Sosyal Medya Ortamlarında Türkçe Dil Özelliklerine Dayalı Olarak Sahte Hesap Tespiti

Emre Beğen[1], Mustafa Kaya[1], Salih Keleş[1], Bora Karadağ[1], Tunga Güngör[2], Müslim Dayı[1], Hakan Çiftçi[1]

[1] Bilgi Birikim Sistemleri, İstanbul
ebegen@bilgibirikim.com, mkaya@bilgibirikim.com, skeles@bilgibirikim.com, bkaradag@bilgibirikim.com, mdayi@bilgibirikim.com, hciftci@bilgibirikim.com

[2] Boğaziçi Üniversitesi, Bilgisayar Mühendisliği, İstanbul
gungort@boun.edu.tr

Özet: Sahte hesap tespiti, Facebook,Twitter ve Linkedin gibi sosyal paylaşım sitelerindeki şirketlerin ve kişilerin gerçek olmayan hesaplarının tespiti işlemidir. Yurtiçinde ve yurtdışında itibar ve kriz yönetimi konusunda çalışan sosyal medya analiz programları fazla sayıda olmasına rağmen, sahte hesap tespiti yapabilen yazılımlar yok denecek kadar azdır. Bu çalışmada, türetme ve benzerlik algoritmaları kullanılarak, sahte hesapları bulunmak istenen kelime veya kelime grubunun benzerleri türetilmiştir. Verilen kelime ile türetilen kelime arasındaki benzerliği bulabilmek amacıyla, kosinüs benzerliği metriği ve geniş çaplı bir derlemden elde edilen bigram listesi kullanılmıştır. Türetilen kullanıcı adlarının sosyal ağ sitelerindeki profil bilgileri kontrol edilerek sahte hesap olup olmadığı tespit edilmeye çalışılmıştır. Hesap bilgilerine erişebilmek için sosyal medya arayüzleri (API) kullanılmıştır. Deneyler 17 adet gerçek kullanıcı adı üzerinde yapılmış ve %95 başarı oranı elde edilmiştir.

Anahtar Sözcükler: Sahte hesap, Kelime benzerliği, Kosinüs benzerliği, Düzeltim uzaklığı, Uygulama geliştirme arayüzü, Türkçe

Fake Account Detection based on Turkish Language Characteristics on Social Media

Abstract: Fake account detection is identification of unreal accounts of companies and individuals on social networking websites like Facebook, Twitter or Linkedin. Although there are several social media analysis programs performing reputation and crisis management processes, softwares detecting fake accounts are very rare. In this work, using some derivation and resemblance algorithms, words similar to the given input word or word pharse are derived. To detect resemblance between the original word and the derived word, cosine similarity metric and bigram statistics obtained from a large corpus are used. Profile informations of derived user names on social networks are checked in order to detect whether they are fake or not. Account informations are retrieved using social media interfaces (API). The experiments were performed on 17 real user names and 95% success rates were obtained.

Keywords: Fake account, Word similarity, Cosine similarity, Edit distance, Application programming interface, Turkish

1. Giriş

Facebook,Twitter,Linkedin gibi sosyal ağlarda kullanıcıların farklı amaçlarla açtıkları gerçek olmayan kullanıcı profilleri sahte hesap olarak adlandırılır. Sahte hesaplar, ünlüler adına açılıp takipçi toplama amaçlı, reklam yapmak amaçlı,bir markayla kötü bir deneyim yaşandığı için karalama kampanyası yürütmek maksatlı veya marka isimlerini kullanarak kullanıcıların kişisel bilgilerini ve profillerini elde etmek amaçlı olabilmektedir [1]. Özellikle günümüzde sosyal medyanın elde ettiği gücü düşünecek olursak, kişilerin veya şirketlerin adı kullanılarak açılan sahte hesapların büyük sorunlara yol açabildiği tahmin edilebilir. Bu gibi durumların önüne geçmek amacıyla sosyal medya takip ajanslarından ve yazılımlarından faydalanılmaktadır. Ancak bu yazılımların çoğu sosyal medya yönetimi ve analizi üzerine işlem yapmaktadır. Sahte hesap tespiti üzerine az da olsa yurtdışında örnek gösterebileceğimiz uygulamalar yer almasına rağmen,

ülkemizde bu alanda yapılan programlar ve çalışmalar yok denebilecek kadar azdır.

Şu ana kadar sahte hesap tespiti amacıyla çoğunlukla kişilerin ya da kuruluşların takip ettiği veya takipçileri arasında sahte hesabı olanları tespit eden programlar kullanılmıştır. Bu tarz programların eksik yönü, sorgulanan hesapların sadece takip edilen ve takipçiler arasında sahte hesap aramasıdır.

