Word2Vec ve SVM Tabanlı Türkçe Doküman Sınıflandırma

Turkish Document Classification Based on Word2Vec and SVM Classifier

Gürkan ŞAHİN
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
gurkansahin08@gmail.com

***Özetçe*—Bu çalışmada farklı kategorilere ait Türkçe metinler word2vec ile elde edilen kelime vektörleri kullanılarak sınıflandırılmıştır. Öncelikle tüm metinlerin içindeki herbir kelimenin vektörü çıkartılmış, herbir metin içerdiği kelimelerin ortalama vektörleri cinsinden temsil edilmiştir. Sonrasında SVM sınıflandırıcı kullanılarak metinler sınıflandırılmış ve yedi farklı kategori için en yüksek 0.92 F ölçüm değeri elde edilmiştir. Sonuçta word2vec'in klasik tf-idf tabanlı sınıflandırmadan daha başarılı olduğu deneysel olarak gösterilmiştir.**

Anahtar Kelimeler — doküman sınıflandırma; SVM word2vec.

***Abstract*—In this study, Turkish texts belonging to different categories were classified by using word2vec word vectors. Firstly, vectors of the words in all the texts were extracted then, each text was represented in terms of the mean vectors of the words it contains. Texts were classified by SVM and 0.92 F measurement score was obtained for seven different categories. As a result, it was experimentally shown that word2vec is more successful than tf-idf based classification for Turkish document classification.**

Keywords — document categorization; SVM; word2vec.

I. GİRİŞ

Doküman sınıflandırma Doğal Dil İşleme (DDİ) alanının en popüler çalışma konularından bir tanesidir. Bu problemde amaç farklı kategorilere (sınıflara) ait etiketli metinlerin makine öğrenmesi algoritmaları kullanılarak sınıflandırılmasıdır.

Doküman sınıflandırmada en sık kullanılan yöntem metinlerde belirli bir frekans değerinin üstünde geçmiş olan *bag of words* (BoW) kelimelerin kullanılmasıdır. Burada herbir metin N adet kelime cinsinden ifade edilmektedir. Metin temsillerinde 1-gram, 2-gram, 3 gram vb. kelime n-gramları özellik olarak kullanılmaktadır. Herbir metin ilgili n-gramın terim frekansı (tf), ters-terim frekansı (idf), terim-ters terim frekansı (tf-idf) vb. ağırlıklandırma yöntemleri kullanılarak temsil edilmektedir. Sonrasında ise metinler ve kategori bilgileri SVM, kNN vb. makine öğrenmesi yöntemleri ile sınıflandırılmaktadır.

BoW yöntemi temelde çok basit ve etkili bir yöntem olsa da en önemli dezavantajı metinlerin yeterince iyi temsil edilememesi yani temsil vektörlerinin çok sayıda sıfırlardan oluşmasıdır (sparsity). Ayrıca BoW yönteminde kelimeler arası anlamsal ilişkiler (semantic similarity) göz ardı edilmektedir. Bütün bu problemler de sınıflandırma başarısında düşüşe neden olmaktadır.

Bu çalışmada dokümanların temsilinde BoW' un yanında word2vec kelime vektörleri de kullanılmış ve sınıflandırma başarısına olan etkileri incelenmiştir. Yapılan denemeler sonucunda word2vec kelime vektörlerinin eşik başarısı (baseline) kabul edilen BoW'a göre daha iyi sonuç ürettiği görülmüştür.

Makalenin ikinci bölümünde geçmiş çalışmalardan bahsedilmiş, üçüncü bölümde önerilen yöntem açıklanmıştır. Dördüncü bölümde deneysel sonuçlar verilmiş, son bölümde ise sonuçlar hakkında değerlendirmeler yapılmıştır.

