Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, Cilt 24, Sayı 1, 2019 ARAŞTIRMA

DOI: 10.17482/uumfd.484525

TÜRKÇE METİNLERİN SINIFLANDIRMA BAŞARISINI ARTIRMAK İÇİN YENİ BİR YÖNTEM ÖNERİSİ

Metin BİLGİN [*]

Alınma: 17.11.2018; düzeltme: 12.02.2019; kabul: 12.02.2019

Öz: Bu çalışma, yazarı bilinmeyen bir dokümanının yazarını tahmin etmeyi amaçlamaktadır. Bunun için 6 farklı köşe yazarına ait 6 köşe yazısı öncelikle ön-işlem aşamasına sokulmuştur. Ardından bu metinler üzerinden n-gram (2-3) ile özellikler çıkarılmıştır. Çıkarılan özellikler üzerinden sistem 6 farklı makine öğrenmesi üzerinde çapraz geçerleme (10) ile test edilmiştir. Buraya kadar olan kısım literatürde şimdiye kadar uygulanmış olan yöntemdir. Bizim önerimiz ön işlem aşamasının ardından eldeki metinleri LZW algoritması ile kayıpsız sıkıştırarak özellik sayısını azaltmak ve bunun sistemin başarısı üzerindeki etkileri araştırmak üzerinedir. Ön-işlemden geçmiş olan metinler LZW algoritması ile binary (ikili) ve decimal (onlu) olarak sıkıştırılır. Sıkıştırmanın ardından n-gram (2-3) ile çıkarılan özellikler ile sistem 6 farklı makine öğrenmesi yönteminde test edilmiş ve çalışma sonuçları 5 farklı metrik için incelenmiştir. Yapılan çalışma sonucunda ikili olarak sıkıştırılmış metinler hem 2-gram hem de 3-gramda, 6 farklı makine öğrenmesi algoritmasında da daha iyi sonuçlar elde etmiştir. Random Tree ve Naïve bayes algoritmasında onlu sıkıştırma, ham verinin gerisinde kalsa da diğer 4 algoritmada daha iyi elde sonuçlar elde etmiş ama ortalama başarı değerlerinde geride kalmıştır. Yapılan çalışma sonucunda ikili sıkıştırma tüm metriklerinde diğer iki yönteme göre daha başarılıdır. Yapılan çalışmada yazar tanıma işlemi yapılmış olsa da önerilen bu yöntemin tüm metin sınıflandırma işlemlerinde kullanılabileceği düşünülmektedir.

Anahtar Kelimeler: Metin Sınıflandırma, Doğal Dil İşleme, LZW, Metin Sıkıştırma, Makine Öğrenmesi

A Novel Method Proposal to Increase the Classification Success of Turkish Text

Abstract: This study aims to estimate the author of an unknown document. For this purpose, first of all, six different columns of 6 different columnists were pre-processed. Then with n-grams (2-3) features were extracted from these texts. The system has been tested with 10-fold cross-validation on 6 different machine learning algorithms. This part of the study is the method that has been applied so far in the literature. Our suggestion is to reduce the number of features with the LZW algorithm and to investigate the effects on the success of the system. The pre-processed texts are compressed binary and decimal with the LZW algorithm. After compression, the system has been tested with 6 different machine learning algorithms, and the study results has been analyzed for 5 different metrics. As a result of the study, the compressed binary text has obtained better results in both 2-gram and 3-gram, for 6 different machine learning algorithms. In the Random-Tree and Naïve Bayes algorithm, decimal compression is behind the raw data. In the other four algorithms, it achieved better results but remained behind the average success values. As a result of the study, binary compression is more successful in all metrics than the other two methods. In the study, although the author recognition process has been done, it can be thought that the proposed method can be used in all text classification procedures.

Keywords: Text Classification, Natural Language Processing, LZW, Text Compression, Machine Learning

[*] Uludağ Üniversitesi Mühendislik Fakültesi Görükle Kampüsü 16059 Nilüfer/BURSA
İletişim Yazarı : Metin Bilgin (metinbilgin@uludag.edu.tr)

1. GİRİŞ

Dünyada bilgisayar çağıyla beraber ülkelerdeki insanların kullandığı bilgi miktarı çok hızlı bir şekilde artış göstermiştir. Bu bilgi yığını içerisinde insanın aradığını bulması ve kullanması için belirli kurallara göre bilgilerin düzenlenmesi gerekmektedir (Doğan ve Diri, 2010). Bu problemin çözümü ancak dokümanları sınıflandırma ile çözülebilir. Metin sınıflandırmanın hedefi sınıfı belli olmayan bir metnin, ilgili metnin özelliklerine bakarak önceden sınıflandırılmış olan bir metin sınıfına dahil edilmesidir (Doğan, 2006). Metin sınıflandırma bilgi alma (information retrieval), bilgi çıkarma (information extraction), doküman indeksleme/filtreleme, otomatik olarak meta data elde etme ve web sayfalarını hiyerarşik olarak düzenleme gibi pek çok alanda kullanım alanına sahiptir (Türkoğlu ve diğ. 2006).

