Bulanık Mantık Algoritmaları Kullanarak Kaynak Kod Benzerliği Bulma

Fatma BOZYİĞİT[1], Deniz KILINÇ[1], Alp KUT[2], Muhammet KAYA[1]

[1] Celal Bayar Üniversitesi, Yazılım Mühendisliği Bölümü, Manisa

fatma.bozyigit@cbu.edu.tr,deniz.kilinc@cbu.edu.tr,
muhammet.kaya@cbu.edu.tr

[2] Dokuz Eylül Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir

alp@cs.deu.edu.tr

Özet: Bir programcı tarafından oluşturulmuş bir yazılımın başkaları tarafından intihali yazılım dünyasında birçok alanda istenmeyen durumdur. Bu çalışmada, özellikle eğitim alanında öğrenciler arasında kod paylaşımının önüne geçmek, ders değerlendirirken haksızlıkların oluşmasını engellemek amacı ile öğrenciler tarafından hazırlanmış olan yazılımların içerisinde hangilerinin benzerlik gösterdiği, benzerlik oranlarının ne olduğu gibi soruların cevabının bulunması hedeflenmektedir. Uygulamanın geliştirilmesinde N-gram ve bulanık mantık algoritmaları yaklaşımı esas alınmıştır. Geliştirilen uygulamanın test çalışmaları Celal Bayar Üniversitesi, Yazılım Mühendisliği öğrencilerinin uygulamaya yönelik ödev dokümanları üzerinden yapılmıştır. Yapılan uygulamanın çıkarmış olduğu sonuçlar, gerçek değerler ile karşılaştırılarak doğruluk oranı hesaplanmıştır.

Anahtar Sözcükler: Kaynak kod benzerliği, N-gram, bulanık mantık

Source Code Similarity Detection With Using Fuzzy Logic Algorithms

Abstract: Plagiarism of a software which is created by a programmer, is the undesirable situation in the many fields of software world**.** In this study the aim is finding the answers of questions such as; which codes are similar, what similarity ratios are, etc. to prevent the sharing source codes among students and to be fair while evaluating grades of courses**.** While developing the application, n-gram and fuzzy logic algorithms will be taken as a base. Tests of application will be completed on home works, depends on applications, of students in Software Engineering Department of Celal Bayar University. The results extracted by application will be compared by the real values and correlation value was computed.

Keywords: Source code similarity, N-gram, fuzzy logic

1.Giriş

Çalıntı kod, kaynağı detaylı bir şekilde anlamadan, basit metin düzenleme işlemleri kullanarak başka bir programdan elde edilen koddur.[1] Bir yazılımın lisansı onay verdiği takdirde, yazılıma ait kaynak kodun bir kısmının veya tamamının kullanılmasında bir sakınca yoktur. Fakat izin verilmez ise, kaynak kodlar üzerinden alıntı yapmak veya tamamen bir kodu alıp sahiplenmek etik olarak uygun olmayan bir şekilde çalıntı kod oluşturmaktır ki bu kod sahibinin istemediği bir durumdur. Bu konuya hukuksal olarak Fikir ve Sanat Eserleri Kanunu'nda değinilmiş ve bilgisayar programları ilim eserleri sınıfına dâhil edilirken, kaynak kodun sahibi bir eser sahibi olarak ele alınmıştır.[2] Kod intihali konusu, birçok alanda karşımıza çıkabilecek bir problemdir. Örnek olarak, şirketlerin kendilerine özel geliştirdikleri uygulamaların kodlarının çalınması ve başka kişi ve kurumlar tarafından izinsiz kullanılması verilebilir. Ayrıca, eğitim alanında, yazılım geliştirme ile ilgili derslerin ödev ve projelerinde kod hırsızlığı konusu önemli bir sorun teşkil etmektedir.

