SuDer Türkçe Haber Derlemlerinin Doküman Sınıflandırması
Document Classification of SuDer Turkish News Corpora

Mehmet Umut Şen
Sabancı Üniversitesi
umutsen@sabanciuniv.edu

Berrin Yanıkoğlu
Sabancı Üniversitesi
berrin@sabanciuniv.edu

***Özetçe* —Kelime Temsil Vektörleri, Doğal Dil İşleme alanındaki çeşitli problemlere başarılı bir şekilde uygulanmaktadır; ancak bu vektörleri eğitmek için büyük miktarda metin verisi gereklidir. İngilizce için metin derlemi pek çok farklı konu ve boyut için rahatlıkla bulunsa da, Türkçe için az sayıda derlem bulunmaktadır. Bu çalışmada iki çevrimiçi haber sitesinden büyük miktarlı metin derlemleri toplanmış ve etiket olarak internet sayfalarında bulunan kategori bilgisi kullanılmıştır. Oluşturulan derlemler çeşitli doküman sınıflandırma modelleri ile denenmiştir. Temsil vektörleri kullanan modellerin, geleneksel TF-TDF özniteliklerini kullanan yöntemlerden daha iyi sonuç verdiği görülmüştür. Aynı anda hem kelime vektörlerini hem de doküman sınıflandırmasını öğrenen bir yapay sinir ağı en iyi sonucu vermiştir.**

Anahtar Kelimeler—doküman sınıflandırma, SuDer haber metinleri, kelime temsil vektörleri, yapay sinir ağları

***Abstract*—Word embeddings are successfully employed in various Natural Language Processing tasks, but training them requires large amount of text, which is scarce for Turkish. In this work, we collected large amounts of articles from two news websites and tags within web pages are used as labels. Obtained corpora are tested with various document classification models. Embedding based models performed better than models with the traditional TF-IDF features. A neural network that simultaneously learns the word embeddings and document classification performed the best.**

Keywords—document classification, SuDer news corpora, word embeddings, neural networks

I. Giriş

Metinlerin otomatik olarak kategorilerine ayrılması olarak tanımlanabilen doküman sınıflandırma probleminin; konu sınıflandırma, tür sınıflandırma, istenmeyen elektronik posta filtreleme, duygu analizi gibi uygulama alanları vardır. Bu bildiride konu ve tür sınıflandırma uygulaması üzerinde çalışılmıştır.

Doküman sınıflandırma için geleneksel yöntemler, dokümanların içinde geçen kelimelerin istatistiklerini kullanarak öznitelik çıkarmaya ve bu öznitelikleri bir makine öğrenimi yöntemiyle modellemeye dayanır. Terim Frekansı-Ters Doküman Frekansı (TF-TDF) yüksek başarı oranı ile en popüler öznitelik çıkarımı yöntemlerinden biridir. Literatürdeki bu ve benzer yöntemlerin ve makine öğrenimi modellerinin farklı varyasyonlarının bir derlemesi Jindal ve arkadaşlarının makalesinde bulunabilir [1].

Kelime Temsil Vektörleri, kelimelerin düşük boyutlu sayısal vektörlerle temsilidir ve son yıllarda pek çok problemde kullanım alanı bulmuşlardır. Bu vektörler girdi olarak Yapay Sinir Ağlarına (YSA) verilip, Geri Yayılım algoritmasıyla güncellenerek eğitilirler. Goldberg, yapay sinir ağı modellerinin Doğal Dil İşleme (DDİ) problemleri üzerine çözümlerini incelemiştir [2].

Mikolov ve arkadaşlarının yaptığı bir çalışmada etiketsiz ve büyük derlemlerde, girdi olarak metinden bir kelime alan ve bu kelimenin yakınındaki kelimeleri kestirmeye çalışan, tek katmanlı bir modelden elde edilen vektörlerin kelimelerle ilgili anlam-bilimsel ve söz-dizimsel bilgileri içerdiği gösterilmiştir [3]. Büyük etiketsiz derlemlerden bu şekilde öğrenilen kelime vektörleri, küçük etiketli derlemlerde eğitilecek modellerin ilklendirilmesi için kullanılabilmektedir.