II. MEVCUT ÇALIŞMALAR

[1] çalışmasında sosyal medya metinlerinden oluşan dokümanlar üç farklı yöntem ile sınıflandırılmıştır. Metinler tf-idf ağırlıklı BoW kelimeler (baseline), word2vec kelime vektörleri (naive doc2vec) ve paragraf vektör (doc2vec) ile temsil edilmiş, SVM sınıflandırıcı ile sınıflandırılmıştır. Ayrıca çalışmada kelimelerin köklerine ayrıştırılıp (stemming) kullanılmasının sınıflandırma başarısına olan etkisi de incelenmiştir. Denemeler sonucunda BoW tf-idf için %77.3, naive doc2vec için %77.8, doc2vec için ise %77.8 F_1 ölçüm değerleri elde

978-1-5090-6494-6/17/$31.00 ©2017 IEEE

edilmiştir. Ayrıca *stemming* işlemi ile F_1 değerinin %73.4'den %77.2'ye çıktığı görülmüştür.

[2] çalışmasında hastanelerden alınan klinik verileri kullanılarak *damp-heat syndrome* hastalığı hasta veya değil olarak (2 sınıf) tespit edilmiştir. Problemin çözümünde word2vec ve tf-idf birleşimi kullanılmıştır. Sonuçların değerlendirilmesinde dört farklı sınıflandırma algoritması (kNN, SVM, decision tree, random forest) kullanılmış ve en iyi sonuç %82.1 doğruluk ile kNN'den elde edilmiştir.

[3] çalışmasında otel ve restoran müşteri değerlendirme puanları (1-5 aralığında) olumlu veya olumsuz olarak iki sınıfa indirgenmiştir. İki farklı veri kümesi farklı ağırlıklandırmalar ile temsil edilerek sınıflandırılmıştır. Birinci veri kümesinden BoW (baseline) için %93, word2vec ve idf için %93, word2vec ve tf-idf için %95 *AUC* değeri elde edilirken; ikinci veri kümesi için sırasıyla %87, %86, %90 *AUC* değerleri elde edilmiştir.

III. YÖNTEM

Bu bölümde metinlerin temsilinde kullanılan yöntemler (BoW, word2vec) ve kullanılan veri kümesi bilgileri verilmiştir.

A. Kullanılan veri kümesi

Çalışmada 7 farklı sınıftan oluşan 22.729 Türkçe doküman [4] kullanılmıştır.

Tablo 1. Veri kümesindeki örneklerin sınıf dağılımları

Sınıf	Örnek sayısı	Sınıf	Örnek sayısı
Ekonomi	3265	Siyaset	1849
Kültür-sanat	1155	Spor	11514
Magazin	2792	Teknoloji	771
Sağlık	1383		

Öncelikle tüm metinlerin içindeki noktalama işaretleri atılmış ve kelimelerin hepsi küçük harfle temsil edilmiştir.

Bilindiği gibi Türkçe sondan eklemeli bir dildir. Bu nedenle aynı kelime farklı ekler alarak farklı yapılarda karşımıza çıkabilmektedir. Örneğin ağaç ve ağaçların kelimelerini ele alırsak her iki kelimenin de kökü ortak ve ağaç'tır. Genelde yapılan çalışmalarda *stemming* işleminin sınıflandırma başarısını arttırdığı gözlenmektedir. Bu çalışmada hem ekli kelimelerin hem de kelime köklerinin kullanılmasının word2vec vektör temsillerindeki başarısı incelenmiştir. Kelime köklerinin elde edilmesinde Zemberek [5] DDİ kütüphanesi kullanılmıştır. Zemberek verilen Türkçe kelimeyi kök, kök türü (sıfat, fiil, isim vb.) ve eklerine ayrıştırmaktadır. Zemberek verilen bir kelime için birden fazla olası çözüm üretebilmektedir ancak bu çalışmada ilk üretilen sonuçlar kullanılmıştır. Örneğin "arabacıların" kelimesi Zemberek ile {araba+ISIM+ISIM_ILGI_CI+ISIM_COGUL_LER+ISIM_TAMLAMA_IN} şeklinde çözümlenmektedir.

B. Metinlerin BoW ile temsili

Klasik yöntemde herbir metin, içerdiği kelimeler (n-gram) türünden temsil edilir. Köklerine ayrışmış metinlerden oluşan derlemden Text-NSP [6] kütüphanesi kullanılarak tüm 1-gram kelimeler çıkartılmıştır. Örneğin "O güzel insanlar beyaz atlara binip gittiler" metnindeki 1-gramlar {o, güzel, insan, beyaz, at, bin, git} şeklindedir.