Teknoloji geliştikçe, yazılım alanındaki uygulamaların sayısının hızla yükselen bir doğrultuda olduğu görülmekte ve bu duruma paralel olarak kod intihali konusu daha da önemli bir sorun olarak karşımıza çıkmaktadır. Kodun çalıntı olup olmadığını anlamak ve kod hırsızlığının önüne geçebilmek için çeşitli yöntemler denenmektedir. Bunlardan biri de kaynak kodlar üzerinde benzerlik oranı bulan bir yazılım aracı ortaya çıkarmaktır. Bunun için hâlihazırda geliştirilmiş araçlar bulunmaktadır. Örnek olarak Plague [3],

MOSS [4], JPlag [5] ve YAP [6] uygulamaları gösterilebilir.[7] Bahsi geçen bu araçlar metrik sistemine dayalı oldukları için programlama diline bağlı olup, sadece belirli diller üzerinde benzerlik bulma çalışmalarında kullanılmaktadırlar. Dolayısı ile bu durum kullanıcılara kısıtlama getirmektedir.

Bu çalışmada amaç, kaynak kodlar arasında benzerlik bulmak ve kodun çalıntı olup olmadığına karar verebilmeyi sağlamaktır. Bu doğrultuda, programlama diline bağlı kalmamak ve her dilde yazılmış kaynak kodları inceleyebilmek için N-gram algoritması kullanılıp benzerlik oranları çıkarılmış ve veri setindeki dokümanların birbirleri ile ne derecede benzer oldukları hakkında yorumlamalar yapılmıştır.

Bildirinin devamında, ikinci bölümde benzer çalışmalar incelenip, genel olarak hangi yöntemleri kullandıkları ve hangi veri setleri ile çalıştıkları gibi bilgiler verilmiştir. Üçüncü bölümde çalışmanın yönteminden ve kullanılan algoritmalar hakkında bilgi verilmiştir. Dördüncü bölümde deneysel veri setleri ve elde edilen sonuçlar ele alınmıştır. Beşinci bölümde sonuç ve tartışmaya yer verilirken, son bölüm olan altıncı bölümde gelecekte yapılması planlanan çalışmalardan bahsedilmiştir.

2. Mevcut Çalışmalar

Keselj tarafından 2003 yılında yayınlanan çalışma N-gram algoritmasına yönelik çalışmalardan birisidir[8]. Bu uygulama Pascal programlama dili üzerindeki benzerlikleri elde etmeye yönelik olarak tasarlanmıştır. Keselj bu uygulama ile belirli uzunluktaki ve en yüksek frekanstaki N-gram değerlerini kullanarak kendi algoritmasına özgü olan Göreli Uzaklık(Relative Distance) değerini hesaplamıştır. Böylece programa daha önceden tanıtılan yazarlar içerisinden, test edilen kaynak kodun hangi yazara ait olabileceği hakkında çıkarımlar yapılmış ve doğruluk oranı hesaplanmıştır.

Frantzeskou tarafından 2007 yılında yayınlanan SCAP Metot ile ilgili bilgi içeren makalede ise Keselj'e ait çalışmadan farklı olarak, SCAP'in tüm programlama dilleri üzerinde etkili bir çalışma olduğu öne sürülmüştür.[9] SCAP metodunun özelliklerinden biri yazar havuzuna eklenen profil örneklerinin basitleştirilmiş ve kısa kodlar içeriyor olmasıdır. Uygulamanın iyi sonuçlar verebilmesi için bir yazara az sayıda kodun yeterli olduğu iddia edilmektedir. Bu çalışmada Basitleştirilmiş Profil Kesişim(Simplified Profile Intersection) değeri hesaplanarak kaynak kodun uygun profile ne kadar yaklaştığı, ne kadar oranda kesişim gösterdiği tahmin edilir. Bu uygulamadaki başarı oranı n değeri 8 ve profil değeri 2000'e kadar çıkarıldığında %100'ü bulmaktadır.

Bracardo(2013) çalışmasında stilometri kullanarak kısa mesajların yazarlarını belirlemeyi hedeflemiştir.[10] Stilometri yazım sırasında ortaya çıkan karakteristik özelliklerdir. Bu çalışma ile yazara ait belirli davranışları ortaya çıkartarak internet ortamındaki dokümanlar üzerinde yazar kimliğinin belirlenmesi sağlandığı iddia edilmektedir. Amacın gerçekleşmesi için gözetimli öğrenme tekniği ile N-gram analiz sonuçları kombine edilmiştir. Bracordo çalışmasında yeni bir yaklaşımı sunduğunu ileri sürmektedir. Bu yaklaşımda ele alınan veri setleri içerisinde N-gram olma ve olmama miktarları üzerinde durulmuştur.