Bu çalışmada, sahte hesapları aranacak olan kelime veya kelime grubundan türetme algoritması kullanılarak türetilen kullanıcı adlarının (sahte olabilecek hesaplar) öncelikle var olup olmadığı kontrol edilir. Var olan kullanıcı adlarının sosyal medya (Facebook, Twitter, Linkedin, vb.) arayüzleri

(API) yardımıyla içerikleri incelenerek sahte hesap olasılığı ölçümlenir ve sınıflandırılır. Kelime türetilirken Türkçe dil bilgisi özellikleri ve sahte hesap açmak isteyen bir kullanıcının ne gibi yöntemler kullanacağı ile ilgili bilgiler dikkate alınmaktadır.

Bildirinin organizasyonu şu şekildedir. İkinci bölümünde ilgili literatür verilerek sosyal medya analiz ve sahte hesap tespiti programlarından kısaca bahsedilmiştir. Üçüncü bölümde geliştirdiğimiz ve kullandığımız algoritmalardan bahsedilmiştir. Dördüncü bölümde değerlendirme sonuçlarına değinilirken, son bölümde ise sonuç ve önerilere yer verilmiştir.

2. Literatür Araştırması

Şu ana kadar sahte hesap tespiti üzerine yapılan çalışmalarda metin madenciliği, kelime benzerliği algoritmaları, sezgisel algoritmalar ve n-gram teknikleri kulllanılmıştır.

Bir çalışmada, hesabın sahteliğinin tespiti için sosyal ilişki yapısı dikkate alınmış, ayrıca bu ilişki bağının oluşturulması için hesapların hangi niteliklerinin kullanabileceğinden bahsedilmiştir [2]. Algoritma gerçek dünyadaki bir problemin çözümü için hazırlandığından ve ilişki modeli çok büyüdüğünden dolayı, sistem dağıtık çalıştırılmaya yönelik olarak Hadoop yapısına uyarlanmıştır. Oluşturulan model üzerinde algoritma hareketleri için çeşitli teknikler denenmiş ve bunların çıktıya olan etkilerinden bahsedilmiştir. Ayrıca, oluşan çok büyük model üzerinde hız durumu da dikkate alınarak sezgisel algoritmalar geliştirilmiştir. Hazırlanan algoritmanın paralelleştirilmesi denemelerinden sonra uygulama gerçek problem için test edilmiş ve başarılı sonuçlar elde edilmiştir.

Diğer bir çalışmada, duvar gönderilerinin güçlü benzerlik gösteren içerikleri veya aynı çıkış URL bağlantısı içerenlerinin gruplandırılması ve bu gruplar üzerinden çeşitli algoritmaların geliştirilmesi konuları araştırılmıştır [3]. Burada belirtilen özelliklerle, duvar gönderileri kenar olmak üzere ve bağlantıları için belirtilen nitelikler ile büyük bir diyagram oluşturulmuştur. Daha sonra bu diyagram çeşitli algoritmalar ile alt diyagramlara bölünmüş ve alt diyagramlar ile gereksiz gönderilerin bölgeleri bulunmaya çalışılmıştır. Çalışmada kullanılan metin benzerlik ölçütü için metinler bir çeşit adres değerine (*hash*) çevrilmiş ve bu değerler sıralanarak bu sıralamada yakın olan metinlerin benzer oldukları kabul edilmiştir.

Bir diğer çalışmada ise, kısa metinlerin benzerliğinin hesaplanması için çeşitli benzerlik yöntemlerinin beraber kullanılması ile doğruluğun arttırılması hedeflenmiştir [4]. Var olan yöntemlerin tek başlarına kullanılmasının kısa metinler için yetersiz kaldığı belirtilmiştir. Bu sebeple, çalışmada benzerliğin hesaplanmasından önce metnin zenginleştirilmesi gibi yöntemlerle benzerlik doğruluğunun arttığı çeşitli grafikler ile gösterilmiştir.
Bir diğer çalışmada ise "MyPageKeeper" sınıflandırıcı algoritması adı altında bir algoritma geliştirilmiştir [5]. Geliştirilen sınıflandırıcı destek vektör makineleri (SVM) tabanlı bir sınıflandırıcıdır. Bu algoritma bir URL'yi içeren tüm gönderilerin hesaplanmasına yönelik bir algoritmadır. Algoritmanın kullanım yerleri olarak, gönderilerde "bedava", "acele edin", "ucuzluk" gibi ifadelerin geçmesi durumunda bunları gereksiz olarak işaretlemesi, gereksiz metinlerin benzerliği kullanılarak diğer metinlerin gereksizliğinin tespit edilmesi, yapılan gönderilerin az yorum alması durumunda potansiyel olarak gereksiz olarak işaretlenmesi ve gönderiler içinde geçecek bazı özel modellerden gönderilerin gereksizliğinin belirlenmesi gösterilmiştir.