N-gramlar çıkartıldıktan sonra veriseti içindeki herbir metin bunlar ile temsil edilmektedir. Burada tüm n-gramların özellik olarak kullanılması yerine daha fazla ayırt edici olanlar seçilmiştir. 1-gram kelimeler tf-idf puanlarına göre sıralanmış ve ilk N tanesi seçilerek özellik olarak kullanılmıştır. Böylece farklı N değerlerinin sınıflandırma başarısına olan etkileri incelenmiştir.

Terim frekansı ağırlıklandırma: İlgili kelimenin (n-gram) metin içindeki geçme frekansı kullanılır. (1)'de t ilgili n-gram terimini, d ilgili metni, $f_{t,d}$ n-gramın metin içindeki terim frekansını, N ise doküman içindeki toplam kelime sayısını ifade etmektedir. Terim frekansını kullanmak yerine ikili değerler *(binary)* de kullanılabilmektedir. Bu yöntemde ilgili n-gram metin içinde varsa 1, yoksa 0 ile temsil edilmektedir.

$$tf_{(t,d)} = \frac{f_{t,d}}{N} \tag{1}$$

Ters doküman frekansı ağırlıklandırma: Herhangi bir kelimenin dokümanlar için ne kadar önemli olduğunu gösteren bir ölçüttür. (2)'de N toplam doküman sayısını, n_t ise t teriminin (n-gram) kaç farklı dokümanda geçtiğini temsil etmektedir. Bu ağırlıklandırma ile az sayıda dokümanda geçen terimler, çok sayıda dokümanda geçen terimlere göre daha fazla ağırlıklandırılmıştır.

$$idf_{(t,d)} = log\left(\frac{N}{1+n_t}\right) \tag{2}$$

Terim – ters doküman frekansı: Terim frekansı ile ters doküman frekansının çarpımından oluşan ağırlıklandırmadır. Eğer bir terim bir dokümanda yüksek frekansta, aynı zamanda da az sayıda dokümanda geçmiş ise tf-idf değeri yüksek çıkmaktadır.

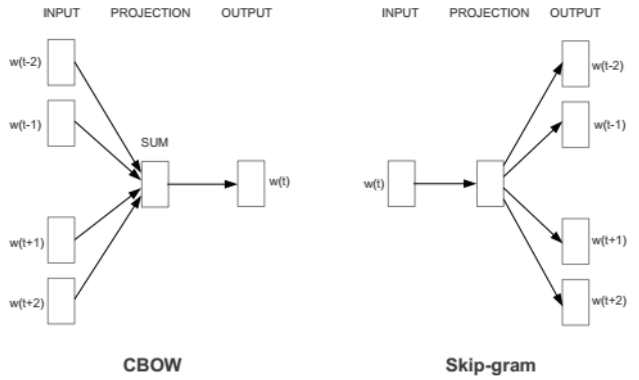
$$tf\_idf_{(t,d)} = tf_{(t,d)} \,.\, idf_{(t,d)} \tag{3}$$

C. Metinlerin Word2Vec kelime vektörleri ile temsili

Word2vec [7] Mikolov ve arkadaşları tarafından geliştirilmiş yapay sinir ağı yapısını kullanan eğiticisiz (unsupervised) bir DDİ aracıdır. Araç girdi olarak bir metin almakta ve metnin içindeki herbir kelimeyi vektörel olarak temsil etmektedir. Temelde word2vec anlamsal olarak birbirine benzer kelimeleri birbirine yakın koordinatlarda kümelemektedir.

Kelime koordinatlarının bulunmasında *continuous bag of words* (CBoW) ve skip-gram (SG) olmak üzere iki farklı öğrenme mimarisi kullanılmaktadır. CBoW mimarisinde

bir kelimenin belli bir pencere boyutu içindeki komşu kelimelerine (sağındaki ve solundaki kelimeler) bakılmakta ve ilgili kelime komşu kelimelerden tahmin edilmeye çalışılmaktadır. Skip-gram mimarisinde ise tam tersi şekilde hedef kelimeye bakılarak komşu kelimeler tahmin edilmektedir.