Metin sınıflandırma sistemlerinin ilk örnekleri 70'li yıllarda karşımıza otomatik doküman indeksleme olarak çıkmıştır. Belirli bir konu için özel sözlükler oluşturulmuş ve bu sözlük içerisindeki kelimeler birer kategori gibi algılanarak metinler sınıflandırılmıştır. Fürnkranz (1998) n-gram (2 ve 3 uzunluğunda) özelliklerini, Tan ve diğ. (2002) 2-gram'ları (bi-gram) kullanarak bir algoritma geliştirmiş ve doküman sınıflandırmada performansı arttırmışlardır. Çatal ve diğ. (2003), n-gram'ları kullanarak NECL adını verdikleri bir sistem geliştirmişlerdir. Diri ve Amasyalı (2003) bir metnin yazarını ve türünü belirlemede kullanılmak üzere 22 adet stil belirleyicisi oluşturmuş ve bunları kullanan bir sınıflandırma sistemi geliştirmişlerdir. Yine Amasyalı ve Diri (2006) 2 ve 3- gram'ları kullanarak metnin yazarını, türünü ve yazarının cinsiyetini belirme, Amasyalı ve Yıldırım (2004) Türkçe haber metinlerinin otomatik olarak sınıflandırılması üzerine çalışmışlardır. Bu çalışmaların yanı sıra literatürde çeşitli metin sınıflandırma problemleri için birçok başarılı çalışma yapılmıştır. Bunlara örnek olarak yazar tanıma Türkoğlu ve diğ. (2007)-Peng ve diğ. (2003)-Stamatos ve diğ. (2000)-Levent ve Diri (2014), metin konusunu belirleme Bekkerman ve diğ. (2002)-Song ve diğ. (2005)-Fattah (2017)-Johnson ve Zhang (2017), e-posta sınıflandırma Çiltik ve Güngör (2008)-Ciya ve diğ. (2001)-Gaines ve Carney (2018), metnin yazarının cinsiyetini belirleme Amasyalı ve Diri (2006), duygu analizi Gaines ve Carney (2018)-Çoban ve Özyer (2015)-Gezici ve Yanıkoğlu (2018) çalışmaları verilebilir. Bu çalışmalarda her problem türü için çeşitli metin temsil yöntemleri önerilmiştir. En çok kullanılan yöntemler; kelime / kelime grubu frekansları Bekkerman ve diğ. (2002), n-gram frekansları Çiltik ve Güngör (2008)-Ciya ve diğ. (2001), kelime kümeleme Bekkerman ve diğ. (2002), saklı anlam indeksleme Özel (2004) ve fonksiyonel kelimelerdir (Holmes 1998).

Makalenin organizasyonu şu şekildedir; ikinci bölümde veri kümesi, üçüncü bölümde önerilen yöntem, dördüncü bölümde deneysel çalışma ve beşinci bölüm de sonuçlar ve tartışma verilmiştir.

2. VERİ KÜMESİ

Gerçekleştirilen çalışmada kullanılan veri kümesi 2012 yılında Yıldız Teknik Üniversitesi Kemik Doğal Dil İşleme gurubu tarafından hazırlanan 50 farklı yazara ait 50 farklı köşe yazısından oluşan veri kümesi içerisinden rastgele seçilen 6 yazara ait 6 farklı köşe yazısından oluşmaktadır [Kemik 2019]. Metin dosyalarının ham hali, ikili (binary) sıkıştırılmış ve onlu (decimal) sıkıştırılmış haldeki metin dosyalarına ait bilgiler Tablo 1 ve metin dosyalarına ait örnekler (kısmen) Tablo 2'de verilmektedir.

Tablo 1. Veri Kümesine ait bilgiler

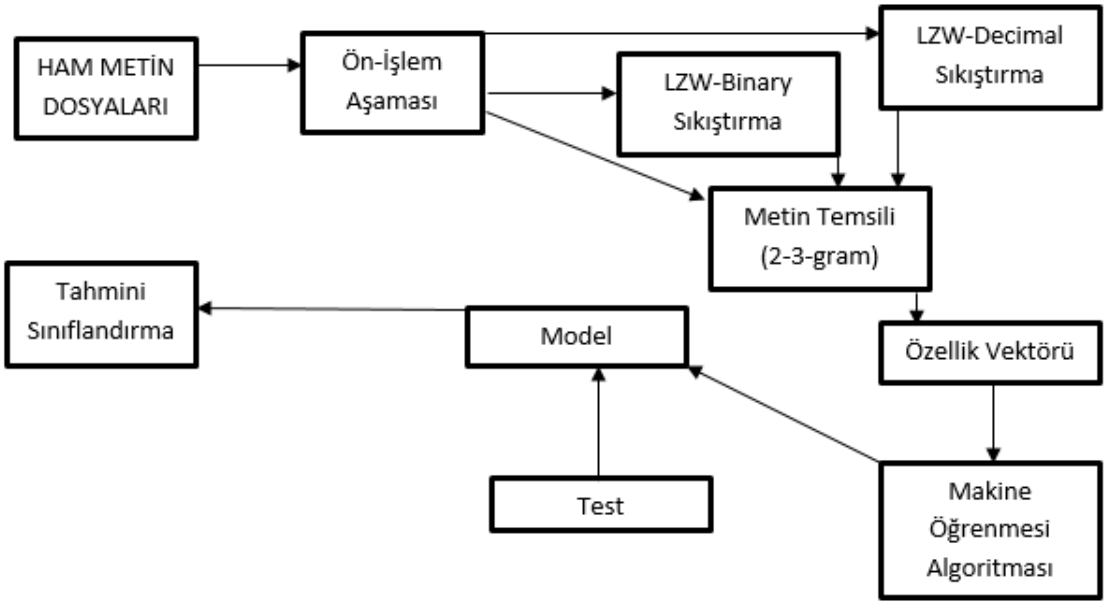
Veri Türü	Dosya sayısı	Sınıf Sayısı	Özellik Sayısı
Ham Metin Dosyası (2-gram)	36	6	644
Ham Metin Dosyası (2-gram)			4685
İkili Sıkıştırılmış Metin Dosyası (3-gram)			5
İkili Sıkıştırılmış Metin Dosyası (2-gram)			9
Onlu Sıkıştırılmış Metin Dosyası(2-gram)			101
Onlu Sıkıştırılmış Metin Dosyası (3-gram)			994