3. Metodoloji

Şu ana kadar sahte hesap tespiti üzerine geliştirilmiş olan yerli uygulamaların ve çalışmaların sayısı yok denecek kadar azdır. Bu çalışmada kullanılan yöntem, öncelikle girdi olarak verilen kelimenin benzerlerinin türetilmesidir. Bu amaçla, ilk olarak girilen kelime belli bir ön işlemden geçirilerek kullanacağımız algoritmaya uygun hale getirilmektedir. Ön işlemden geçirilen kelimenin veya kelime grubunun benzerlerini türetebilmek için düzeltim uzaklığı (*edit distance*) operasyonları kulllanılmakta ve türetilen kelimeler bir listeye eklenmektedir.

Düzeltim uzaklığı algoritmasının bir örneği Tablo 1'de gösterilmiştir. Tablonun sağ alt bölmesinde görüldüğü gibi, "kalem" ile "kelam" kelimeleri arasındaki düzeltim mesafesi 2 olarak bulunmaktadır.

	k	a	l	e	m
k	0	1	2	3	4
e	1	1	2	3	4
l	2	2	1	2	3
a	3	3	2	2	3
m	4	4	3	3	2

Tablo 1. Düzeltim uzaklığı matrisi

Düzeltim uzaklığı yaklaşımı ile elde edilen listenin yapısı incelendiğinde, girdi kelimesine çok benzeyen sonuçlar olduğu gibi, oldukça farklı yazımların da olduğu görülmüştür. Listenin daha düzenli bir hale gelebilmesi için, liste sonuçları arasında verilen kelimeye benzerlik değerlerine göre sıralama ve filtreleme yolu izlenmiştir. Sıralama için gerekli olan benzerlik değerlerini elde edebilmek için, metin madenciliğinde de çok sık kullanılan ve kelime ya da metinler arasındaki benzerliği bulan kosinüs benzerliği (*cosine similarity*) yaklaşımı kullanılmıştır. Benzerlik değerlerine göre sıralanan listededе filtreleme yapılıp listenin uzunluğu

azaltılmıştır. Sonuçta elde edilen liste, girdi değerimize benzer kelimelerden oluşan kullanıcı adı listesi olarak kabul edilmiştir. Daha sonra kullanıcı adı listesinin içindeki her bir kullanıcı adına ait olan sosyal medya hesaplarının varlığının kontrolü yapılmıştır. Var olduğu tespit edilen hesapların içeriğini elde edebilmek için sosyal medya arayüzleri kullanılmıştır. Arayüzler sayesinde elde edilen bilgilerin puanlaması yapılarak sahte hesap ya da gerçek hesap olduğu ayırt edilmeye çalışılmıştır.

3.1 Ön İşleme

Çalışma kapsamında gerçekleştirilen ön işleme adımlarında, girdi tek bir kelimeden ibaret ise yapılan çalışma bu kelimenin içindeki büyük harflerin küçük harflere dönüştürülmesidir. Ancak girilen değer bir kaç kelimeden birden oluşuyor ise öncelikle bu kelime grubundaki kelimeler ayrılır ve aralarında noktalama işaretleri ya da boşluklar varsa bunlar silinir. Sonrasında tek kelimeden oluşan durumda yapıldığı gibi bütün kelimelerdeki bulunan büyük harfler küçük harflere dönüştürülür.

3.2 Benzer Kelimelerin Türetilmesi

Ön işleme aşamasından geçen kelime ya da kelime grubuna düzeltim yzaklığı operasyonlarının birlik ve ikilik işlemleri uygulanır. Düzeltim uzaklığı, basitçe ifade etmek gerekirse iki dizi, iki kelime, iki cümle gibi varlıklar arasındaki değiştirme, ekleme ve silme işlemlerinin yapılarak bu varlıklar arasındaki mesafe farkını bulmamızı sağlayan yöntemdir [6]. Uyguladığımız düzeltim uzaklığı operasyonlarını standart ve sezgisel olmak üzere iki farklı şekilde sınıflandırabiliriz.