Türkçe metinlerin sınıflandırılması için literatürde çeşitli çalışmalar mevcuttur. Kılıç ve arkadaşlarının çalışmasında TF-TDF'nın iki yeni varyasyonu tanıtılmış ve Türkçe derlemde başarımın arttığı gösterilmiştir [4]. Ay ve arkadaşlarının çalışmasında genetik algoritma kullanılmış ve yeni bir nitelik ağırlıklandırma yöntemi sunulmuştur [5]. Şahin'in çalışmasında gözetimsiz öğrenilen kelime temsil vektörlerinin ortalamaları Destek Vektör Makinesine (DVM) girdi olarak verilmiş ve TF-TDF'ndan daha iyi başarı sağladığı gösterilmiştir [6]. Bu bildiride bu son çalışmadaki yöntem tekrarlanmış, ayrıca kelime vektörlerinin gözetimli öğrenilmesinin başarıyı daha da artırdığı gösterilmiştir.

Doküman sınıflandırma için Türkçe derlemler gün geçtikçe artmaktadır. Şahin ve arkadaşlarının çalışmasında Türkçe Vikipedi sayfaları otomatik kategorilenmiş ve yaklaşık 10 milyon kelimeli bir derlem oluşturulmuştur [7]. Tüfekçi ve arkadaşlarının çalışmasında 5 farklı haber portalından toplanmış 5 kategoriden oluşan toplam 750 dokümanlık derlem oluşturulmuş ve çeşitli morfolojik önişleme yöntemlerinin

Mehmet Umut Şen is supported by a TÜBİTAK Bideb-2211-A scholarship.

978-1-5386-1501-0/18/$31.00

sınıflandırmaya etkisi incelenmiştir [8]. Kelime haznesinde sadece isim türündeki kelimelerin kullanılmasıyla öznitelik boyutlarının yüksek oranda düşürüldüğü ve başarımın azalmadığı gösterilmiştir. Kılınç ve arkadaşlarının çalışmasında, 6 haber portalından toplanmış $3,600$ dokümandan oluşan bir derlem paylaşılmıştır [9].

Kelime temsillerinin öğrenimi, büyük metin derlemlerini gerektirmektedir. Bu nedenle, bu çalışmada büyük ölçekli iki yeni derlem toplanmıştır. Bu derlemler üzerinde TF-TDF, Saklı Dirichlet Ataması (SDA), Kelime Temsil Vektörleri ve Yapay Sinir Ağları kullanan doküman sınıflandırma yöntemleri uygulanmış ve değerlendirilmiştir.

II. Derlemler

Sabah[1] ve Cumhuriyet[2] gazetelerinin çevrimiçi internet sitelerinden metin içerikli haber, köşe yazısı, resim galerisi ve video paylaşımı içeren sayfalar indirilmiş ve bu sayfalardan metin, başlık, tarih ve kategori bilgileri ayıklanmıştır.

Sabah'ın sitesinden 2010-Ocak ile 2017-Temmuz arasında yayınlanmış toplamda yaklaşık $426,000$ sayfa elde edilmiş; metin ve başlıktaki toplam kelime sayısı 10'dan az olan sayfalar elenmiş ve geriye $420,513$ sayfa kalmıştır. Toplamda 4 farklı kategori vardır ve bu kategorilerle ilgili bilgiler Tablo-I'de belirtilmiştir.Bu istatistikler başlıklar kullanılmadan çıkarılmıştır. Deneylerde de başlıklar kullanılmamaktadır.

Cumhuriyet'in sitesinden 2017-Eylül tarihine kadar yayınlanan, toplamda yaklaşık $463,000$ sayfa elde edilmiştir. Ancak 2014 senesinden önceki sayfaların çoğunda kategori bilgisi bulunmamaktadır; toplamda $273,000$ sayfanın kategori bilgisi mevcuttur. Metindeki kelime sayısı 10'dan az olan ve toplam sayfa sayısı az olan 7 kategoriye ait sayfalar elendikten sonra 14 kategoriye ait $268,784$ sayfa elde edilmiştir. Kategoriler ile ilgili bilgiler Tablo-II'de verilmiştir.

