

Word2Vec, Doc2Vec ve FastText Algoritmaları ile Yarı Öğreticili Türkçe Metin Sınıflandırma

Semi-supervised Turkish Text Categorization with Word2Vec, Doc2Vec and FastText Algorithms

Hakkı Yağız Erdiñ
Bilgisayar Mühendisliğı Bölümü
Donanım Haber, Doğış Üniversitesi
İstanbul, Türkiye
yagizerdinc@donanimhaber.com

Aysun Güran
Bilgisayar Mühendisliğı Bölümü
Doğış Üniversitesi
İstanbul, Türkiye
adogrusoz@dogus.edu.tr

Özetçe—Bu çalışmada yarı öğreticili bir teknik ile Word2Vec, Doc2Vec ve FastText algoritmalarının metin sınıflandırma problemi üzerindeki başarımlarını kıyaslanmıştır. Bu algoritmaların ön işleme aşamalarından etkilenme düzeyleri hem etiketli hem de etiketsiz dokümanlardan oluşan yaklaşık 5 milyon Türkçe haber dokümanını içeren bir derlem üzerinde analiz edilmiştir. Haber derlemi sınıflandırılırken Naive Bayes, Destek Vektör Makineleri, Yapay Sinir Ağları, Karar Ağaçları ve Logistik Regresyon sınıflandırma algoritmaları kullanılmış ve elde edilen sonuçlar paylaşılmıştır.

Anahtar Kelimeler — *word2vec; doc2vec; fasttext; metin sınıflandırma*

Abstract—In this study, the performance values of Word2Vec, Doc2Vec and FastText algorithms are compared on the text categorization problem based on a semi-supervised learning technique. The impact of some preprocessing techniques are also analyzed on a corpus that contains approximately 5 million Turkish news documents which are both in labeled and unlabeled manners. Naive Bayes, Support Vector Machines, Artificial Neural Networks, Decision Trees and Logistic Regression classification algorithms are used at the classification phase and the obtained results are shared.

Keywords — *word2vec; doc2vec; fasttext; text categorization*

I. GİRİŞ

Son zamanlardaki teknolojik gelişmeler nedeniyle yapay sinir ağları (YSA) tekrar popüler hale gelmiş ve YSA'ların metin madenciliğinde özellikle kelimelerin anlamsal benzerliklerinin tespit edilme aşamasında kullanılması, geleneksel kelime çantası modellerinin üretebildiği çözümlerden daha yüksek başarımların elde edilmesini sağlamıştır. Kelimeler arasındaki anlamsal ilişkinin modellenebilmesi için kelimeler öncelikle vektörler ile sayısallaştırılmalıdır. Mikalov ve diğerleri 2013 yılında YSA'lara dayalı olan Word2Vec isimli bir model önermişlerdir [1]. Bu modelde kelimeler belirli pencere boyutundaki komşularıyla birlikte değerlendirilmekte ve benzer anlamlı kelimeler birbirlerine yakın olacak şekilde vektörler ile temsil edilmektedir. Tahmin temelli bir yöntem olan Word2Vec kelimeleri sayısallaştırırken Sürekli Kelime Çantası Modeli

(Continuous Bag of Words- CBOW) ve Gram Atla Modeli (Skip-gram) isimli iki farklı öğrenme mimarisi kullanmaktadır. CBOW mimarisinde bir kelimenin belli bir pencere boyutu içindeki komşu kelimelerine (sağındaki ve solundaki kelimelere) bakılmakta ve hedef kelime komşu kelimelerden tahmin edilmeye çalışılmaktadır. Skip-gram mimarisinde ise tam tersi şekilde pencere merkezine oturtulan hedef kelimeye bakılarak komşu kelimeler tahmin edilmektedir. Hesaplama maliyetini indirmek için CBOW ve Skip-gram mimarilerinin her biri öğrenme için Hiyerarşik Softmax (Hierarchical Softmax-HS) veya Olumsuz Örneklem (Negative Sampling-NS) eğitim algoritmalarından birini kullanmaktadır. HS eğitim algoritması genelde düşük frekanslı kelimeleri içeren derlemlerde iyi sonuçlar üretirken, NS eğitim algoritması yüksek frekanslı kelimeleri içeren derlemlerde daha iyi sonuçlar üretmektedir.