3.2.1 Standart Operasyonlar

Standart olarak değerlendirebileceğimiz operasyonlara örnek verecek olursak, birli ya da ikili harf gruplarının çıkarılması, eklenmesi, kelime içerisindeki harfler arasında değişiklik yapılması sayılabilir. Bu operasyonları uygularken, alfabe listesi olarak oluşturduğumuz birli ve ikili harf ve harf gruplarından oluşan karakter dizilerini kullandık. Örneğin, “pegasus” kelimesinden türetme yaparken, alfabe listesindeki bir karakter “pegasus” kelimesindeki her bir karakterin yerine koyulmuştur.

pegasus →aegasus(p →a)

pegasus →pagasus(e →a)

pegasus →peaasus(g →a)

pegasus →pegasus(a →a)

pegasus →pegaaus(s →a)

pegasus →pegasas(u →a)

pegasus →pegasua(s →a)

Burada kullanılan yöntem, kelimedeki harf değiştirme işlemidir. Aynı şekilde, harf listesindeki bütün harf karakterleri girdi kelimesine ekleme, çıkarma gibi operasyonları uygulayarak da türetme yapılmaktadır. Harf listesinin yanı sıra sesli ve sessiz harflerden oluşan liste de kullanılarak sadece sesli ve sessiz harflere de bu operasyonlar uygulanmaktadır.

3.2.2 Sezgisel Operasyonlar

Standart olarak uygulanan operasyonlara ek olarak, sezgisel operasyonlar olarak adlandırabileceğimiz bazı operasyonlar üretilmiştir. Bu operasyonları oluştururken kendimizi sahte hesap açacak kişi yerine koyarak ne gibi yollar izlenebileceğini düşünmeye çalıştık. Örneğin, sahte hesap oluştururken ‘g’ harfi yerine ‘q’ harfi kullanmak, ‘z’ yerine ‘s’ kullanmak ya da ‘I’ yerine ‘l’ kullanmak gibi durumlar sık karşılaşılan durumlardır. Bir başka düşünce tarzı olarak da Türkçe dil yapısına göre aynı sınıfa giren harfleri kullanmayı seçtik. Sert ünsüzler ve yumuşak ünsüzler bu sınıflara örnek verilebilir. Bu gibi olasılıkları türetme algoritmamıza katabilmek için tıpkı alfabe listesi gibi ekstra listeler oluşturduk. Bunlardan bazıları şu şekildedir:

{l,i}

{j,s,z}

{c,g,k,p,t,q}

{f,h,k,p,s,t}

Yukarıdaki listeler türetme algoritmamızın sezgisel operasyonlarını oluşturan harf gruplarıdır.

3.3 Filtreleme

Standart ve sezgisel operasyonlar sonucunda elde ettiğimiz listede bulunan aynı kelimelerin veya kelime gruplarının listeden çıkarılması işlemidir. Ayrıca, ilk harfi verilen kelimenin ilk harfinden farklı olan kelimeler de listeden çıkarılmaktadır. Örneğin, verilen kelime “emniyet” ise, “umniyet”, “kmniyet”, vb. gibi ilk harfi “e”den farklı olan benzer kelimeler listeden çıkarılıyor. Burada kullanılan yöntem sezgisel olarak sınıflandırılabilir. Pratikte oluşan durumlar incelendiğinde, sahte hesabı açan kişinin hesap adını değiştirirken ilk harfinde değişiklik yapmak istemeyeceği varsayılarak böyle bir filtreleme yapılmaktadır. Ayrıca kullanılan başka bir sezgisel filtrelemeye örnek, verilen kelimede bulunan yan yana sessiz harf sayısından daha fazla

yan yana sessiz harf içeren türetmelerin listeden çıkarılması olarak gösterilebilir.

3.4 Benzerlik Değerinin Bulunması

Filtreleme işlemi sonunda sadeleşen listemizde benzerlik algoritmaları uygulanarak elde edilen benzerlik değelerine göre sıralama işlemi yapılmaktadır. Böylece verilen kelimeye daha çok benzeyen türetmeler listemizin üst sırasına alınarak kontrol kolaylığı sağlanmıştır. Benzerlik değeri bulunurken izlenen yol sırası ile bigram eşlenme sayısını bulmak ve eşlenen bigramların frekans değerini almak, kosinüs benzerliği ile elde ettiğimiz değeri bigram işlemleri sonucunda elde ettiğimiz değer ile normalize etmektir.