Tablo 2. Metin Dosyası Örnekleri

Ham Metin Dosyası	
Metin	Sınıf Bilgisi
Türk futbol kamuoyu dün geceki maçı Galatasaray'ın kazanamayacağını düşünüyordu ama rakipler için evdeki hesap çarşıya uymadı. Eğer aynı Galatasaray iki hafta sonra Kadıköy'de kaybetmezse şampiyonluk için play-off'a dev bir adımla ilerlemiş olur. Gelelim hakem Halis Özkahya'ya, kartlarını çok yerinde kullandı. Özellikle hava toplarında Melo ve Riera'ya verdiği sarı kartlar doğru. İkinci yarıda Eboue'ye yapılan harekette birçok kişi kırmızı kart beklerken hakemin kararı bence yine doğruydu. ….	Yazar A
İkili Metin Dosyası	
001000010 001110101 000100000 001101101 001100101 001101101 001101100 001100101 001101011 001100101 001110100 001110100 001100101 010000010 001100001 001101110 001111001 001100001 001101011 000100000 …	Yazar A
Onlu Metin Dosyası	
66 117 32 109 101 109 108 101 107 101 116 116 101 130 97 110 121 97 107 32 99 111 107 116 117 114 46 32 128 110 97 108 116 63 108 63 32 100 111 110 132 134 114 100 140 171 44 32 114 117 104 108 97 114 163 104 97 115 116 …	Yazar A

2. ÖNERİLEN YÖNTEM

Ham metin dosyaları üzerinden çeşitli metin temsil yöntemleri ile elde edilen özellikler kullanılarak çeşitli makine öğrenmesi algoritmaları ile sınıflandırma işlemi şimdiye kadar birçok çalışmada gerçekleştirilmiştir. Biz bu çalışmamızda ön işlemden geçmiş veriler üzerinde kayıpsız bir sıkıştırma algoritması olan LZW yöntemini kullanarak metinleri ikili ve onlu formatta sıkıştırarak sınıflandırmayı öneriyoruz. Ardından sıkıştırılmış bu metinleri tıpkı klasik yöntemde olduğu gibi herhangi bir metin temsili ile ifade ederek özelliklerini çıkarıp, ardından da makine öğrenmesi algoritmaları ile sınıflandırmaya çalışıyoruz. Önerdiğimiz yöntemin akış diyagramı Şekil 1'de görülmektedir.



Şekil 1:
Önerilen Yöntemin Akış Diyagramı

2.1. Ön işlem

Elimizdeki tüm metin dosyalarına sırasıyla aşağıdaki işlemler uygulanarak metinler özellik çıkarılabilecek hale getirilmiştir. Adımlar;

- Tüm metinler küçük harfe çevrilir,
- Noktalama işaretleri temizlenir,
- Metin içindeki durma kelimeleri (stop-words) temizlenir,
- Metinler kelimelere (token) ayrılır,
- Oluşturulan kelimeler daha ayırt edici olması için kelimelere gövdeleme (stemming) işlemi uygulanır.

Ön işlemler ardından elimizdeki metin dosyaları herhangi bir metin temsili yöntemi ile özellik çıkarılabilir hale getirilmiş olmaktadır.

2.2. Metin Temsil Yöntemi

Bir metin temsil yöntemi olan n-gram, bir karakter katarının n adet karakter dilimidir. n-gram tabanlı sınıflandırma yöntemi, doküman içerisindeki karakter tabanlı n-gram'ların kullanım sıklığına dayalı bir işlemdir. Bu çalışmada, n-gram'ın farklı birkaç uzunluğu alınarak 2- ve 3- gramlar kullanılmıştır Doğan ve Diri (2001). Çok seyrek frekans matrisleri üretiyor olmaları sebebiyle, n'nin daha büyük pencere boyutuna sahip n-gramlar kullanılmamıştır. Örneğin "metin sınıflandır" cümlesi için;