Caunor ve Trankle(1994) metin sınıflandırma çalışmasında metinsel hatalarda tolerans sağlamasından dolayı N-gram yöntemini tercih etmişlerdir.[11] Veri seti olarak Usenet haber grup makalelerinden faydalanmışlardır. Sistemin temeli N-gram frekanslarının hesaplanması ve kıyaslanmasına dayanmaktadır. Öncelikle gerekli profiller elde edilmiş daha sonra veri seti içerisinde bulunan her bir dokümanın bu profile uzaklığı hesaplanır. En az mesafede olan dokuman seçilerek işlem tamamlanır. Bu çalışmanın dezavantajı küçük boyutlu dosyalar üzerinde çalışmaya daha uygun olmasıdır.

3. Yöntem

3.1 N-gram Algoritması

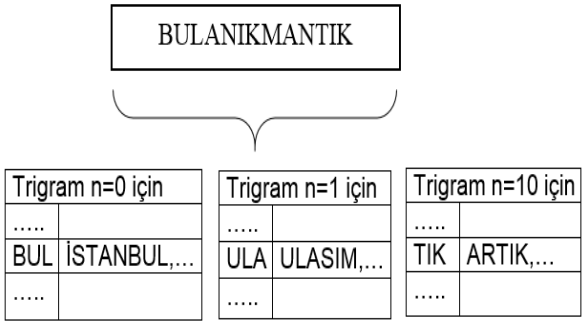
N-gram algoritması verilen bir dizideki alt ifadelere ait kombinasyonları elde ederek, bu kombinasyonların karşılaştırılacak olan dizi içerisindeki tekrar oranını bulmaya yarayan bir yöntemdir. Doğal dil işleme alanında sıkça kullanılan N-gram algoritması, teknolojinin ilerlemesi ile birlikte programlama dillerine ait dokümanların incelenmesinde de kullanılmaya başlanmıştır. Bu yöntem ifadelerin benzerlik oranını bulup sınıflandırma konusundaki başarısı ile en basit ve kullanışlı yöntemlerden biri olarak kabul edilmektedir.

Bir N-gram algoritması bitişik bir ifadeyi kullanıcı tarafından belirlenmiş olan N uzunluğunda alt ifadelere ayırarak çalışmaya başlar. Sondan (N-1). dizi elemanında gruplama işlemi sonlandırılır. Bu n değeri 1 ise algoritma uni-gram, 2 ise bi-gram, 3 ise tri-gram vb. şeklinde adlandırılır. Bir örnek ile açıklamak gerekir ise; "***bulanık mantık***" kelime grubunu ele alalım. Bu ifadenin 3-gram algoritmasının girdisi olarak kullanıldığını varsayarsak, alt ifadeler Tablo 1 de olduğu gibi sıralanabilir.

N-Gram	Frekans
bul	**1**
ula	**1**
lan	**1**
…	**…**
…	**…**
tık	**1**

Tablo 1 tri-gramlar ve frekansları

Tablo 1 de gösterildiği gibi "***bulanık mantık***" ifadesindeki sondan üçüncü karakter olan t harfi sonrası için gruplandırma işlemi devam edilmemektedir. Metin içerisindeki tri-gramların geçme sıklığı frekansları belirlenerek, karşılaştırma yapılacak metinde geçen tri-gram frekansları ile kıyaslanarak benzerlik bulunabilir.



Şekil 1 tri-gram

Özetle bu algoritmada, bir metin içerisinde geçen N-gramların frekansları ile benzerlik oranının hesaplanmak istediği dokümandaki N-gram miktarları karşılaştırılır. İki dokümanın birbirine olan yakınlığı 0 ve 1 arasında değerler alır. Elde edilen değer 1'e yaklaştıkça benzerlik oranının arttığı çıkarımı yapılmaktadır.

N-gram algoritmasındaki önemli noktalardan birisi de N değerinin belirlenmesidir. N değerinin düşük tutulduğu çalışmalarda benzerlik oranının yüksek, N değerinin yüksek alındığı uygulamalarda ise bu oranın giderek düştüğü gözlemlenir. İki durum da, benzerlik bulma çalışmalarında sağlıklı sonuçlar vermeyecektir. Örnek vermek gerekirse uni-gram ile yapılan çalışma, karakterleri tek tek ele alıp karşılaştırma yaptığı için benzerlik oranı bulunmak istenen dokümanda daha çok ortak nokta bulunup, yüksek bir benzerlik oranı verecektir. Bu durumda birbiri ile fazla benzerliği olmayan iki ayrı doküman için yüksek benzerlik oranı çıkması beklenmeyen bir durum olarak ortaya çıkabilir. Bu çalışmada tri-gram sonuçlarını daha optimal sonuçlar verdiği gözlemlenmiş ve uygulamada N değeri üç olarak belirlenmiştir.