3.4.1 Bigram İşlemleri

Filtreleme işlemi sonunda**,** kelimenin sırası ile bigramlarının derlemden faydalanılarak oluşturduğumuz bigram listesi içinde varlığı kontrol edilmektedir [7]. Kelimenin bigramları eğer bu listede varsa bigram sayacı 1 artırılır ve var olan bigramların frekans değerleri toplanır.

Bigram	Frekans ağırlığı
ar	0,021250571
la	0,019801422
an	0,019298044
er	0,018522993
in	0,018490537
le	0,017201178
de	0,01439105
en	0,013385064
ın	0,013214157

Tablo 2. Örnek bigram listesi

Tablo 2'de kullanılan bigram listesinin ufak bir parçası gösterilmektedir. Girdi kelimesinde bu liste içerisinde yer alan bigramların kontrolü yapılmakta ve var olanların frekans değerleri toplanmaktadır.

3.4.2 Kosinüs Benzerliği

İki metin arasındaki benzerlik değerini bulmak için metin madenciliğinde de sıkça kullanılan yöntemlerden birisi kosinüs benzerliğidir. Metinlerin birer vektör olarak düşünüldüğü bu yaklaşımda, iki vektörün birbirleri ile olan ilişkisi bir açı ile ifade edilmektedir [8]. Tamamen aynı yönü gösteren iki vektör için kosinüs değeri 1 olurken, tamamen birbiri ile ilişkisiz iki vektör için ise kosinüs değeri 0 olacaktır. Örneğin; "GOOGLE" ile "YAHOO" kelimeleri arasındaki ilişki düşünüldüğünde, terim frekansını kullanarak iki vektörde de geçen terimlerin sayısını bir vektörde gösterecek olursak:

Terimler:{G,O,L,E,Y,A,H}

Google:[2,2,1,1,0,0,0]

Yahoo:[0,2,0,0,1,1,1]

şeklinde gösterebiliriz.

Bu iki vektör (w1, w2) arasındaki kosinüs benzerliğini aşağıdaki şekilde hesaplayabiliriz:

$$\text{Kosinüs}(w1,w2) = \frac{w1.w2}{\|w1\|\|w2\|} \qquad (1)$$

Burada iki vektör arasındaki kosinüs bağlantısı için iki vektörün skaler çarpımının iki vektörün vektörel çarpımına oranı alınmıştır.

w1.w2 = [2,2,1,1,0,0,0]. [0,2,0,1,1,1]

=(2*0)+(2*2)+(1*0)+(1*1)+(0*1)+(0*1)+(0*1)

w1.w2=$\sqrt{5}$ = 2,23606797749979

Bu sonuç iki vektörün noktasal çarpımıdır.

$$\|w1\| = \sqrt{2^2 + 2^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2}$$

$\|w1\| = \sqrt{10}$ = 3,162277660168379

$$\|w2\| = \sqrt{0^2 + 2^2 + 0^2 + 0^2 + 1^2 + 1^2 + 1^2}$$

$\|w2\| = \sqrt{7}$ = 2,645751311064591

$\|w1\|\|w2\| = \sqrt{10} * \sqrt{7} \cong 8{,}3666$

Yukarıda bulduğumuz sonuç vektörlerin vektörel büyüklüklüklerinin çarpımını gösteriyor.

$\cos(w1,w2) = \frac{2{,}2360}{8{,}3666} \cong 0{,}2672531255$

Skaler çarpım ve vektörel çarpımları yerine koyduğumuzda elde edilen sonuç iki metin arasındaki kosinüs benzerliği değerini elde etmemizi sağlıyor ve bu örnekte bu değer yaklaşık olarak 0,26722531255 gibi bir değere eşit oluyor.

Bigram işlemleri uygulandıktan sonra elde edilen sonuç ve kosinüs benzerliğinden elde edilen sonuç, bizim verilen kelime ile türetilen kelime arasındaki benzerlik değerini elde etmemizi sağlımaktadır. Benzerlik değeri bulunan kelimeler, bu değerlere göre büyükten küçüğe sıralanıyor. Listenin son hali ise hesap içeriği incelenmek üzere sosyal medya arayüzlerinde kullanılmak için parametre olarak gönderiliyor.

3.5 Hesap İçeriğinin Kontrolü

Benzer kelimelerin türetilmesi ve filtrelenmesi işlemlerinin ardından elimizde oluşan kullanıcı adı listesini kullanarak hesabın varlığı tespit edilir ve hesap içeriğinin elde edilebilmesi için kullanıcı adı kullanılarak sosyal medya arayüzleri ile hesap içeriği çekilir. Çekilen hesap içeriği başlıca kriterlere göre puanlanarak hesabın sahte ya da gerçekliği tespit edilmeye çalışılır.