Şekil 1. Cbow ve skip-gram mimarileri [7]

Genelde SG mimarisi derlem frekansı az olan (infrequent words) kelimeler için daha iyi kelime vektörleri üretirken, CBoW daha büyük çaptaki derlemlerde daha iyi sonuçlar üretmektedir.

CBoW ve SG mimarilerinin herbiri öğrenme için *hierarchical softmax (HS)* veya *Negative sampling (NS)* eğitim algoritmalarından birini kullanmaktadır. HS eğitim algoritması genelde frekansı düşük olan kelimelerde iyi sonuçlar üretirken, NS yüksek frekanslı kelimelerde daha iyi sonuçlar üretmektedir.

Tablo 2. Word2vec parametreleri

Parametre	Görevi	Değerler
-size	Vektör uzunluğu	Varsayılan: 100
-window	Bakılacak komşu kelime sayısı, pencere boyutu	Genellikle: 5 – 10
-hs	Eğitimde kullanılacak öğrenme algoritması	hs : 1 (HS kullanılır) hs : 0 (NS kullanılır)
-negative	Kullanılacak negatif örnek sayısı	Varsayılan : 5
-min_count	# frekansından düşük frekanslı kelimeleri sil	Varsayılan : 5
-alpha	Öğrenme oranı	Varsayılan : 0.025
-cbow	CBoW veya SG mimarisini kullan	cbow : 0 (SG kullanılır) cbow : 1 (CBoW kullanılır)

Word2vec sayesinde verilen bir kelimeye en benzer kelimeler elde edilebilmektedir. Benzer kelimelerin bulunmasında kelime vektörleri arasındaki kosinüs benzerliği kullanılmaktadır.

Tablo 3. Bazı hedef kelimeler için elde edilen en benzer kelimeler ve benzerlik değerleri (Derlem olarak BOUN web derlemi [8] kullanılmış, CBoW mimarisi HS ile eğitilmiştir)

güzel	elma	fil	bilgisayar
etkileyici 0.61	şeftali 0.61	deve 0.52	yazılım 0.64
güzle 0.59	limon 0.58	sincap 0.52	masaüst 0.59
harika 0.58	erik 0.57	çita 0.52	java 0.57
keyifli 0.58	muz 0.57	ejderha 0.52	internet 0.56
şahane 0.57	kayısı 0.56	maymun 0.51	kablosuz 0.54
iyi 0.54	kivi 0.56	kedi 0.49	monitör 0.53
mükemmel 0.52	üzüm 0.56	kurbağa 0.49	modül 0.53
beğenilen 0.51	vişne 0.55	köpek 0.49	modem 0.53
fiyakalı 0.51	armut 0.55	tilki 0.48	linux 0.53
anlamlı 0.49	mandalina 0.54	antilop 0.47	unix 0.53

Word2vec'in sağladığı en büyük kolaylıklardan birisi de kelime vektörleri arasında aritmetik işlemlerin yapılmasına olanak sağlamasıdır. Wor2vec kelime vektörleri arasında vec(Russia) – vec(Moscow) = vec(Turkey) – vec(Ankara) gibi aritmetik işlemler yaparak benzer anlamsal ilişkiye ait farklı ikililer çıkartılabilmektedir.

Öncelikle hem kök halindeki derlem hem de ekli kelimelerden oluşan derlem word2vec'e verilmiş ve herbir kelime vektörü çıkartılmıştır. Kelimeler ve vektörleri sonrasında yapılacak hızlı sorgu işlemleri için Apache Lucene ile indekslenmiştir.

Kelime vektörleri çıkartılırken hem CBoW hem de SG mimarileri hem HS hem de NS öğrenme algoritmaları ile birlikte kullanılmıştır. Kelime pencere boyut (-window) olarak 5 ve 10 arası değerler seçilmiştir. Kelime vektör uzunluğu (-size) olarak N=400 değeri seçilmiştir. Diğer tüm parametreler varsayılan değerleri ile kullanılmıştır.

