-	100
Jaccard	86,90	-	100
Dice	92,99	-	100

Çizelge 10. Yazılımların sekizinci veri kümesi için karşılaştırma sonuçları (% Benzerlik)

Yöntem	M3(T,K)-M4(T,K)	M4(T,K)-M3(T,K)	M3(T,K)-M3(T,K)
CopyCatch Gold	86	-	100
WCopyfind	87		100
Sim	90		100
Sherlock	100	-	100
JPlag	62,5	-	95,5
Metin Eşleme Sistemi	89	-	100
Doküman Benzerliği - Cosine	93,52	-	100
Doküman Benzerliği - Jaccard	72,83	-	100
Doküman Benzerliği - Dice	84,27	-	100
Doküman Benzerliği - Tanimoto	87,83	-	100
Kosinüs	93,52	-	100
Jaccard	87,83	-	100
Dice	93,52	-	100

Türkçe karakter dönüşümünün yapıldığı altıncı veri kümesine göre daha iyi bir sonuç verse de sadece kelime köklerinin kullanıldığı yedinci veri kümesine göre daha düşük bir sonuç vermiştir.

6. SONUÇ ve ÖNERİLER

Tablolardaki sonuçlardan da görüldüğü üzere Türkçe karakterlerin Latin karakterlere dönüşümü benzerlik hesaplamasını biraz iyileştirirken, kelimelerin köklerinin kullanılması benzerliği olumlu yönde daha da artırmaktadır. Sonuçların daha başarılı ve gerçekçi olmasını sağlamaktadır. Ancak kelimelerin önce Türkçe karakterlerden Latin karakterlere dönüşümleri sağlanıp, ardından kökleri alındığında sadece köklerinin alınmasına göre benzerlik oranları düşmektedir. Bu çalışmayla Türkçe dokümanlar için metinlerin benzerlik tespitinde kelime köklerinin kullanımının etkisinin çok büyük olduğu sonucuna varılmıştır.

7. TEŞEKKÜR

Bu çalışmayı 2211-C Öncelikli Alanlar Bursu

kapsamında destekleyen TÜBİTAK'a ve MMFD08 Proje Numarası ile destekleyen Çukurova Üniversitesi BAP Birimi'ne teşekkür ederiz.

8. KAYNAKLAR

1. Honor Council, 2014. http://orgs.odu.edu/hc/pages/What_is_the_Honor_Council.shtml

2. Hacker, D., (2007), A Writer's Reference, 6th ed., pp. 344-347, 418-421.

3. Huang, A., 2008. Similarity Measures for Text Document Clustering, in New Zealand Computer Science Research Student Conference - Proceedings of NZCSRSC, pp. 49-56.

4. Dursun, B., Sönmez, A. C., 2008. Türkçe Metin Benzerlik Hesaplamasi için Yeni Bir Yöntem, Signal Processing, Communication and Applications Conference, SIU 2008, IEEE 16th, DOI:10.1109/SIU.2008.4632581, pp. 1 - 4.

5. Amasyalı, F., Beken, A., 2009. Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi ve Metin Sınıflandırmada Kullanılması, IEEE Signal Processing and Communications Applications Conference, SİU-2009.

6. Işık, M., Çamurcu, A. Y., 2008. Web Belgeleri Kümelemede Benzerlik ve Uzaklık Ölçütleri Başarılarının Karşılaştırılması, Marmara Üniversitesi Fen Bilimleri Dergisi, 20, 35-49.

7. Schleimer, S., Wilkerson, D. S., Aiken, A., 2003. Winnowing: Local Algorithms for Document Fingerprinting, in Proceedings ACM SIGMOD International Conference on Management of Data, pp. 76-85.

8. Karp R.M., Rabin M.O., 1987. Efficient Randomized Pattern-Matching Algorithms, IBM Journal of Research and Development - IBM J. Res. Dev. 31(2):249-260.

9. Wise, M. J., 1993. String Similarity via Greedy String Tiling and Running Karp-Rabin Matching, ftp://ftp.cs.su.oz.au/michaelw/doc/RKR_GST.ps, Dept. of CS, University of Sydney, December 1993.