3.5.1 Hesabın Varlığının Tespiti

Girilen kelimenin benzerlerinin türetilmesi ve filtrelenmesi işlemi sonunda elde ettiğimiz liste sosyal ağlarda hesapların varlığının denetlenmesi için kullanılıyor. Listeden elde edilen türetilmiş kullanıcı adları Facebook, Twitter, Linkedin gibi sosyal ağ sitelerinde URL'lere eklenerek o uzantıdaki adresin HTTPWebRequest metodu ile request isteminde bulunduğumuzda dönen cevaba göre hesabın varlığı tespit edilmeye çalışılıyor. Eğer sunucudan gelen cevap "200OK" ise o hesap mevcuttur, eğer gelen cevap "404NotFound" ise öyle bir hesap bulunamadı demektir. Böylece varlığı tespit edilen hesabın kullanıcı adı sosyal medya arayüzlerinde kullanılarak hesap içeriği inceleniyor.

3.5.2 Hesabın İçeriğinin İncelenmesi

Varlığı tespit edilen hesapların kullanıcı adları sosyal medya arayüzlerinde (Facebook Api,Twitter Api, Linkedin Api, vb.) kullanılarak hesap içeriği elde edilir. Hesap içeriğinde dikkat edilen hususlar profil fotoğrafının varlığı, paylaşımlarının genel olarak ne üzerine olduğu, ne sıklıkta paylaşım yapıldığı, kaç arkadaşı ya da takipçisi olduğu, kişisel bilgilerini (hakkında, okul geçmişi, iş geçmişi, vb.) paylaşma durumu, kişisel web sitesi varlığı, paylaşımlarının yorum alma durumu, vb. kriterlerdir. Bu kriterlerin puanlaması, hesabı sahte ya da gerçek olarak sınıflandırabilmemizi sağlar. Sayılan kriterleri sağladığı düşünülen hesaplar (+) puanlama alırken kriterleri sağlamayanlar (-) puan almaktadır. (+) puan aralığımız 0-50 arası, (-) puan aralığımız -50-0 arasında değişmektedir. Bir hesap için eğer puan toplamı belirlenen eşik değerinin altında ise ya da eşik değeri belirlenmediği durumda puan toplamı 0'ın altında kalıyorsa, o hesap sahteliği yüksek bir hesap olarak değerlendirilmektedir. Eşik değerinin üstünde ya da 0'ın üstünde kalan hesaplar ise sahtelik olasılığı düşük olan hesaplar olarak değerlendirilebilir. Örneğin, kullanıcı adı "anadolujet" olan bir Facebook hesabının kriterlere göre puanlanması Tablo 3'te gösterildiği gibi olsun.

Kriter	Puan
Hakkında skoru	25
Tescilli Skoru	-25
Profil Fotoğrafı Skoru	25
Web Sitesi Skoru	-25
Kullanıcı Adı Skoru	25
Telefon Numarası Skoru	25
Açıklama Skoru	-15
Tekrarlanan Paylaşım Skoru	20
Beğenme Sayısı Skoru	40
Aktif Paylaşım Skoru	20
Link'li Paylaşım Skoru	0
Toplam	115

Tablo 3. Sahtelik Kriterleri ve Puanlama

Örneği kısaca açıklayacak olursak, sahteliği kontrol edilen hesabın ilk olarak hakkında bölümü kontrol ediliyor. Eğer hakkında bölümünde hesap hakkında bilgi varsa skora (+) puan ekleniyor. Aynı şekilde tescilli hesap olup olmadığı kontrol ediliyor. Hesap tescilli ise (+) puan alıyor, aksi halde (-) puan alıyor. Aynı şekilde diğer kriterler de kontrol edilerek skoru hesaplanıyor. Sahtelik eşik değerinin de 85 olarak verildiği örneğimizde, kriterlerin puanlandırılmasına göre toplam skorumuz 115 çıkmıştır. Bu puana göre hesabımızın sahte hesap olmadığı, gerçek hesap olduğu düşünülebilir.

Geliştirilmiş olan sahte hesap algoritması Tablo 4'te gösterilmektedir.