Yine Mikalov ve diğerleri daha sonraki çalışmalarında Word2Vec modeliyle benzer mantıkla sadece kelimelerin değil, metinlerin de sayısallaştırılabildiğini göstermişlerdir [2]. Doc2Vec adını verdikleri çalışmaları Word2Vec temelli olup, bu modelde kelime vektörlerinin yanına her bir doküman için bir doküman vektörü eklenmiştir. Böylece dokümanlar da sabit boyuttaki vektörler ile ifade edilmiştir. Bu modelde hem doküman, hem de kelime vektörleri elde edilebilmektedir. Doc2Vec modeli öğrenimindeki yaklaşımlar, Doküman Vektörlerinin Dağıtılmış Bellek Modeli (Distributed Memory Model of Paragraph Vectors- PV-DM) ve Doküman Vektörünün Dağıtılmış Kelime Çantası Modeli (Distributed Bag of Words-Paragraph Vector- PV-DBOW) olarak belirlenmiştir. Bu yaklaşımlar sırasıyla Word2Vec'in CBOW ve Skip-gram yaklaşımına benzemektedirler.

2016'da Facebook tarafından geliştirilmiş olan FastText modeli de Word2Vec modeline dayanmaktadır [3]. FastText kelimeleri karakter n-gramlar halinde parçaladıktan sonra yapay sinir ağına sunmaktadır. Nihayetinde herhangi bir kelimenin vektörü, o kelimenin n-gram vektörlerinin toplamı ile bulunmaktadır. FastText modelinde karakter n-gramların kullanımı sayesinde az sayıda bulunan kelimelerin Word2Vec'e göre daha iyi bir şekilde ifade edildiği belirtilmiştir [3].

Bu çalışmada YSA'lara dayalı olan Word2Vec, Doc2Vec ve FastText modellerinin başarımlarının değerlendirildiği bir metin sınıflandırma problemi ele alınmıştır. Bu bağlamda yarı eğitimci bir mantık ile hem etiketli hem de etiketsiz dokümanlardan oluşan yaklaşık 5 milyon haber dokümanından oluşan bir derlem kullanılmıştır. Bilindiği gibi ön işleme aşamaları sınıflandırma problemleri üzerinde genellikle olumlu bir etkiye sahiptir. Çalışmamızda noktalama işaretlerinin atılması, durak kelimelerinin dikkate alınmaması, kelimelerin gövdelerine ayrılması ve temel olarak bu üç ön işlem aşamasının farklı kombinasyonlarından oluşan ön işlem aşamalarının YSA'lara dayalı olan Word2Vec, Doc2Vec ve FastText algoritmaları üzerindeki etkileri incelenmiştir. Oluşturulan derlemdeki dokümanların Word2Vec, Doc2Vec ve FastText algoritmaları ile elde edilen vektör gösterimleri kullanılarak dokümanlar, Naive Bayes (NB), Destek Vektör Makineleri (DVM), Yapay Sinir Ağları (YSA), Karar Ağaçları (KA) ve Logistik Regresyon (LR) sınıflandırma algoritmaları ile sınıflandırılmıştır. Bildiğimiz kadarıyla çalışmamız Word2Vec, Doc2Vec ve FastText algoritmalarının başarımlarını Türkçe metinler üzerinde kıyaslayan ve ön işlem aşamalarının etkilerini de ayrıntı ile ele alan ilk çalışmadır. Şu ana kadar bu algoritmaları kullanarak Türkçe metinler üzerinde uygulanan çalışmalar şu şekilde özetlenebilir: Arabacı ve diğerleri [4] Word2Vec modeli ile birlikte Fisher kodlamasını kullanarak anlamsal olarak benzer cümleleri tespit etmişlerdir. Karakuş ve diğerleri [5] Word2Vec öğrenme mimarisini skip-gram metodu ile kullanarak film yorumlarını kategorize etmiştir. Seyfioğlu ve Demirezen [6] Word2Vec ve Doc2Vec öğrenme mimarilerini kullanarak bir hava yolu şirketi hakkında yapılan yorumları sınıflandırmışlardır. Özer ve diğerleri [7] çalışmasında sosyal medya yorumları yazılırken kullanılan ifadelerin düzeltilmesi konusunda Word2Vec metodunu kullanmışlardır. Karasoy ve Ballı [8] Word2Vec metodu ile kısa mesajları sınıflandırmışlardır. Güngör ve Yıldız [9] word2vec metodu ile Türkçe sözcüklerin vektör temsillerini elde etmiştir. Bu temsillerin Türkçe'deki söz dizimsel ve biçim bilimsel özellikleri, temsil etme yeteneği farklı köklere, çeşitli çekim ve yapım ekleri eklenip, üretilen deney kümeleri kullanılarak benzeşim yöntemi ölçülmüş ve çıkan sonuçları paylaşmışlardır. Şen ve Yanikoğlu [10] metin sınıflama probleminde yapay sinir ağlarına dayalı modellerinin geleneksel yöntemlere göre daha iyi sonuç verdiğini belirtmişlerdir.