2-gram :
“me”,”et”,”ti”,”in”,n_”,”_s”,”sı”,”ın”,”nı”,”ıf”,”fl”,”la”,”an”,”nd”,”dı”,”ır”,”rm”,”ma”
3-gram:
“met”,”eti”,”tin”,in_”,”n_s”,”_sı”,”sın”,”ını”,”nıf”,”ıfl”,”fla”,”lan”,”and”,”ndı”,”dır”,”ırm”,”rm
a”

n-gramlar için özellik vektörü oluşturulurken frekans hesaplamasında Terim Frekansı (Term Frequency) metodu kullanılmış ve Eşitlik 1’de denklemi verilmiştir. TF metodu, bir özelliğin bir metin içerisinde geçme sayısının bulunmasıdır. Böylelikle bir özelliğin bir metin içindeki geçme sayısı ne kadar fazla ise o özellik o metin için o kadar önemlidir bilgisi çıkarılabilir (Bilgin, 2018). Bu yönteme göre satırlarında metinlerin, sütunlarında terimlerin (n-gram) yer aldığı bir matris oluşturulmaktadır. Oluşturulan matrisin [i, j] gözünde i. Metinde j. kelimenin kaç kez geçtiğinin sayısı tutulur. Oluşturduğumuz matrisin satır sayısı metin sayısına (36), sütun sayısı ise tüm metinlerde geçen farklı terimlerin (n-gram) sayısına eşittir. 2-gram TF ile oluşturulmuş bir özellik vektörü örneği Tablo 3’de verilmektedir.

$$m_j = tf_j \tag{1}$$

m_j= Kelimenin j. metindeki ağırlığı, tf_j = Kelimenin j. metinde geçme sayısı

Tablo 3. TF 2-gram için özellik vektörü

Metin	me	et	ti	in	n_	_s	sı	…
M1	1	3	3	2	0	1	5	…
M2	3	1	0	0	5	4	1	…
…	…	…	…	…	…	…	…	…

2.3. Sıkıştırma Algoritması

Bu çalışmada kayıpsız metin sıkıştırma algoritması olarak LZW algoritması kullanılmıştır. LZW algoritması , Terry Welch tarafından 1984’te LZ78 algoritmasının bir türevi olarak ve daha gelişmiş versiyonu olarak oluşturulmuştur. LZ78 yaklaşımını yüksek performanslı disk ünitelerine uyarlamış ve ortaya çıkan yeni algoritma LZW olarak kabul görmüştür (Welch, 1984). LZW hem sıkıştırma hem de açma performansı açısından LZ78 ailesinin en iyisi olarak isimlendirilebilir. Her tip veri üzerinde iyi sonuçlar veren bir algoritma olduğu için, kendisinden sonra gelen birçok algoritma LZW’yi temel almıştır.

LZW algoritmasında, LZ78’de kullanılan ikili yapısındaki ikinci elemanın gerekliliği ortadan kalkmıştır. Kodlayıcı, önce kaynaktaki tüm karakterlerden bir sözlük oluşturarak gönderir. Bu karakterler bir ön geçişle bulunabilir. Eğer ASCII Tablosundaki tüm karakterler kullanılacaksa ön geçiş yapılmasına ve sözlüğün gönderilmesine gerek yoktur. Kodlama aşamasında okunan her karakter sözlükte aranır. Bulunursa bir sonraki karakter de okunur ikisi birleştirilerek aranır. Sözlükte karşılığı bulunamayana kadar bu şekilde devam eder. Sözlükte karşılığı bulunmayan bir girdiye ulaşıldığında ise son karakteri hariç önceki karakterlerinin sözlükteki karşılığı kodlanır. Bu kod ile son karakterin sözlükteki kodu birleştirilerek sözlükte yeni bir girdi oluşturulur. Son karakter, sonraki adımda ilk karakter yapılarak kodlamaya devam edilir.

Sıkıştırmış metnin gösterim şekline göre ikili (binary) ya da onlu (decimal) şeklinde isimlendirilmektedir.

2.4. WEKA

WEKA (Waikato Environment for Knowledge Analysis), GPL lisansına sahip ve açık kaynak kodlu olarak JAVA dilinde geliştirilmiş bir üründür. WEKA üzerinde bulunan onlarca hazır kütüphane sayesinde veri ön işleme, özellik seçimi, sınıflandırma ve kümeleme gibi bir çok işlem yerine getirilebilir. Farklı formatlarda veri girişini destekleyen programda ağırlıklı olarak arff formatındaki dosyalar işlenmektedir.

Arff Dosya Formatı İngilizce, Attribute Relationship File Format kelimelerinin baş harflerinden oluşmuştur. Arff dosya yapısı, Weka'ya özel olarak geliştirilmiştir ve dosya, metin yapısında tutulmaktadır. Dosyanın ilk satırında, dosyadaki ilişki tipi (relation) tutulmakta olup ikinci satırdan itibaren veri kümesindeki özellikler (attributes) yazılmaktadır. Özelliklerin hemen ardından veri kümesi yer alır ve veri kümesindeki her satır bir örneğe (instance) işaret etmektedir. Ayrıca veri kümesindeki her örneğin her özelliği arasında da virgül ayıracı kullanılmaktadır. Arff formatında yazılmış örnek kod Şekil 2'de görülmektedir.

```
@ATTRIBUTE kelime_basina_harf_sayisi NUMERIC
@ATTRIBUTE cumle_basina_kelime_sayisi  NUMERIC
@ATTRIBUTE yazari {omer_seyfettin, peyami_sefa}
@DATA
5.1 3.5 omer_seyfettin
4.9 6.0 peyami_sefa
4.7 7.2 omer_seyfettin
4.6 4.1 peyami_sefa
5.0 9.6 peyami_sefa
```

Şekil 2: *Arff Dosya Örneği*

Şekil 2'de Arff dosyasında 5 adet metne ait 2'şer özellik ve metinlerin ait oldukları kategoriler verilmiştir. Ömer Seyfettin'e ait 2 metin ve Peyami Safa'ya ait 3 metine ait kelime ve harf sayılarıyla ilgili özellikler tanımlanmıştır. Öncelikle metinlerin @ATTRIBUTE ile özellik tanımlamaları yapılmış daha sonra @DATA kısmı ile de metinlerin bu özelliklere ait değerleri verilmiştir. Sütundaki son değer ise metnin sınıfını vermektedir (Amasyalı ve diğ. 2010).