Girdi:
Sahte hesapları aranacak kullanıcı adı
Çıktı:
Skor toplamı eşik değeri üzerinde kalan hesaplar
Başla:
1: Input değerini ön işleme tabi tut
2: Düzeltim uzaklığı operasyonları ile benzer kullanıcı adları türet ve listeye yaz
3: Listeyi filtrele
4: **Döngü:** Listedeki her kullanıcı adı için:
5: Bigram listesi girdinin bigramlarını içeriyor ise;
6: Bigram frekanslarını topla
7: Kosinüs benzerliği ile benzerlik değerini bul
8: **Döngü sonu**
9: Listedeki kullanıcı adlarının sosyal ağlardaki varlığını tespit et
10: **Döngü:** Var olduğu bilinen her kullanıcı adı için hesabın içeriğini incele
11: Kriterlere uyan hesapların skorlarını topla
12: Eşik değeri < skor toplamı ise hesap gerçektir
13: Eşik değeri > skor toplamı ise hesap sahtedir
14: **Döngü sonu**
Son

Tablo 4. Sahte hesap algoritması kodu

4. Deneyler

Bu çalışma kapsamında yapılan deneylerde, toplamda 17 gerçek kullanıcı adı için türetilen toplam 1942 kullanıcı adı

kullanılmıştır. Değerlendirme metriği olarak doğruluk (*accuracy*) ölçülmüştür. Bu metriğin formülü aşağıdaki gibidir:

$$\text{Doğruluk} = \frac{\text{Doğru olarak bulunan}}{\text{Doğru olarak bulunan+ Doğru olarak bulunmayan+ Silinmiş Hesaplar}}$$

Başarıyı ölçmek için manuel olarak sahte ve gerçek olmak üzere iki sınıfa ayrılan hesaplar ile algoritmamıza göre gerçek ve sahte olarak sınıflandırılan hesaplar karşılaştırılmıştır.

Arama Kelimesi	Hesaplar	Başarılı	Başarısız	Silinmiş Hesaplar	Başarı Yüzdesi(%)
anadolu jet	7	3	4	0	42,86%
Ankara Büyükşehir	36	12	24	0	33,33%
arçelik	363	75	271	17	21,68%
Aselsan	10	4	6	0	40,00%
Axa sigorta	19	4	15	0	21,05%
denizbank	308	268	40	0	87,01%
Doğuş Üniversitesi	56	5	51	0	8,93%
ensonhaber	5	1	4	0	20,00%
Gittigidiyor	42	11	31	0	26,19%
Hepsiburada	29	5	24	0	17,24%
LC waikiki	501	49	444	8	9,94%
nike türkiye	49	12	34	3	26,09%
Opel Türkiye	23	1	20	2	4,76%
samsung türkiye	23	5	18	0	21,74%
turkish airlines	109	24	83	2	22,43%
türk hava yolları	49	17	26	6	39,53%
Vestel	313	100	202	11	33,11%

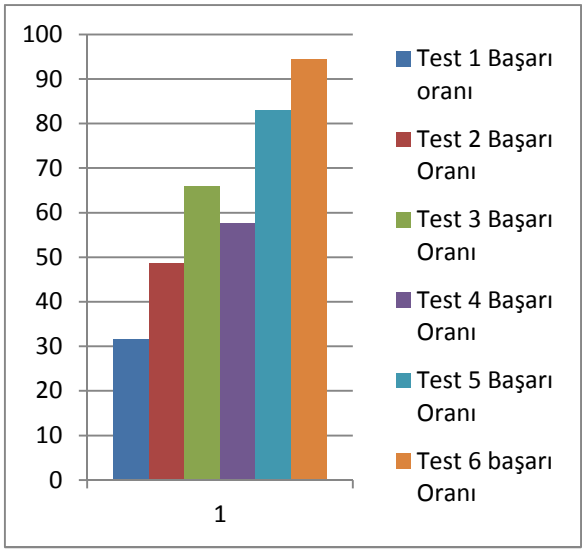
Tablo 5. Deneyler başarı yüzdesi

Tablo 5'te gerçek olan kullanıcı adlarına göre türetilmiş kullanıcı adları arasından, bizim tahminimize göre gerçek olduğu düşünülen kullanıcı adları ile algoritmamıza göre gerçek olduğu düşünülen kullanıcı adları karşılaştırması yapılmıştır. Örneğin, "anadolu jet " için var olduğu tespit edilmiş 7 hesap için manuel olarak sahte ve gerçek olduğu düşünülen hesaplar ile algoritmamıza göre gerçek ve sahte olarak değerlendirilen hesapların karşılaştırılmasında 3 tane başarılı, 4 tane başarısız sonuç olduğu görülmüştür. Bunun anlamı bizim yaptığımız tahminler ile algoritmamıza göre bakılan hesapların 3 tanesi aynı sınıflandırmaya girerken, 4 tanesi farklı olarak değerlendirilmiştir. Diğer kullanıcı adları için de ilk deneylerde elde edilen başarı sayısı ve yüzdesi tabloda görülmektedir.