Stop word kelimeler tüm metinlerde geçtikleri için ayırt edici özellikleri içermemekte ve sınıflandırma başarısını düşürmektedir. Bu nedenle herbir metin içindeki *stop word* (gibi, sanki, kez vb.) olmayan kelimelerin vektörleri toplanmış ve kelime sayısına bölünerek ilgili metni temsil eden ortalama vektör elde edilmiştir. Böylece herbir sınıfa ait metinler N boyutlu vektör ile temsil edilmiştir.

D. Sınıflandırma

Metinlerin sınıflandırılmasında Weka içindeki SVM sınıflandırma algoritmasından yararlanılmıştır.

IV. DENEYSEL SONUÇLAR

Verisetindeki örneklerin 15.656 tanesi eğitim, 7.073 tanesi ise test verisi olarak kullanılmıştır. Kelime vektörleri oluşturulurken CBoW mimarisi HS algoritması ile (CBoW_HS), NS algoritması ile (CBoW_NS) ayrıca SG mimarisi HS (SG_HS) ve NS algoritması (SG_NS) ile eğitilmiştir.

Tablo 4. Kelime *kök* vektörlerinden elde edilen başarılar, (-window=5~10 için)

Yöntem	Vektör boyutu (özellik sayısı = 400)					
	5	6	7	8	9	10
CBoW_HS	0.85	0.90	0.87	0.87	0.89	0.88
CBoW_NS	0.85	0.91	0.91	0.90	0.91	0.88
SG_HS	0.84	0.89	0.87	0.91	0.87	0.90
SG_NS	0.90	0.91	0.91	0.91	0.91	0.89

Tablo 4 incelendiğinde en yüksek sınıflandırma başarısı %91 olarak elde edilmiştir. Genelde SG_NS yönteminden

farklı pencere boyutları için sonuç fazla değişmezken, diğer yöntemlerde farklı pencere boyutları için başarı değerleri de farklılık göstermiştir.

Kullanılan derlemin büyüklüğünün word2vec kelime vektörlerinin kalitesine olan etkilerini incelemek için kelime vektörlerinin oluşturulmasında yaklaşık 12 milyon cümle ve 500 milyon kelimeden oluşan [8] derlemi kullanılmıştır. Buradan elde edilen vektörlerin kullanılmasıyla SG_NS yöntemi için –size:400 ve –window:7 parametreleri için 0.90 F ölçüm değeri elde edilmiştir. Buradan hareketle beklenenin aksine çok büyük derlem kullanılması sonucu sınıflandırma başarısında yaklaşık %1'lik bir azalma elde edildiği görülmüştür.

Tablo 5. *Ekli* kelime vektörlerinden elde edilen başarılar, (-window=5~10 için)

Yöntem	Vektör boyutu (özellik sayısı = 400)					
	5	6	7	8	9	10
CBoW_HS	0.91	0.89	0.90	0.89	0.89	0.84
CBoW_NS	0.86	0.87	0.88	0.87	0.90	0.89
SG_HS	0.92	0.90	0.87	0.87	0.89	0.90
SG_NS	0.89	0.89	0.89	0.88	0.90	0.90

Ekli kelimelerin kullanılmasıyla elde edilen kelime vektörlerinin kullanılması sonucu en yüksek sınıflandırma başarısı 0.92 F ölçüm değeri ile SG_HS yönteminden elde edilmesine karşın genelde kök kelimelere göre biraz daha düşük başarılar elde edilmiştir.

Metin temsillerinde metin içindeki tüm kelimeler yerine farklı etiketteki kelimelerin kullanılmasının sınıflandırmaya etkileri incelenmiştir. Bunun için metinlerin içindeki sadece *isim*, *fiil* ve *sıfat* kelime vektörlerinin ortalamaları alınmıştır. Sonuçlar incelendiğinde en ayırt edici kelime vektörlerinin *isim* etiketli kelimeler, en az ayırt edici kelime vektörlerinin ise *sıfat* türündeki kelimelerden elde edildiği görülmüştür. Ancak tekil kullanımlardan elde edilen başarının her üç etiketin de kullanılmasında elde edilen başarıdan düşük olduğu görülmüştür.