Şekil 2'de verilen örnek dosyada, hava tahmini için kullanılan nem, sıcaklık ve basınç değerleri bir dosya içerisinde 4 örnek içerecek şekilde gösterilmiştir. Bu değerler tip olarak sayısal değerler olduğundan "numeric" olarak ifade edilmiştir. Ancak bu değerler nominal, real,string ve date gibi değerlerde olabilir.

2.5. Text2Arff

Metinlerden özellik çıkarımı için hazırlanmış olan TexTtoArff, Java dilinde yazılmış bir kütüphanedir. Metin işleme, veri madenciliği, makine öğrenmesi gibi birçok konuda yapılan çalışmada kullanılmak üzere hazırlanmıştır (Amasyalı ve diğ. 2010). Bu çalışmada 2 ve 3 gramların çıkarılması için Text2Arff programının 5. Versiyonu kullanılmıştır.

3. DENEYSEL SONUÇLAR

Bu çalışmada metin sınıflandırma başarısını artırabilmek için yeni bir yöntem önerilmiştir. Önerilen yöntem 6 farklı yazarın 6 farklı köşe yazısı için oluşturulan deney ortamında 6 farklı makine öğrenmesi algoritması (Random Tree-RT, Naive Bayes-NB, Bagging, KStar, Random SubSpace-RSS, K Nearest Neighbor-KNN) eğitilip ardından test edilmiştir. Çalışmanın

doğruluğu artırmak için Çapraz Geçerleme (Cross-Validation) yöntemi n=10 olacak şekilde uygulanmıştır.

Sonuçların değerlendirilmesi için 5 temel değerlendirme metriği (doğruluk, duyarlılık, anma, F-Ölçütü ve Kappa) kullanılmıştır. Doğruluk metriği için elde edilen sonuçlar Tablo 4'te, kullanılan metriklere ait denklemler ise Eşitlik 2-6'da verilmiştir.

$$Doğruluk(Acuuracy) = \frac{(True\ Positive(TP) + True\ Negative(TN))}{(TP + False\ Positive(FP) + False\ Negative(FN) + TN)} \quad (2)$$

$$Kesinlik(Precision) = \frac{TP}{(TP + FP)} \quad (3)$$

$$Duyarlılık(Recall) = \frac{TP}{(TP + FN)} \quad (4)$$

$$F - Ölçütü(F - Measure) = \frac{2 * Kesinlik * Duyarlılık}{(Kesinlik + Duyarlılık)} \quad (5)$$

$$Kappa = \frac{(P_o - P_c)}{(1 - P_c)} \quad (6)$$

(P_o *kabul edilen oran*, P_c *kabul edilmesi beklenen oran*)

Tablo 4. Deneysel Sonuçlar-1 (Doğruluk-Accuracy)

		RT	NB	Bagging	KStar	RSS	KNN	Ortalama
Ham Metin	2-Gram	30,56	36,11	33,33	25	30,56	25	30,09333
	3-Gram	25	47,22	22,22	16,67	22,22	19,44	25,46167
İkili Sıkıştırma	2-Gram	**36,11**	**52,78**	**44,44**	**36,11**	**50**	**38,89**	**43,055**
	3-Gram	**33,33**	**50**	**58,33**	**38,89**	**50**	**36,11**	**44,44333**
Onlu Sıkıştırma	2-Gram	19,44	22,22	38,89	33,33	36,11	30,56	30,09167
	3-Gram	22,22	19,44	25	19,44	25	22,22	22,22

Hem 2-gram hemde 3-gram için ikili sıkıştırma kullanılan tüm makine öğrenmesi algoritmalarında ham metin ve onlu sıkıştırmaya göre daha yüksek başarı oranlarına ulaşmıştır.Onlu sıkıştırma Bagging ve KStar algoritmaları dışında ham metinin gerisinde kalmıştır. Genel ortalamalarda ise ikili sıkıştırma, ham metin ve onlu sıkıştırma şeklindedir.

Doğruluk metriği dışında Kesinlik, Duyarlılık, F-Ölçütü ve Kappa metrikleri de hesaplanmıştır. Bu metriklerde doğruluk metriğine paralel sonuçlar üretmiştir. Binary sıkıştırma yöntemi bu metrikler içinde en iyi değerlere ulaşmayı başarmıştır. 6 algoritma için metriklerin sonuçlarına vermek yerine İkili sıkıştırma (3-gram) için en yüksek doğruluk değerine ulaşılan Bagging ve en düşük başarı değerlerine ulaşılan RT algoritması için Kesinlik, Duyarlılık, F-Ölçütü ve Kappa metriğine ait sonuçlar Tablo 5 'te ve karışım matrisleri (Confusion Matrix) Tablo 6'te verilmiştir.