10. Wise, M. J., 1993. Running KarpRabin Matching and Greedy String Tiling, Technical Report Number 463, Dept. of CS, University of Sydney, March 1993.

11. Zeidman, R. M., 2010. Detecting Plagiarism in Computer Source Code, United States Patent Application 20100325614 A1.

12. Cosine Similarity, (2014), http://en.wikipedia.org/wiki/Cosine_similarity

13. Euclidean Distance, (2014), http://en.wikipedia.org/wiki/Euclidean_distance

14. Levenshtein, V. I., 1966. Binary Codes Capable of Correcting Deletions, İnsertions, and Reversals", Soviet Physics Doklady 10 (8): 707–710.

15. Dice, L. R., 1945. Measures of the Amount of Ecologic Association Between Species, Ecology 26 (3): 297–302.

16. Jaccard Index, (2014), http://en.wikipedia.org/wiki/Jaccard_index

17. Winkler, W. E., 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research Methods (American Statistical Association): 354–359.

18. Tanimoto, T., 1957. An Elementary Mathematical theory of Classification and Prediction, Internal IBM Technical Report 1957.

19. Hamming, R. W., 1950. Error detecting and error correcting codes, Bell System Technical Journal, 29 (2): 147–160.

20. Black, P. E., 2006. Manhattan Distance, The National Institute of Standards and Technology - NIST, Dictionary of Algorithms and Data Structures, Vreda Pieterse and Paul E. Black, eds.

21. Minkowski Distance, 2014. http://en.wikipedia.org/wiki/Minkowski_distance

22. SIM, 2014. http://dickgrune.com/Programs/similarity_tester/

23. Hage, J., Radameker, P., Vught, N. V., 2010. A Comparison of Plagiarism Detection Tools, Technical Report UU-CS-2010-015, ISSN: 0924-3275.

24. Özen, Z., Gülseçen, S., 2012. Kaynak Kod Benzerliği ve Klon Kod Tespit Araçları, Akademik Bilişim'12 - XIV. Akademik Bilişim Konferansı Bildirileri Kitapçığı, Uşak.
25. Sherlock, 2014. http://sydney.edu.au/engineering/it/~scilect/sherlock/.
26. Cosma, G., Joy, M., 2012. An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis, Computers, IEEE Transactions on , vol.61, no.3, pp.379,394, DOI: 10.1109/TC.2011.223.
27. WCopyfind, 2014. http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/
28. CopyCatch Gold, 2014. http://www.cflsoftware.com/GoldFull.html
29. Prechelt, L., Malpohl G. , Phlippsen, M., 2002. JPlag: Finding Plagiarisms Among a Set of Programs, Technical Report 2000-1, University of Karlsruhe, J.UCS - The Journal of Universal Computer Science, Vol. 8, Issue 11, , 1016-1038, DOI: 10.3217/jucs-008-11-1016.
30. Metin Eşleştirme Sistemi, 2014. http://www.kemik.yildiz.edu.tr/data/File/ogr_projeler/Text%20Matching%20System.PDF
31. Doküman Benzerliği, 2014. http://www.kemik.yildiz.edu.tr/data/Document Similarity.rar
32. Yüksel, M. E., Turna, Ö. C., Ertürk, M. A., 2010. Bilgiye Erişim Sistemlerinde Veri Arama ve Eşleştirme, Akademik Bilişim'10 - XII. Akademik Bilişim Konferansı Bildirileri Kitapçığı, Muğla.
33. Başak, S., 2009. Türkçe Dokümanların Benzerliği, Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul.
34. Flajolet, P., Fusy, É, Gandouet, O., Meunier, F., 2007. HyperLogLog: the Analysis of a Near-Optimal Cardinality Estimation Algorithm, AOFA'07: Proceedings of the 2007 International Conference on Analysis of Algorithms DMTCS Proc. AH, pp. 127-146.
35. Kafka, F., 1915. Dönüşüm (Die Verwandlung).
36. Zemberek, 2014. http://code.google.com/p/zemberek/
37. Beyazperde, 2014.http://www.beyazperde.com/
38. HD Film Vadisi, 2014.http://www.hdfilmvadisi.com/