Tablo 6. *Kök, ekli <isim,sıfat,fiil>* kelime vektörlerinden elde edilen başarılar, (-window=7, -size=400 için)

	kök			***ekli***		
Yöntem	***isim***	***fiil***	***sıfat***	***isim***	***fiil***	***sıfat***
CBoW_HS	0.86	0.55	0.51	0.82	0.68	0.51
CBoW_NS	0.79	0.74	0.48	0.87	0.62	0.46
SG_HS	0.88	0.32	0.50	0.83	0.72	0.53
SG_NS	0.87	0.68	0.51	0.90	0.61	0.48

Önerilen yöntem klasik BoW tf-idf sınıflandırma başarısı ile karşılaştırılmıştır. BoW için farklı kelime vektör uzunlukları {50, .., 2.10^3} kullanılmıştır.

Tablo 7. *BoW* kelime köklerinden elde edilen başarılar

Yöntem	Vektör boyutu (özellik sayısı)						
	50	200	300	400	500	10^3	2.10^3
BoW_tf_idf	0.58	0.80	0.83	0.85	0.87	0.89	0.89

Tablo 8. SG_NS, –size:400, -window:7, kelime *kök* vektörler için elde edilen karmaşıklık matrisi (a:ekonomi, b:kültür-sanat, c:magazin, d:sağlık, e:siyaset, f:spor, g:teknoloji)

		Gerçek						
		a	**b**	**c**	**d**	**e**	**f**	**g**
Tahmin edilen	**a**	936	5	13	7	83	13	32
	b	8	143	209	3	16	0	6
	c	8	5	883	7	7	17	4
	d	23	0	5	404	14	1	14
	e	24	5	10	3	563	2	10
	f	25	2	51	4	16	3234	1
	g	20	12	10	4	1	3	207

V. SONUÇLAR

Bu çalışmada yedi farklı kategorideki Türkçe metinlerin sınıflandırılmasında word2vec kelime vektörlerinin kullanımı klasik BoW metin temsiliyle karşılaştırılmıştır. Herbir metin içerdiği kelimelerin vektör ortalamaları cinsinden ifade edilmiş ve SVM ile sınıflandırılmıştır. Kelime vektörleri oluşturulurken word2vec'in farklı parametre değerleri için denemeler yapılmış, sınıflandırma başarısına etkileri incelenmiştir. Tf-idf ağırlıklandırmalı BoW yöntemi ile 0.89 F ölçüm değeri elde edilirken, word2vec kelime vektörlerinin kullanılmasıyla en yüksek 0.92 F ölçüm değeri elde edilmiştir.

Türkçe metin sınıflandırma için word2vec'in kullanıldığı ilk çalışma olması ve kelime vektörlerinin sınıflandırmada başarılı olarak kullanılabileceğinin gösterilmesi çalışmanın en önemli katkılarındandır. Gelecek çalışmalarda kelime vektörlerinin yanında paragraf vektörlerinin (Doc2Vec) kullanımının etkilerinin incelenmesi hedeflenmektedir.

KAYNAKLAR

[1] Venekoski, Viljami, Samir Puuska, and Jouko Vankka. "Vector Space Representations of Documents in Classifying Finnish Social Media Texts." *International Conference on Information and Software Technologies*. Springer International Publishing, 2016.

[2] Zhu, Wei, et al. "A study of damp-heat syndrome classification using Word2vec and TF-IDF." *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*. IEEE, 2016.

[3] Jiang, Suqi, et al. "Integrating rich document representations for text classification." *Systems and Information Engineering Design Symposium (SIEDS), 2016 IEEE*. IEEE, 2016.

[4] *http://www.kemik.yildiz.edu.tr/?id=28*

[5] Akın, Mehmet Dündar, and Ahmet Afşin Akın. "Türk dilleri için açık kaynaklı doğal dil işleme kütüphanesi: ZEMBEREK." *Elektrik Mühendisliği* 431 (2007): 38.

[6] Pedersen, Ted, et al. "The Ngram statistics package (text:: nsp): A flexible tool for identifying ngrams, collocations, and word associations." *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*. Association for Computational Linguistics, 2011.

[7] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.

[8] Sak, Haşim, Tunga Güngör, and Murat Saraçlar. "Turkish language resources: Morphological parser, morphological disambiguator and web corpus." *Advances in natural language processing*. Springer Berlin Heidelberg, 2008. 417-427.