TABLO I: Sabah Derlemi İstatistikleri

Kategori	Doküman Sayıları			Kelime Sayıları	
	Toplam	Eğitim	Test	Toplam	Ortalama
gündem	143,842	117,019	26,823	35,749,880	248.54
yaşam	123,086	108,202	14,884	22,878,732	180.86
ekonomi	85,485	75,512	9,973	22,261,600	247.38
yazarlar	68,100	60,683	7,417	16,335,364	239.87
Toplam	420,513	361,416	59,097	95,494,110	227.09

TABLO II: Cumhuriyet Derlemi İstatistikleri

Kategori	Doküman Sayıları			Kelime Sayıları	
	Toplam	Eğitim	Test	Toplam	Ortalama
türkiye	84,741	56,140	28,524	22,829,220	269.39
yazarlar	33,835	29,694	4,141	16,663,717	492.49
video	33,409	23,686	9,723	2,007,691	60.09
spor	31,396	24,627	6,730	7,240,974	230.63
dünya	21,005	14,684	6,152	4,416,708	210.26
siyaset	15,969	11,274	4,686	6,409,811	401.39
foto	14,302	9,729	110	248,871	17.40
ekonomi	8,187	5,811	2,356	2,520,473	307.86
teknoloji	7,913	5,089	2,810	1,734,268	219.16
kültür-sanat	6,506	4,680	1,806	2,664,020	409.47
yaşam	4,833	3,931	886	918,754	190.10
sağlık	2,573	2,047	514	863,208	335.48
eğitim	2,380	1,544	805	744,396	312.77
çevre	1,735	1,081	607	477,811	275.39
Toplam	268,784	194,017	69,850	69,739,922	259.46

[1]www.sabah.com.tr

[2]www.cumhuriyet.com.tr

Bu çalışmamızda bu iki derlemin de 1 Eylül 2016'dan önceki dokümanları eğitim kümesi, sonrakiler ise test kümesi olarak kullanılmıştır. Kelime haznesine, kesme işareti ile ayrılmış ekler dahil edilmiş; tek harfli kelimeler ve sayılar dahil edilmemiştir[3].

III. Yöntemler

A. TF-TDF ve Destek Vektör Makineleri

Terim Frekansı - Ters Doküman Frekansı (TF-TDF) öznitelikleri, her bir dokümanı sabit boyutta sayısal vektörler şeklinde gösterebilen bir yöntemdir. Vektörlerdeki her boyut bir terimin dokümanda geçme sıklığına dayanır. t teriminin d dokümanındaki görülme sayısına c_{dt} ve d dokümanındaki toplam kelime sayısına N_d dersek, Terim Frekansı şu şekilde bulunur: $tf(d,t) = c_{dt}/N_d$. Çok fazla sayıda dokümanda geçen, dolayısıyla bağlamla ilgisi olma ihtimali düşük kelimelerin etkisini azaltmak amacıyla da bir terimin Ters Doküman Frekansı şu şekilde tanımlanır:

$$tdf(t) = \log\left(\frac{1+D}{1+m_t}\right) \quad (1)$$

Burada, D toplam doküman sayısı, m_t ise t teriminin geçtiği doküman sayısıdır. TF-TDF öznitelikleri bu iki değerin çarpımıdır: $tftdf(d,t) = tf(d,t) \times tdf(t)$.

Bu çalışmada terim olarak sadece tekli kelimeler kullanılmıştır. TF-TDF öznitelikleri bulunurken kelime hazne boyu için $1,000$ ile $50,000$ arasında değişen farklı değerler denenmiştir. Terim vektörlerini normalize etmek için l_1 normalizasyonu kullanılmıştır. Çıkarılan öznitelikler doğrusal Destek Vektör Makinesi (DVM) ile sınıflandırılmıştır. Veri sayısının öznitelik sayısından çok olduğu durumlarda doğrusal DVM'nın birincil formülasyonunun optimizasyonunun doğrusal olmayan çekirdekli formülasyonlara göre çok daha hızlı olduğu ve doğruluk oranlarında yakın sonuç verdiği için [10] doğrusal DVM kullanılmış ve birincil formülasyonla optimize edilmiştir. Çok sınıfla sınıflandırma için "bire-hepsi" yöntemi [11] kullanılmıştır.