Kaynak kod benzerliklerini tespit etmeyi hedefleyen bu çalışmada makine öğrenmesi algoritmaları yerine N-gram algoritmalarının tercih edilmesinin sebebi programlama dilleri konusunda esneklik sağlaması, karşılaştırma yapılacak dokümanlarda dil bağımsızlığının söz konusu olması ve kod içerisindeki her ifadeyi kapsamasıdır. Bu sebeplerden dolayı, daha doğru sonuçlar elde edilmektedir.

3.2 Bulanık Mantık Algoritmaları

Bulanık mantık, ilk olarak 1965 yılında Dr. Lotfi A. Zadeh tarafından "*Information and Control*" dergisinde bu konu hakkında makale yayınlanması ile ortaya çıkan bir kavramdır. Bulanık mantık, kesin ve tam değerlerden ziyade yaklaşık olarak nitelendirilen ve muhakeme yapılabilen değerler ile ilgilenen çok değerli mantık biçimidir.[12] Bulanık mantık, insan mantığında olduğu gibi, uzun-kısa yerine çok uzun-uzun-orta-kısa-çok kısa gibi ara değerler tutmaktadır ve her şey [0,1] aralığındaki değerler ile gösterilir. Özetle, eğer bir sistemdeki sonuçların kesin olarak tanımlanması gerekmiyorsa, sonuçların aralık değerlerinde görüntülenmesi isteniyorsa ya da matematiksel ölçütlerin sıfat olarak belirlenmesi ve sınıflandırılması gerekiyorsa bulanık mantık yöntemi tercih edilmektedir.

Bu çalışmada kaynak kodların benzerlik oranı artan sıradan azalana doğru göz önünde bulundurularak kodlar içerisinde çok benzer, benzer, orta düzeyde benzer, az benzer ve benzerlik yoktur gibi çıkarımlarda bulunulmuştur. Çalışmada N-gram sonuçları test edilirken birinci veri setinde, temel olarak alınmış bir kaynak kodun üzerinde değişiklikler yapılarak beş adet farklı program örneği elde edilmiştir. Orijinal kod ile az farklılığa sahip olan kod arasında benzerlik oranı en yüksek değerde çıkarken, en farklı kod ile benzerlik oranı en düşük çıkmıştır. Test edilen kodların benzerlik oranları göz önünde bulundurularak Tablo 2 deki gibi benzerlik sınıflandırmaları yapılmıştır.

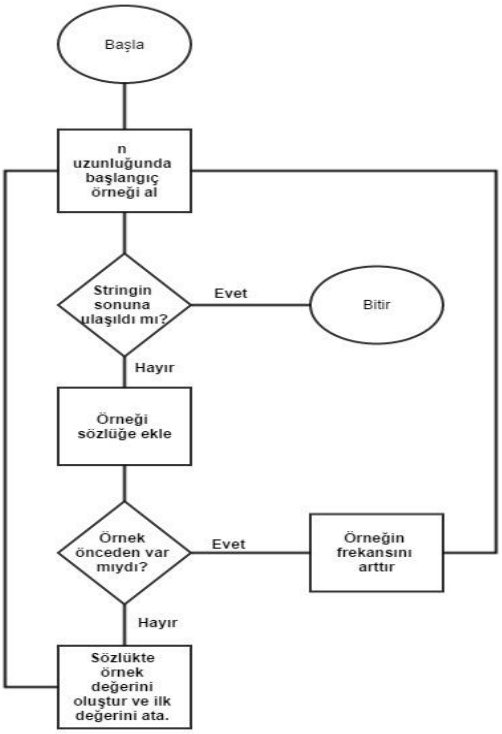
Benzerlik Oranı	Kategori
%100-%85	**Çok benzer-Kopya**
%85-%70	**Benzer**
%70-%55	**Orta düzeyde benzer**
%55-%40	**Az benzer**
%40 ve aşağısı	**Benzerlik yoktur**