Çalışmanın geri kalanı şu şekildedir: ikinci bölümde Word2Vec, Doc2Vec ve FastText algoritmaları anlatılmıştır; üçüncü bölümde kullanılan veri setleri tanıtılmış ve sistem özelliklerinden söz edilmiştir. Sonuçlar bölümünde ise genel yorumlarda bulunularak gelecek çalışmalardan bahsedilmiştir.

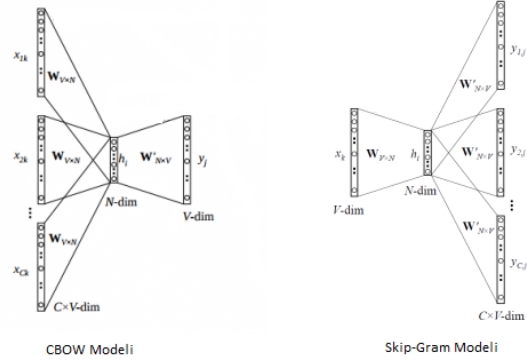
II. WORD2VEC, DOC2VEC VE FASTTEXT ALGORİTMALARI

Bu bölümde Word2Vec, Doc2Vec ve FastText modelleri ele alınacaktır.

A. Word2Vec Modeli

Word2Vec modeli giriş katmanı, gizli katman ve çıkış katmanından oluşan sığ bir YSA'dır. Word2Vec, CBOW ve Skip-Gram isimli iki öğrenme mimarisine sahiptir. Her iki öğrenme mimarisinde de giriş katmanı ile gizli katman arasında aktivasyon fonksiyonu kullanılmamaktadır. Gizli

katman ile çıkış katmanı arasında softmax aktivasyon fonksiyonu uygulanmaktadır. İki modelde de derlemde elde edilen sözlükteki her kelime için giriş vektörü (bir-sıcak vektör) ve çıkış vektörü (gömülü kelime vektörü) olmak üzere iki vektör gösterimi mevcuttur. Şekil 1'de görüldüğü gibi CBOW mimarisinde bir kelimenin belli bir pencere boyutu içindeki komşu kelimelerine (sağındaki ve solundaki kelimelere) bakılmakta ve hedef kelime komşu kelimelerden tahmin edilmeye çalışılmaktadır. Skip-gram mimarisinde ise tam tersi şekilde pencere merkezine oturtulan hedef kelimeye bakılarak komşu kelimeler tahmin edilmektedir.