B. Saklı Dirichlet Ataması

Saklı Dirichlet Ataması (SDA) gözetimsiz konu öğrenimi için sık kullanılan üretici bir olasılıksal modeldir [12]. Bu yöntemde her bir doküman bir konuya atanmak yerine bir konu dağılımına atanır ve bu atama Dirichlet Dağılımı ile temsil edilir. Doküman içindeki her bir kelimenin ise tek bir konudan geldiği varsayılır. Konular ise sözcük haznesindeki kelimeler üzerine bir ihtimal dağılımı ile temsil edilir. Konu sayısı modele girdi olarak verilir.

Bu çalışmada SDA modelindeki önceden sabitlenmesi gereken toplam konu sayısı (K) için farklı değerler denenmiştir. Veri büyük olduğu için Varyasyonel Bayes yöntemiyle çıkarsama yapan Çevrimiçi SDA [13] yöntemi kullanılmıştır[4]. Model eğitildikten sonra, gözetimli sınıflandırmada kullanılmak üzere, her konu bir kategoriye atanmıştır. Bu atamayı belirlemek için, eğitim verisindeki her bir doküman için konu dağılımları bulunmuş (γ_{dk}:

[3]Derlemler şu adresten indirilebilir: https://github.com/suverim/suder

[4]SDA kodu: github.com/wellecks/online_lda_python

d dokümanının k konusuna ait olma ihtimali); her konu, ihtimallerinin ortalaması en yüksek kategoriye atanmıştır:

$$m_k = \arg\max_c \frac{1}{|\mathcal{D}_c|} \sum_{d:d\in\mathcal{D}_c} \gamma_{dk} \quad (2)$$

Burada $\mathcal{D}_c$, c sınıfına ait doküman kümesi; m_k, k konusunun hangi sınıfa ait olduğudur.

C. Kelime Temsilleri ve Destek Vektör Makineleri

Kelime haznesindeki her bir kelimenin, hazne boyuna kıyasla çok daha küçük boyutlu, rasyonel vektörlerle temsil edilmesine *kelime temsili* denir. Büyük veri tabanlarında gözetimsiz öğrenilen vektörlerin, kelimelerle ilgili anlam-bilimsel ve söz-dizimsel bilgileri yakalayabildiği gözlemlenmiştir [3], [14]. Bu çalışmada kelime vektörlerinin gözetimsiz öğrenimi için Atla-Gram yöntemi kullanılmıştır [3], [15]. Bu yöntemde her kelimenin "girdi" ve "çıktı" vektörleri bulunmaktadır. Modele girdi olarak bir kelimenin "girdi" vektörü verilir ve yakındaki kelimelerin "çıktı" vektörlerinin kestirimi, çıktı katmanındaki yumuşak-maksimum katmanı ile hedeflenir. Eğitimden sonra "girdi" vektörü kelimenin temsili için kullanılır. Standart formülasyon pratikte çalışmadığı için geliştirilen yakınlaştırmalardan *Eksi-Örnekleme* yöntemi kullanılmıştır.

Bu çalışmada, kelime vektörleri bulunduktan sonra doküman özniteliklerini bulmak için dokümandaki kelime temsillerinin ortalaması alınmıştır. Daha sonra çıkan öznitelikler DVM'ne girdi olarak verilmiştir. Başka bir çalışmada, bu yöntemin Türkçe bir derlemde iyi çalıştığı görülmüştür [6].