Tablo 2 Benzerlik sınıflandırmaları

3.3 Uygulama

Uygulamanın ilk aşamasında N-gram algoritması

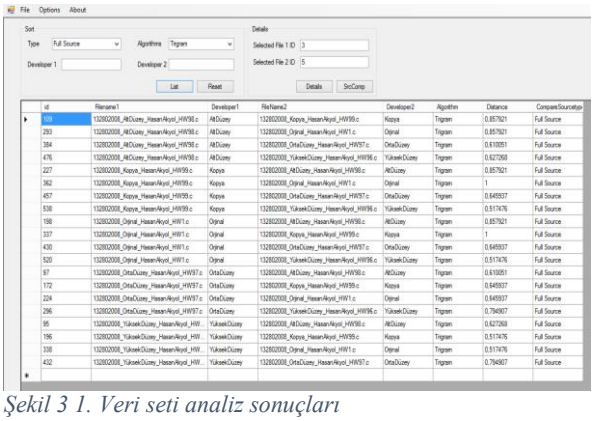
kodları yazılmıştır. Oluşturulan N-gram algoritmasına ait akış diyagramı Şekil 2 de görüldüğü gibidir.



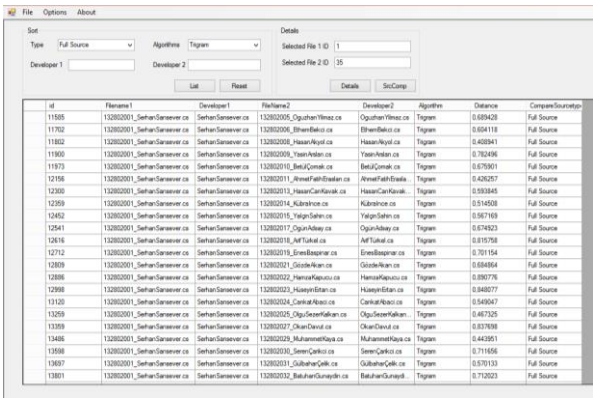
Şekil 2 N-Gram algoritması akış diyagramı

İkinci aşamada ise elde edilen benzerlik oranları bulanık mantık algoritması kullanılarak sınıflandırılmıştır. Belirli aralıklarda kodların kopya dereceleri hakkında sınıflandırmalar ortaya çıkarılmıştır.

Son aşamada iki farklı deneysel veri seti oluşturulmuş, uygulamanın test çalışmaları tamamlanmıştır. Şekil 3'te birinci veri setine ait ekran görüntüsü yer alırken, Şekil 4'te ikinci veri setine ait ekran görüntüleri ve analiz sonuçları yer almaktadır.



Şekil 3 1. Veri seti analiz sonuçları



Şekil 4 2.Veri seti analiz sonuçları

4. Deneysel Çalışmalar

4.1 Deneysel Veri Seti Oluşturulması

Çalışmanın test aşamasında kullanılmak üzere 2 adet veri seti oluşturulmuştur. Veri seti kaynağı olarak Celal Bayar Üniversitesi Yazılım Mühendisliği öğrencilerinin "Algoritma ve Programlama" ve "Nesneye Yönelik Programlama" derslerine ait ödev dokümanları kullanılmıştır.

Birinci veri seti belirli bir öğrencinin "Algoritma ve Programlama Dersi" ödevi için oluşturduğu C kaynak kodunun üzerinde değişiklikler yapılarak elde edilmiş 5 farklı kaynak kod kombinasyonudur. Bu veri seti benzerlik sırası ile öğrencinin değişiklik yapılmamış ödevinden çok değişiklik yapılmış ödevine kadar olan tüm dokümanları içerir.

İkinci veri seti olarak, Yazılım Mühendisliği 1. ve 2. Sınıf öğrencilerinin programlama ile alakalı tüm ödev dokümanlarını içermektedir. Bu veri seti 120 adet C programlama dilinde yazılmış ve 75 adet C# dili ile yazılmış kaynak kod içermektedir. 1. Sınıf öğrencilerinin Algoritma ve Programlama dersine ait ödevler C programlama dili ile yazılmışken, 2. Sınıf öğrencilerinin Nesneye Yönelik Programlama dersi C# programlama dili ile yazılmıştır. N-gram algoritması programlama dili açısından esneklik sağladığı için her iki sınıf öğrencilerinin ödevlerinin karşılaştırması rahatlıkla yapılabilmiştir.