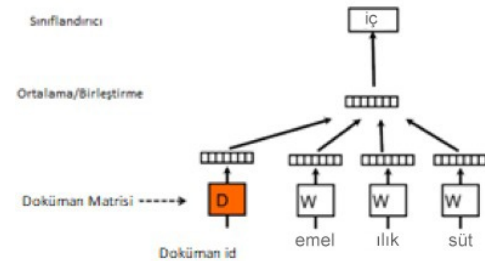


Şekil 1. CBOW ve Skip-gram Modelleri

CBOW modelinde, öncelikle incelenen eğitim veri setinden elde edilen sözlükteki her bir kelime için bir sıcak vektörler oluşturulur ve sözlük boyutunun $|V|$ olduğu varsayımı altında ağa sunulan bir-sıcak vektörler ile ağ eğitilmeye çalışılır. Gizli katmanın çıkış değerleri hesaplanırken, ağa sunulan giriş vektörlerinin ortalaması alınır. Ağın eğitiminin tamamlanmasının ardından elde edilen gizli katman ile çıkış katman arasındaki W ağırlık matrisi kelimelerin gömülü vektörlerini içeren matristir. Bu matristeki n . sütun sözlükteki n . kelimenin N boyutlu uzaydaki vektörüdür. Skip-gramda ise ağın eğitiminin tamamlanmasının ardından giriş katmanı ve gizli katman arasında oluşan ağırlık matrisi kullanılmaktadır.

B. Doc2Vec Modeli

Doc2Vec modelinde dokümanlar uzunluklarından bağımsız bir şekilde vektörleştirilebilmektedirler. Bu model Word2Vec modeline her doküman için bir doküman vektörünün (Doküman-id vektörü) eklenmesi ile oluşturulmuştur. Durum Şekil 2 ile görselleştirilmiştir:

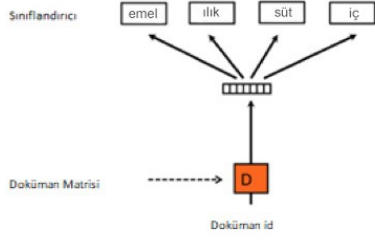


Şekil 2. PV-DM Modeli

Bu şekil incelendiğinde durumun CBOW modeline benzediği görülebilir. W matrisi eğitilirken aynı zamanda D doküman matrisi de elde edilmiştir. W matrisi ile kelime

vektörleri elde edilirken, D matrisi doküman vektörlerini vermektedir. Bu model PV-DM olarak isimlendirilmiştir.

Skip-gram modeline karşılık gelen yine paragraf vektörlerini içeren model ise Şekil 3. ile belirtilmiştir. Bu durum PV-DBOW modeli olarak adlandırılmıştır.



Şekil. 3. PV-DBOW Modeli

C. FastText Modeli

Çalışma [3] ile önerilmiş olan FastText algoritmasında kelimeler n-gramlarına ayrılmıştır. Diğer yaklaşımlarda kelimelerin birlikte kullanılması istatistiğine göre bir vektör temsili yaratılırken, kelimelerin iç düzeni hakkında bir bilgi elde edilememektedir. FastText modelinde karakter seviyesindeki n-gram bilgisi ile kelimelerin iç düzenindeki saklı bilgi de kelime temsili kullanabilmektedir. Böylece test derlemi içerisinde bulunan fakat eğitim derlemi içinde hiç yer almayan bir kelimenin sayısallaştırılması mümkün kılınmıştır.

III. VERİ SETİ VE SİSTEM ÖZELLİKLERİ

Bilindiği gibi ön işleme aşamaları, metin madenciliğinin ilk adımı olup çoğu zaman sınıflandırma algoritmalarının başarımlarını artırmasını sağlamaktadır. Çalışmamız kapsamında geniş kapsamlı iki derlem (etiketsiz (~4.76 milyon doküman) ve etiketli (~1.645 milyon eğitim ve ~323bin test dokümanı) oluşturulmuş olup bu derlemler üzerinde eş zamanlı olarak aşağıdaki ön işleme adımları uygulanmıştır:

- Noktalama İşaretlerinin Atılması (NA)
- Durak kelimelerinin atılması (DA)
- Noktalama işaretleri ve durak kelimelerini atılması (NDA)
- Kelimelerin Gövdelerine İndirgenmesi (Gİ)
- Kelime Gövdelerine indirgenmiş derlemden noktalama işaretleri ve durak kelimelerinin atılması (GNDA)

Durak kelime listesi için [11] referansı ile belirtilen liste kullanılmıştır. Cümle içerisindeki anlam bütünlüğünün bozulmaması adına kelimeler köklerine değil Zemberek yazılımı kullanılarak gövdelerine indirgenmiştir.