D. Kelime Temsilleri ve Yapay Sinir Ağları

Bu yöntemde dokümandaki kelime temsillerinin ortalaması alındıktan sonra YSA ile konu sınıflandırması yapılmıştır. Bir t kelimesinin vektörü $\mathbf{w}_t \in \mathbb{R}^d$ ve $f : \mathbb{R}^d \to \mathbb{R}^C$ bir YSA olmak üzere (C sınıf sayısı), verilen bir $S_d = \{t_1, \ldots, t_{N_d}\}$ dokümanının sınıflandırması şu şekilde yapılır:

$$y_c(d) = f\left(\frac{1}{|S_d|}\sum_{t\in S_d} \mathbf{w}_t\right) \quad (3)$$

Burada $y_c(d)$, d dokümanının c sınıfına ait skorudur. Hedef fonksiyonu olarak Ortalama Kareler Toplamı kullanılmıştır:

$$\Phi = \frac{1}{CD}\sum_{d=1}^{D}\sum_{c=1}^{C}(y_c(d) - \delta_{dc})^2 \quad (4)$$

Burada δ_{dc}, d dokümanı c sınıfına aitse 1, diğer durumlarda 0'dır ve D toplam doküman sayısıdır. Bu çalışmada, önceki benzer Türkçe doküman sınıflandırma yöntemlerinden (örn. [6]) farklı olarak, kelime vektörleri de geri yayılım algoritması kullanılarak güncellenmiştir. Böylece daha önce gözetimsiz öğrenilen kelime vektörlerinin etiket bilgisi kullanılarak ayrıştırıcı eğitimi sağlanmış ve bunun doğruluk oranlarını artırdığı gözlenmiştir. Önceki bölümde bahsedilen gözetimsiz öğrenilen kelime vektörleri, ağın kelime vektörlerinin ilklendirilmesi için kullanılmıştır.

IV. Deneyler

Metinler modellere verilmeden önce küçük harflere dönüştürülmüş; daha sonra özel isimlere eklenen ekleri yakalamak için aralarında kesme işareti bulunan kelimeler ayrılmış ve bu ekler atılmıştır. Sonrasında tek harfli kelimeler ve sayılar atılmıştır. Bağlam dışı kelimeler de, internetteki çeşitli kaynaklar kullanılarak atılmıştır [5] [6] [7]. Toplamda 553 tane bağlam dışı kelime elde edilmiştir.

Morfolojik işlem için Zemberek araç kutusu [16] kullanılarak kelimelerin morfolojik analizi yapılmış ve analizi yapılabilen kelimelerin analiz seçeneklerinden en uzun köklü olanın kökü kullanılmıştır. Bu yöntemin daha önce iyi sonuç verdiği literatürde gözlemlenmiştir [8], [14], [17].

A. Parametreler

TF-TDF vektörleri, $1,000$ ile $50,000$ arasında değişen kelime hazne boyu için çıkarılmıştır. Gerçekleme için *Gensim* araç kutusu kullanılmıştır [18]. Kelime haznesi bulunurken derlemde toplamda en sık geçen kelimeler kullanılmıştır. DVM gerçeklemesi için *scikit-learn* araç kutusu [19] ve C parametresi için varsayılan değer kullanılmıştır. Farklı kelime haznesi boylarına göre sonuçlar Tablo-III'te verilmiştir.

TABLO III: TF-TDF Hazne Boyunun Etkisi(%)

Derlem/Hazne Boyu	1K	5K	10K	20K	50K
Sabah	84,29	86,22	86,41	86,52	86.50
Cumhuriyet	69,12	71,71	71,81	71,72	71,69

Sonuçlara göre her iki derlemde de kelime haznesi boyu olarak $10,000$ ile $20,000$ civarında iyi sonuçlar elde edildiği ve bu boyu daha fazla arttırmanın doğruluk oranlarına bir faydası olmadığı görülmektedir. Bu sonuçlara bağlı olarak, Saklı Dirichlet Ataması (SDA) deneylerinde her iki derlem için de hazne boyu $10,000$ alınmıştır.

Çevrimiçi Saklı Dirichlet Ataması (SDA) yöntemindeki, ilk verilen dokümanların etkisini azaltmak için olan öğrenme parametresi (τ) 1024, düşüş faktörü parametresi (κ) 0.7 alınmıştır. Toptan boyutu olarak 100 kullanılmış ve eğitim verisinin üzerinden toplamda 3 devir yapılmıştır. Konu sayısı için, derlemlerdeki sınıf sayısı ve daha yüksek değerler denenmiştir.