4.2 Deneysel Sonuçlar

Belirli bir öğrencinin farklılaştırılmış kod örneklerine sahip olan 1. veri seti incelendiğinde, Tablo 2 de görüldüğü gibi, bulanık mantık yaklaşımına uygun şekilde yapılmış sınıflandırmanın tutarlı sonuçlar verdiği gözlenmektedir. Örnek olarak, Tablo 3 de incelenen Kaynak Kod 1 orijinal kaynak kod iken, Kaynak Kod2 az değişikliğe uğramış ve yazarın kodu

ile çok benzer olan dokümandır. Kaynak kodlardan yeni kodlar üretilirken çok benzerlik gösterecek halinden az benzerlik gösterecek haline kadar sıralama yapılmıştır. Bu kombinasyonları oluştururken amaç benzerlik ölçütlerini sınıflara ayırmaktır. Çok benzer, benzer, orta seviyede benzer, az benzer gibi 4 farklı kategoride N-gram sonuçları karşılaştırılmıştır. Karşılaştırılan sonuçlar doğrultusunda gerçek değerler ile N-gram sonuçlarının birbiri ile tutarlılık gösterdiği saptanmıştır. Gerçekte yüksek benzerlik oranına sahip Kaynak Kod 1 dokümanı ve Kaynak Kod 2 dokumanı Tablo 1 de gösterilen benzerlik oranı ile tutarlıdır.

	Kod1(Orijinal Kod)
Kod1(Orijinal Kod)	**100%**
Kod2(Çok Benzer)	**%91.2**
Kod3	**%84.8**
Kod4	**%69.4**
Kod5(Az Benzer)	**%43.7**

Tablo 3 1. veri seti için tri-gram sonuçları

C ve C# programlama dilleri ile programlama ödevi hazırlayan tüm öğrencilerin kodlarını içeren ikinci veri setinin sonuçları değerlendirilirken esas olarak beklenilen dil bağımsız ödev benzerliklerini ortaya çıkarabilmektir. N-gram sayesinde beklenilen sonuçlara ulaşılmış, farklı programlama dilleri ile oluşturulmuş ödevlerin kendi aralarında benzerlik oranları elde edilmiş ve 195 ödev içerisinde 42 tanesinin kopya olduğu sonucuna varılmıştır. Ayrıca kullanılan bu veri setinin N-gram sonuçları değerlendirilirken göze çarpan en önemli nokta, dersteki başarıları ve sosyal çevreleri nedeni ile kopya girişiminde bulunmaları tahmin edilemeyecek öğrenciler arasında çok yüksek benzerlik oranlarının ortaya çıkmasıdır. Sonuç olarak bu öğrencilere ait ödevler gözle de incelendiğinde, net bir şekilde bir kopya durumunun var olduğu görülmüştür. Bu durum çalışmanın beklenen sonuçlar verdiğinin göstergesi olarak kabul edilmiştir.

5.Sonuç ve Tartışma

Kaynak kodda benzerlik bulma çalışmaları, doğal dil ile yazılmış eserlerde benzerlik çıkarımı yapmaktan çok daha zordur. Doğal dil kullanımında her bireyin kendine özgü üslûba sahip olması, ele alınan eserde ayırt edici noktaları belirlemeyi kolaylaştıran bir etkendir. Kaynak kod benzerliği bulunurken, kullanılan klasik yöntemlerden biri yazarın tarzını belirlemeye yönelik çalışmalardır. Bu çalışmalarda ilk olarak yazarın karakteristik özelliklerini belirlemeye yarayan belirli metriklerin çıkarımı yapılır. İkinci adım ise elde edilen metriklerin sınıflandırma algoritmaları kullanarak test edilmesi, kaynak kodun hangi yazara ait olduğu hakkında çıkarımlar yapmak ve böylelikle benzerlikleri ortaya çıkarabilmektir.