Çalışmamız kapsamında kullandığımız etiketsiz derlem çeşitli yayın kuruluşlarının çevrimiçi sayfalarının taranmasıyla elde edilen yaklaşık 4.76 milyon haber dokümanını içermektedir. Bu derlemin kullanım amacı kelime vektörlerinin elde edilmesidir. Belirtilen beş ön işlem aşamasından geçirildikten sonra beş farklı versiyona bürünen etiketsiz derlem üzerinde, Word2Vec, Doc2Vec ve FastText algoritmaları çalıştırılmış olup, her bir algoritmanın çıktısı olan kelime vektör modeli ayrı ayrı kaydedilmiştir. Tüm bu işlemler

esnasında vektör uzunluğu 100 olarak tanımlanmış, pencere boyutu 5 olarak belirlenmiş ve en az 5 kez geçen kelimeler dikkate alınmıştır. Ayrıca iterasyon sayısı 5 olarak kabul edilmiş ve öğrenme mimarisi olarak NS kullanılmıştır. Word2Vec ve FastText'te Skip-gram Modeli, Doc2Vec'de Skip-gram'a karşılık gelen PV-DBOW modeli ele alınmıştır. Eğitilen modeller [11] referansı ile belirtilen adresten indirilebilir halde sunulmuştur.

Tüm bu parametreler dikkate alındığında etiketsiz derlemden elde edilen toplam kelime sayısı ve bellek ihtiyaçları Tablo I ile belirtilmiştir.

TABLO I. ÖN İŞLEMLERİN ETİKETSİZ DERLEM ÜZERİNDEKİ ETKİLERİ

	Toplam kelime sayısı	Bellek Gerekşinimi (GB)		
		FastText	Word2Vec	Doc2Vec
Ham Veri Seti	1674166	3.25	2.03	4.67
NA	1192704	2.4	1.44	4.08
DA	1673708	3.24	2.03	4.66
NDA	1192246	2.4	1.44	4.08
Gİ	737042	1.7	0.89	3.01
GNDA	607501	1.46	0.74	2.83

Çalışmamızda kullandığımız etiketli derlem ise 11 farklı kategori altında birleştirilen ve etiketsiz derlemin alt kümesinden seçilen 1,644,883 adet eğitim dokümanını ve daha önce hiç kullanılmamış olan yani etiketsiz derlemden tamamen farklı olan 323,608 adet test dokümanını içermektedir. Etiketli derlem de hem ham haliyle hem de belirtilen beş adet ön işlem aşamasından geçirildikten sonra kullanılmıştır. Eğitim ve test dokümanı adedi %80-%20 oranı dikkate alınarak yaratılmıştır.

Tablo II. etiketli derlemi oluşturan eğitim ve test veri setlerinin her bir kategori altında kaç adet doküman içerdiği bilgisini göstermektedir.

TABLO II. ETİKETLİ DERLEM HAKKINDAKİ AYRINTILI BİLGİLER

Kategoriler	Eğitim Veri Seti	Test Veri Seti	Toplam
Dünya	170640	39068	209708
Eğitim	18293	1562	19855
Finans	294545	58853	353398
Gündem	437259	91775	529034
Kültür-Sanat	30628	6119	36747
Magazin	67453	13487	80940
Sağlık	47220	8233	55453
Siyaset	20745	4136	24881
Spor	371688	73690	445378
Teknoloji	116969	13088	130057
Yaşam	69443	12727	82170
Toplam (%80 Eğitim-%20 Test)	1644883	323608	1,968,491

Etiketsiz derlemde Word2Vec, Doc2Vec ve FastText ile elde edilen vektör modeller, eğitim ve test veri setlerini oluşturan haber dokümanlarının vektörleştirilmesi için kullanılmıştır. Haber dokümanları ilgili modelle elde edilen kelime vektörlerinin ortalaması alınarak sayısallaştırılmıştır.