Kelime vektörlerinin gözetimsiz öğrenimi için *Gensim* araç kutusu kullanılmıştır [18]. Pencere boyutu 20, Eksi-Örnekleme parametresi 5 alınmıştır. Eğitim derleminin üzerinden 20 kere geçilmiştir. Vektör boyutları için 100, 200, 400 ve 600 denenmiştir. Derlemde 10'dan az geçen kelimeler elenmiş ve geriye Cumhuriyet derlemi için $70,118$, Sabah derlemi için $60,718$ kelime kalmıştır.

YSA modelinde, 50 düğümlü ve doğrusal olmayan aktivasyon fonksiyonu ReLU olan 2 tane saklı katman kullanılmıştır. Çıktı katmanının aktivasyonu için S-biçim fonksiyonu kullanılmıştır. Optimizasyon algoritması olarak RMSprop, öğrenme oranı için 0.01 kullanılmıştır. Eğitim verisi üzerinde toplamda 10 devir yapılmıştır. Toptan

[5] https://github.com/ahmetax/trstop/blob/master/dosyalar/turkce-stop-words

[6] https://github.com/crodas/TextRank/blob/master/lib/TextRank/Stopword/turkish-stopwords.txt

[7] https://github.com/stopwords-iso/stopwords-tr/blob/master/stopwords-tr.txt

boyutu 100 alınmış ve bu toptanlar yerine koyarak rastgele örneklemeyle oluşturulmuştur. Bu model *Pytorch* araç kutusuyla gerçeklenmiştir [20].

B. Sonuçlar

Deneyi yapılan yöntemlerin doğruluk oranları Tablo-IV'te gösterilmiştir. Etiket bilgisi kullanmayan Saklı Dirichlet Ataması en düşük sonuçları vermiştir; ancak SDA'nın büyük miktarlarda etiketsiz verinin de olduğu durumlarda daha iyi sonuçlar vermesi beklenir. Ayrıca Cumhuriyet derleminde en iyi sonuç sınıf sayısına eşit konu sayısı ile elde edilmesine rağmen, Sabah derleminde konu sayısını artırmak doğruluk oranını artırmıştır. Bu sonucun muhtemel sebebi Sabah derleminde sadece 4 konu kategorisi olması, dolayısıyla metinlerin konularına çok bağlı olmamasıdır.

Gözetimli modellerde DVM ile birlikte kullanılan TF-TDF öznitelikleri ile KTV öznitelikleri birbirine yakın sonuçlar vermiştir. Ancak kelime temsillerinin boyutlarını artırarak TF-TDF ile alınan doğruluk oranlarından daha yüksek sonuçlar elde edilebilmektedir; oysa TF-TDF yönteminde $20,000$ kelimeden sonra doğruluk oranlarının artmadığı görülmüştür. (Tablo-III). Bunun muhtemel sebebi olarak yüksek boyutlu TF-TDF özniteliklerinde DVM'nın etkili öğrenememesi olduğu düşünülebilir. KTV ve YSA yöntemlerinde Sabah ve Cumhuriyet derlemleri için sırasıyla yaklaşık $70,000$ ve $60,000$ kelimeden oluşan kelime hazneleri kullanılmıştır.

Kelime temsil vektörlerini ve doküman sınıflandırmayı aynı anda öğrenen Yapay Sinir Ağı yaklaşımı bütün vektör boyutları için KTV+DVM kombinasyonundan iyi sonuç vermiş ve en iyi sonuçlar bu yöntemle alınmıştır (Sabah ve Cumhuriyet derlemleri için sırasıyla %88.28 ve %74.31). Bu da kelime temsillerinin etiket bilgisi kullanılarak güncellenmesinin doğruluk oranlarını artırdığını göstermektedir. Ayrıca vektör boyutu küçüldükçe başarının arttığı görülmektedir, bu da doküman sınıfı ile ilgili bilgilerin çok düşük boyutlu kelime vektörlerinde ihtiva edilebileceğini göstermektedir.

TABLO IV: DOĞRULUK ORANLARI (%)
SDA için K değerleri sırasıyla Sabah ve Cumhuriyet derlemleri içindir.