Tablo 5. Deneysel Sonuçlar-2 (Diğer Metrikler- Bagging ve Random Tree)

Bagging		Kesinlik	Duyarlılık	F-Ölçütü	Kappa
Ham Metin	2-Gram	0,286	0,333	0,301	0,2
	3-Gram	0,195	0,222	0,203	0,0677
İkili Sıkıştırma	2-Gram	**0,372**	**0,444**	**0,403**	**0,3333**
	3-Gram	**0,487**	**0,583**	**0,527**	**0,5**
Onlu Sıkıştırma	2-Gram	0,356	0,389	0,358	0,2667
	3-Gram	0,245	0,25	0,242	0,1
Random Tree		Kesinlik	Duyarlılık	F-Ölçütü	Kappa
Ham Metin	2-Gram	0,306	0,306	0,297	0,1667
	3-Gram	0,225	0,25	0,228	0,1
İkili Sıkıştırma	2-Gram	**0,319**	**0,361**	**0,333**	**0,2333**
	3-Gram	**0,357**	**0,333**	**0,339**	**0,2**
Onlu Sıkıştırma	2-Gram	0,185	0,194	0,187	0,0333
	3-Gram	0,192	0,222	0,203	0,0667

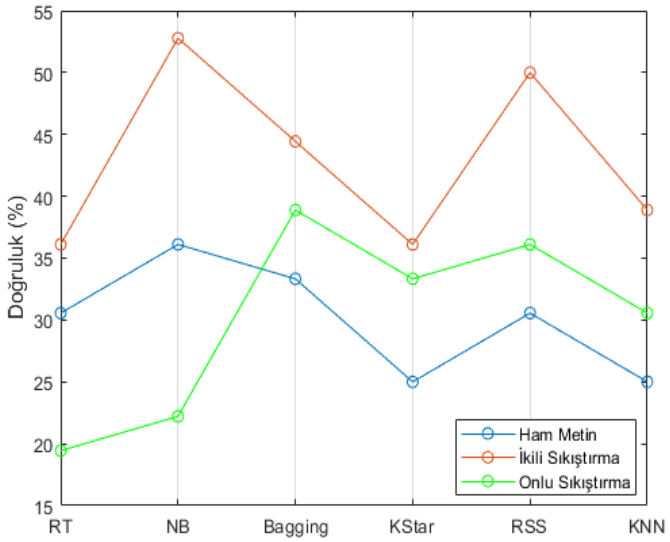
Kesinlik, Duyarlılık, F-Ölçütü ve Kappa metrikleri için en yüksek değerlere doğruluk metriğinde olduğu gibi İkili sıkıştırma için ulaşılmıştır. Ham metin için elde edilen değerlerde Onlu sıkıştırmanın üstünde çıkmıştır.

Tablo 6. Karışım Matrisi

Bagging	Yazar A	Yazar B	Yazar C	Yazar D	Yazar E	Yazar F
Yazar A	**4**	1	0	1	0	0
Yazar B	0	**2**	2	0	2	0
Yazar C	0	1	**5**	0	0	0
Yazar D	2	1	0	**0**	1	2
Yazar E	1	0	0	0	**4**	1
Yazar F	0	0	0	0	0	**6**
RT	Yazar A	Yazar B	Yazar C	Yazar D	Yazar E	Yazar F
Yazar A	**1**	1	0	4	0	0
Yazar B	1	**2**	1	1	0	1
Yazar C	0	1	**5**	0	0	0
Yazar D	2	1	0	**1**	1	1
Yazar E	1	2	0	0	**2**	1
Yazar F	0	1	0	3	1	**1**

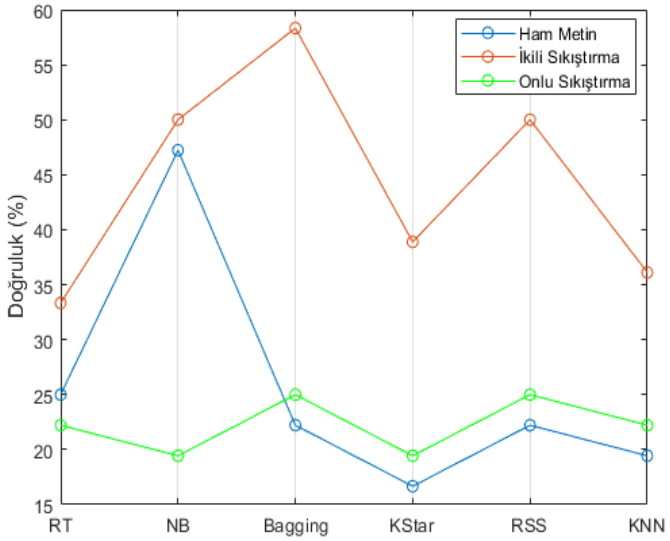
Bagging algoritmasında Yazar F'ye ait tüm metinler doğru şekilde sınıflandırılırken yazar D'nin hiçbir metni sınıflandırılamamıştır. Random Tree algoritmasında ise en yüksek doğruluk değeri Yazar C için 5 olarak ölçülürken her yazarın en az bir metni doğru sınıflandırılmıştır.

2 ve 3 gram için elde edilen doğruluk metriği sonuçları grafiksel olarak Şekil 3 ve Şekil 4'te gösterilmektedir.



Şekil 3: *2-Gramlar için Doğruluk Metriği*

2-gramlar için İkili sıkıştırma tüm algoritmalarda en yüksek doğruluk değerlerine ulaşırken, RT ve NB'de geride kalsa da kalan diğer 4 algoritmada onlu sıkıştırama ve ham metin üçüncü sırada kalmıştır. Genel Ortalamalar ise ham metin ve onlu sıkıştırmanın birbirine yakın değerlerdir.