Sonuç	**TOPLAM**
Hesaplar	1942
Başarılı	596
Başarısız	1297
Silinen Hesaplar	49
Başarı Yüzdesi	31,48%
	27,99%

Tablo 6. Test 1 başarı sonucu

Tablo 6'da toplam 17 gerçek kullanıcı adı için türetilen ve var olduğu tespit edilen 1942 hesabın daha önceden manuel olarak değerlendirilmiş sınıflandırılması ile algoritmamıza göre elde edilen sınıflandırılması arasındaki karşılaştırmanın test sonucu gösterilmektedir. Bu test bizim ilk testimiz olmakla birlikte %31,48 gibi bir başarı oranına sahiptir. Başarı oranımızı artırmak için kriter puanlamamızda değişiklikler yapılmıştır. Örneğin, ilk testimizde bütün kriter skorları için pozitif skorumuz (+25) ve negatif skorumuz (0) olarak değerlendirilmiş ve başarı yüzdesi %31,48 olarak tespit edilmiştir. İkinci testimizde ise kriter skorları için pozitif skorlarımız ve negatif skorlarımızda değişiklik yapılmış ve başarı yüzdemizin %48,75 olarak değiştiği görülmüştür. Bütün testler sonucunda elde ettiğimiz başarı sonucu ise %94,26 olarak görülmektedir. Uygulanan bütün testlerimiz için kriterlerin toplam skorlarını gösteren grafik Şekil 1'de görülmektedir.



Şekil 1. Tüm testlerin sonuçları

5. Sonuç ve Öneriler

Bu çalışmada daha önce ülkemizde örneğine nadir rastlanan bir çalışma olan sahte hesap tespiti üzerine uygulama geliştirilmiş ve başarılı sonuçlar elde edilmiştir. Geliştirilen ve uygulanan metotlar daha çok metin madenciliği ve sosyal ağlarda hesap denetimi üzerine olmuştur. Metinler arası farklılıkları bulmak amacıyla kullanılan düzeltim uzaklığı ve sezgisel yöntemlerin ve benzerlik derecesini bulmada kullanılan kosinüs benzerliğinin başarılı olduğu gözlemlenmiştir. Karşılaşılan bir zorluk, türetilen kullanıcı adları ile hesapların varlığını tespit ederken sosyal ağ arayüzleri kullanıldığında istem (*request*) limitine takıldığı görülmüştür. Bu sorunu çözmek için arayüzler yerine HttpRequest yöntemi kullanılmış ve sorgu limiti bu şekilde aşılmıştır. Varlığı tespit edilen hesapların içeriğinin incelenmesi sırasında kriter puanlaması testlerinin başarılı olduğu ve uygun kriter puanlarının tespit edildiği gözlemlenmiştir.

İleride sahte hesap tespiti üzerine bu makalede uygulanan yöntemlerden farklı olarak daha gelişmiş benzetme algoritmaları ve makine öğrenmesi algoritmaları kullanılarak daha başarılı sonuçlar elde edilebilir.

Teşekkür

Bu çalışma, TÜBİTAK TEYDEB tarafından 7131134 nolu proje numarası ile desteklenmiştir.

Kaynaklar

[1] http://gokhanahi.com/2012/12/09/sirketiniz-adina-sahte-hesap-acilirsa-ne-yapmalisiniz/

[2] Qiang Cao, Michael Sirivianos, Xiaowei Yang, Tiago Pregueiro,“ Aiding the Detection of Fake Accounts in Large Scale Social Online Services”**.**

[3] Hongyu Gao, Jun Hu, Xiaowei Yang, Christo Wilson, Zhichun Li , Yan Chen, Ben Y. Zhao “Detecting and Characterizing Social Spam Campaigns.

[4] Vasileios Hatzivassiloglou, Judith L. Klavans, Eleazar Eskin, “Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning”.

[5] Md Sazzadur Rahman, Ting-Kai Huang, Harsha V. Madhyastha, Michalis Faloutsos “FRAppE: Detecting Malicious Facebook Applications”.

[6]https://web.stanford.edu/class/cs124/lec/med.pdf

[7] Shasha Xie, Yang Liu, Huang,“Using Corpus And Knowledge-Based Similarity Measure In Maximum Marginal Relevance For Meeting Summarization”.

[8] Timothy J. Hazen, “Direct And Latent Modeling Techniques For Compting Spoken Document Similarity”.