Yöntem	Sabah	Cumhuriyet
SDA ($K = 4$ / $K = 14$)	65.41	47.94
SDA ($K = 10$ / $K = 20$)	67.60	43.31
SDA ($K = 20$ / $K = 30$)	72.08	45.37
TF-TDF ($10K$ K. Haznesi) + DVM	86.41	71.81
KTV ($d = 100$) + DVM	85.47	70.34
KTV ($d = 200$) + DVM	86.16	71.55
KTV ($d = 400$) + DVM	86.72	72.24
KTV ($d = 600$) + DVM	86.89	72.50
KTV ($d = 100$) + YSA	**88.28**	**74.31**
KTV ($d = 200$) + YSA	87.93	73.64
KTV ($d = 400$) + YSA	87.94	72.29
KTV ($d = 600$) + YSA	87.53	72.97

V. SONUÇ

Bu çalışmada, iki büyük ve yeni Türkçe metin derlemi konu kategorileri ile oluşturulmuş ve paylaşıma açılmıştır. Kelime vektörlerinin ortalamasını alarak çalışan bir yapay sinir ağının, diğer yöntemlere göre daha iyi sonuç verdiği gözlemlenmiştir. İleride, etiketsiz veri de kullanarak, gözetimsiz yöntemlerin avantajlarından faydalanabileceğimiz yarı-gözetimli yöntemler üzerinde çalışılacaktır.

KAYNAKÇA

[1] R. Jindal, R. Malhotra, and A. Jain, "Techniques for text classification: Literature review and current trends," *webology*, vol. 12, no. 2, p. 1, 2015.

[2] Y. Goldberg, "A primer on neural network models for natural language processing." *J. Artif. Intell. Res.(JAIR)*, vol. 57, pp. 345–420, 2016.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[4] E. Kilic, N. Ates, A. Karakaya, and D. O. Sahin, "Two new feature extraction methods for text classification: Tesdf and sadf," in *Signal Processing and Communications Applications Conference (SIU), 2015 23th*. IEEE, 2015, pp. 475–478.

[5] S. Ay, Y. S. Doğan, S. Alver, and Ç. Kaya, "A novel attribute weighting method with genetic algorithm for document classification," in *Signal Processing and Communication Application Conference (SIU), 2016 24th*. IEEE, 2016, pp. 1129–1132.

[6] G. Şahİn, "Turkish document classification based on word2vec and svm classifier," in *Signal Processing and Communications Applications Conference (SIU), 2017 25th*. IEEE, 2017, pp. 1–4.

[7] H. B. Sahin, C. Tirkaz, E. Yildiz, M. T. Eren, and O. Sonmez, "Automatically annotated turkish corpus for named entity recognition and text categorization using large-scale gazetteers," *arXiv preprint arXiv:1702.02363*, 2017.

[8] P. Tüfekci, E. Uzun, and B. Sevinç, "Text classification of web based news articles by using turkish grammatical features," in *Signal Processing and Communications Applications Conference (SIU), 2012 20th*. IEEE, 2012, pp. 1–4.

[9] D. Kılınç, A. Özçift, F. Bozyigit, P. Yıldırım, F. Yücalar, and E. Borandag, "Ttc-3600: A new benchmark dataset for turkish text categorization," *Journal of Information Science*, vol. 43, no. 2, pp. 174–185, 2017.

[10] S. S. Keerthi and D. DeCoste, "A modified finite newton method for fast solution of large scale linear svms," *Journal of Machine Learning Research*, vol. 6, no. Mar, pp. 341–361, 2005.

[11] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of machine learning research*, vol. 5, no. Jan, pp. 101–141, 2004.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[13] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *advances in neural information processing systems*, 2010, pp. 856–864.

[14] M. U. Sen and H. Erdogan, "Learning word representations for turkish," in *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*. IEEE, 2014, pp. 1742–1745.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[16] A. A. Akın and M. D. Akın, "Zemberek, an open source nlp framework for turkic languages," *Structure*, vol. 10, pp. 1–5, 2007.

[17] Z. Cataltepe, Y. Turan, and F. Kesgin, "Turkish document classification using shorter roots," in *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*. IEEE, 2007, pp. 1–4.

[18] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.