Şekil 4: *3-Gramlar için Doğruluk Metriği*

3-gramlar için İkili sıkıştırma tüm algoritmalarda en yüksek doğruluk değerlerine ulaşırken, RT ve NB'de geride kalsa da kalan diğer 4 algoritmada ham metin ikinci ve onlu sıkıştırma üçüncü sırada kalmıştır.

4. SONUÇLAR VE TARTIŞMA

Bu çalışma 6 farklı köşe yazarına ait 6 farklı toplamda 36 adet köşe yazısının sınıflandırılması üzerinedir. En sık kullanılan metin temsil yöntemi olan n-gram ile ifade edilmiş metinler yine literatürde en sık kullanılan makine öğrenmesi algoritmaları ile eğitim ve teste tabi tutulmuştur. Çalışmanın yenilikçi yönü ön işlemden geçen metinlerin LZW algoritması ile kayıpsız sıkıştırılıp boyut sayısının azaltılması ve böylece sınıflandırma başarısının artırılması üzerinedir.

Bir metnin tamamını temsil yöntemleri ifade edip sınıflandırmak yerine o metni eksiksiz şekilde temsil edecek bir özetinin olması o metnin sınıfı hakkında bize bilgi verebilir savı ile önerilen yöntem çalışma sonunda mevcut yöntemlere göre başarılı sonuçlar elde etmeyi başarmıştır.

Metinlerin ön işlemlerden geçmesinin ardından LZW sıkıştırma algoritmasının iki farklı temsil şekli bu çalışmada kullanılmıştır. İkili (binary) gösterimde çıkan metinler sıkıştırma esnasında ikili olarak kodlanmaktadır. Onlu (decimal) gösterimde ise çıkan metinler sıkıştırma esnasında onlu olarak kodlanmaktadır.

Yapılan çalışmalar sonucunda ikili sıkıştırılmış metinler ile yapılan sınıflandırma başarıları hem ham metinden hem de onlu sıkıştırmadan daha iyi sonuçlar elde etmiştir. Böylece klasik yöntem diyebileceğimiz ham metine göre doğruluk, kesinlik, duyarlılık, f-ölçütü ve kappa metriklerinde neredeyse iki katlık bir artış sağlanmıştır. Aynı durum onlu sıkıştırma için geçerli değildir ama yine de mevcutta kullanılan altı farklı makine öğrenmesi içinden dört tanesinde ham metine göre daha yüksek başarılar göstermeyi başarmıştır. Ama ortalama başarı oranında azda olsa ham metnin gerisinde kalmıştır.

Metin temsilinde 2-gram kullandığında en yüksek metrik değerlerine ikili sıkıştırma ve ham metin NB, onlu sıkıştırma ise Bagging algoritması ile ulaşmıştır. Metin temsilinde 3-gram kullandığında en yüksek metrik değerlerine ikili sıkıştırma Bagging, onlu sıkıştırma Bagging ve RSS algoritması, ham metin ise NB ile ulaşmıştır. Sistem en yüksek başarı değerine 3 gram için ikili sıkıştırma için Bagging algoritması ile 58.33(%)'lük doğruluk değerine ulaşmayı başarmış ve 21 metinin sınıfını doğru tespit etmeyi başarmıştır.

Geleceğe yönelik çalışmalar kapsamında kullanılan metin ve sınıf sayısının artırılması ve bu durumda ikili ve onlu sıkıştırmanın performansının analiz edilmesi hedeflenmektedir. Ayrıca farklı metin temsil yöntemleri kullanılması da hedeflerimiz arasındadır. Yazar tanımanın yanı sıra duygu analizi gibi metin sınıflandırma problemlerinde önerilen yöntemin denenmesi hedeflenmektedir.

KAYNAKLAR

1. Amasyalı .M.F. ve Yıldırım T. (2004) Otomatik Haber Metinleri Sınıflandırma, 12.IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Kuşadası, Aydın, Türkiye, 224-226.

2. Amasyalı, M. F. ve Diri, B. (2006) Automatic Turkish Text Categorization in Terms of Author, Genre and Gender, International Conference on Applications of Natural Language to Information Systems, Klagenfurt, Austria, 221-226. Doi: 10.1007/11765448_22

3. Amasyalı, M.F., Davletov, F., Arslan, T. ve Çiftçi, Ü. (2010) Text2arff: Automatic feature extraction software for Turkish texts, 18.IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Diyarbakır, Türkiye, 629-632. Doi: 10.1109/SIU.2010.5651686

4. Bekkerman R., El-Yaniv, R., Naftali T. ve Yoad W. (2002) Distributional Word Clusters vs. Words for Text Categorization, *Journal of Machine Learning Research*, 3,1-48. Doi: 10.1.1.19.7938

5. Bilgin, M. (2018) Makine Öğrenmesi Teorisi ve Algoritmaları, Papatya Bilim, İstanbul.

6. Bilgin, M. ve Şentürk, İ.F. (2017) Sentiment analysis on Twitter data with semi-supervised Doc2Vec, International Conference Computer Science and Engineering, 661-666. Doi: 10.1109/UBMK.2017.8093492

7. Ciya L., Shamim A., ve Paul D. (2001) Feature Preparation in Text Categorization, *Oracle Text Selected Papers and Presentations*, 1-8.