Kaynak kodlar üzerinde benzerlik bulma çalışmalarında metrik çıkarımı yapmak gibi klasik yöntemleri kullanmanın dezavantajları bulunmaktadır. Bu dezavantajlardan en önemlisi, bahsi geçen yöntemin dil bağımlı olmasıdır. Farklı programlama dillerinin, farklı özellik ve sözdizimleri ile oluşturulması sebebiyle çıkarılan karakteristik metrik çeşitleri sadece belirli diller için geçerli olmaktadır. Bu sebeple, bir veya birkaç dil üzerinden elde edilen özel metrikler, başka programlama dilleri ile yazılmış dokümanlarda kullanılamazlar.

Bu çalışmada, kaynak kod üzerinden metrik çıkarma ve elde edilen metrikleri sınıflandırarak benzerlik bulmaya çalışmanın dezavantajlarını göz önünde bulundurarak, her dil üzerinde bağımsızca çalışabilmek için N-gram algoritması kullanılmıştır. Veri setleri üzerinde tri-gram algoritması denenmiş ve başarılı sonuçlar ile karşılaşılmıştır.

Kullanılan veri setlerinin analizi yapıldığında, öğrencilerin yazmış oldukları programları içeren 2. veri setinde 195 ödev içerisinde 42 adet kopya kod olduğu fark edilmiştir. Kopya aldıkları tespit edilen öğrenciler ile görüşüldüğünde, bazıları başka birinden kod aldığını kabul ederken, bazı öğrenciler grup halinde çalıştıklarını iddia etmişlerdir. N-gram algoritması ile test edilen ödevlerde kopya çeken öğrencilerin belirlenmesi ve uygun şekilde notlandırılması ile daha sonra verilen ödevlerde kopya oranının düştüğü gözlemlenmiştir. Karşılaşılan bu durum çalışmanın oluşturulma amacına uygun sonuçlar verdiğini göstermektedir.

6. Gelecek Çalışmalar

Bulanık Mantık Algoritmaları Kullanarak Kaynak Kod Benzerliği Bulma çalışmasının devamında yapılması planlanan çalışma, kod yazar kimliğini belirlemek ve test dokümanının hangi yazara ait olabileceği çıkarımları yapabilmektir. Bu amaçla ortaya çıkan N-gram frekansları metrik gibi ele alınıp, kullanılarak kod yazarı profili ortaya çıkarmaktır. Yazar profillerinin belirlenmesi sonrasında kümeleme algoritmaları kullanarak ile test verilerinin hangi yazara ait olduğu bulunması planlanmaktadır.

Referanslar

1. A. Parker ve J. Hamblen, “Computer algorithms for plagiarism detection,” IEEE Trans. Education, vol. 32,

pp. 94–99, May 1989.

2. Fikir ve Sanat Eserleri Kanunu. (2003). - http://mevzuat.meb.gov.tr/html/7981_5846.html

3. G. Whale, "Plague: Plagiarism Detection Using Program Structure," Dept. Comput. Sci., Univ. New South Wales, Kensington, Australia, Tech. Rep. 8805, 1988

4. A. Aiken. Measure of Software Similarity.- http://www.cs.berkeley.edu/~aiken/moss.html

5. G. Malpohl. JPlag: Detecting Software Plagiarism.- http://www.ipd.uka.de:2222/index.html

6. "YAP3: Improved detection of similarities in computer program and othertexts," in Proc. 27th SCGCSE Tech. Symp., Philadelphia, PA, 1996, pp. 130–134

7. Chen, X., Francia, B., Li, M., McKinnon, B., Seker, A. "Shared Information and Program Plagiarism Detection", IEEE Trans. Education , vol. 50,No. 7, July 2004.

8. Keselj, Peng, Cercone, Thomas, 2003. N-gram based author profiles for authorship attribution. In Proc. Pacific Association for Computational Linguistics.

9.Frantzeskou, Georgia, Gritzalis, Efstathios, Stamatatos, Efstathios, Katsikas, Sokratis, 2006. "Identifying Authorship by Byte-Level N-Grams: The Source Code Author Profile (SCAP) Method." Inc Proc. Institute for Linguistic Evidence.

10. M. L. Brocardo, I. Traore, S. Saad, I. Woungang," Authorship Verification for Short Messages using Stylometry".

11. W. Cavnar ve J. Trenkle, "N-Gram Based Text Categorization".

12. Fuzzy Logic.- http://en.wikipedia.org/wiki/Fuzzy_logic