8. Çatal Ç., Erbakırcı K. ve Erenler Y. (2003) Computer-based Authorship Attribution for Turkish Documents, Turkish Symposium on Artificial Intelligence and Neural Networks, Çanakkale,Türkiye, 539-541.

9. Çiltik, A. ve Güngör, T. (2008) Time-Efficient Spam E-mail Filtering Using N-gram Models, *Pattern Recognition Letters*, 29(1), 19-33. Doi: 10.1016/j.patrec.2007.07.018

10. Çoban, Ö., Ö. B. ve Özyer, G.T. (2015) Sentiment analysis for Turkish Twitter feeds. 23.IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Malatya, Türkiye, 2388-2391. Doi: 10.1109/SIU.2015.7130362

11. Diri B. ve Amasyalı M.F. (2003) Automatic Author Detection for Turkish Texts, Artificial Neural Networks and Neural Information Processing, İstanbul, Türkiye, 138-141.

12. Doğan, S. (2006) Türkçe Dokümanlar için N-gram Tabanlı Sınıflandırma: Yazar, Tür ve Cinsiyet, Yıldız Teknik Üniversitesi, Yüksek Lisans Tezi, İstanbul.

13. Doğan, S. ve Diri, B. (2010) Türkçe Dokümanlar İçin N-gram Tabanlı Yeni Bir Sınıflandırma (Ng-ind): Yazar, Tür ve Cinsiyet, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 3(1), 11-19.

14. Fattah, M.A. (2017) A Novel Statistical Feature Selection Approach for Text Categorization, *Journal of Information Processing Systems*, 13(5), 1397-1409. Doi: 10.3745/JIPS.02.0076

15. Fürnkranz J. (1998) A Study using n-gram Features for Text Categorization, *Austrian Research Institute for Artifical Intelligence*, 3(1998), 1-10. Doi: 10.1.1.49.133

16. Gaines, B.J. ve Carney, H.E. (2018) Communication and management of electronic mail classification information, U.S. Patent No. 9,942,184.

17. Gezici, G. ve Yanıkoğlu, B. (2018) Sentiment Analysis in Turkish, Turkish Natural Language Processing. Springer, Cham, 255-271. Doi: 10.1007/978-3-319-90165-7_12

18. Holmes, D. I. (1998) The Evolution of Stylometry in Humanities Scholarship, *Literary and Linguistic Computing*, 13 (3), 111-117. Doi: 10.1093/llc/13.3.111

19. Johnson, R. ve Zhang, T. (2017) Deep pyramid convolutional neural networks for text categorization, 55.Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada,562-570. Doi: 10.18653/v1/P17-1052

20. Kemik, (2019). Kemik-Veri Kümelerimiz. Erişim Adresi: http://www.kemik.yildiz.edu.tr/data/File/2500koseyazisi.rar (Erişim Tarihi: 12.02.2019)

21. Levent, V.E. ve Diri, B. (2014) Türkçe Dokümanlarda Yapay Sinir Ağları İle Yazar Tanıma, Akademik Bilişim, Mersin, Türkiye.

22. Özel, B. (2004) Küresel k-Ortalamalı Gruplama Yöntemi ile Metinlerin ve Terimlerin Saklı Anlam İndekslenmeleri, Akıllı Sistemlerde Yenilikler ve Uygulamaları Konferansı, İstanbul, Türkiye, 223-227.

23. Peng F. ve Schuurmans D. (2003) Combining Naive Bayes and N-gram Language Models for Test Classification, European Conference on Information Retrieval, Berlin, Almanya, 335-350. Doi: 10.1.1.2.1184

24. Peng F., Keselj V., Cerconey N. ve Thomasy C. (2003) N-Gram-Based Author Profiles For Authorship Attribution, The conference pacific association for computational linguistics Nova Scotia, Canada,255-264. Doi: 10.1.1.9.7388

25. Song, F., Liu, S. ve Yang, J. (2005) A comparative study on text representation schemes in text categorization, *Pattern analysis and applications*, 8(1-2), 199–209. Doi: 10.1007/s10044-005-0256-3

26. Stamatatos E., Fakotakis N. ve Kokkinakis G. (2000) Automatic Text Categorization in Terms of Genre and Author, *Computational Linguistics*, 26(4), 471-495. Doi: 10.1162/089120100750105920

27. Tan C.M., Wang Y.F. ve Lee C.D. (2002) The Use of Bi-grams to Enhance Text Categorization, *Journal Information Processing and Management*, 30(4), 529-546. Doi: 10.1016/S0306-4573(01)00045-0

28. Türkoğlu F., Diri B. ve Amasyalı M.F. (2006) Farklı Özellik Vektörleri ile Türkçe Dokümanların Yazarlarının Belirlenmesi, Turkish Symposium on Artificial Intelligence and Neural Networks, Muğla, Türkiye.

29. Türkoğlu, F., Diri, B. ve Amasyalı, M. F. (2007) Author Attribution of Turkish Texts by Feature Mining, 3.International Conference on Intelligent Computing, Qingdao, China, 1086-1093. Doi: 10.1007/978-3-540-74171-8_110

30. Welch, T. A. (1984) A Technique for High-Performance Data Compression, *IEEE Computer*, 17(6), 8-19. Doi: 10.1109/MC.1984.1659158