IV. HABER METİNLERİNİN SINIFLANDIRMASI VE SONUÇLAR

Ön işlem aşamalarından geçirilen haber dokümanları Naive Bayes (NB), Destek Vektör Makineleri (DVM), Yapay Sinir ağları (YSA), Karar Ağaçları (DT) ve Lojistik Regresyon (LR) sınıflandırma algoritmaları kullanılarak sınıflandırılmıştır.

Word2Vec algoritmasının etiketsiz derleme uygulanması neticesinde elde edilen kelime vektörlerinin kullanılması ile sayısallaştırılan eğitim ve test seti üzerinde uygulanan sınıflandırma algoritmalarının başarımları Tablo III'de belirtildiği gibidir.

TABLO III. WORD2VEC SONUÇLARI

	NB	DVM	YSA	KA	LR
Ham Veri Seti	0.558	0.775	0.771	0.616	0.771
NA	0.573	0.778	0.775	0.619	0.775
DA	0.573	0.776	0.771	0.619	0.773
NDA	0.589	0.779	0.775	0.624	0.776
Gİ	0.563	0.771	0.768	0.613	0.767
GND	0.578	0.777	0.778	0.620	0.774

Tablodan görüldüğü gibi en yüksek başarımlı sonucu (0.779) noktalama işaretlerinin ve durak kelimelerinin atılması durumunda DVM sınıflandırıcısıyla elde edilmiştir. Noktalama işaretlerinin ve durak kelimelerinin atılması sınıflandırıcıların başarımlarını arttırırken, sadece gövdeleme işlemi başarımları değerlerini düşürmüştür, GND işlemi ise sınıflandırıcılar üzerinde olumlu etkiye sahip olmuştur.

Tablo IV. Doc2Vec algoritması kullanıldığında elde edilen durumu özetlemektedir.

TABLO IV. DOC2VEC SONUÇLARI

	NB	DVM	YSA	KA	LR
Ham Veri Seti	0.661	0.757	0.757	0.470	0.755
NA	0.661	0.758	0.760	0.474	0.755
DA	0.652	0.753	0.755	0.466	0.751
NDA	0.651	0.756	0.758	0.480	0.754
Gİ	0.667	0.751	0.752	0.452	0.749
GND	0.653	0.750	0.754	0.462	0.749

Tablo IV'den görüldüğü gibi en iyi başarımlı değeri (0.758) NA ön işlem aşamasında DVM'nin kullanıldığı durumda elde edilmiştir. NB'nin başarımları Word2Vec'deki duruma göre artmış, KA'nın başarımları ise yine Word2Vec'e göre azalmıştır.

Tablo V. FastText algoritması kullanıldığında sınıflandırıcıların başarımlarını belirtmektedir. Tablo V'deki verilere dayanarak en iyi sonucun (0.780) DVM sınıflandırıcısıyla elde edildiği görülmüştür. En iyi başarımlı

değerine noktalama işaretleri ve durak kelimelerini atılması durumunda erişilmiştir.

TABLO V. FASTTEXT SONUÇLARI

	NB	DVM	YSA	KA	LR
Ham Veri Seti	0.578	0.774	0.766	0.610	0.770
NA	0.582	0.780	0.777	0.615	0.777
DA	0.594	0.774	0.766	0.616	0.770
NDA	0.607	0.780	0.777	0.624	0.778
Gİ	0.579	0.766	0.764	0.601	0.763
GND	0.594	0.776	0.771	0.615	0.774

Sadece noktalama işaretlerinin atılması sınıflandırıcı sonuçlarını iyileştirirken, diğer ön işlem aşamaları sonuçları bazı durumlarda iyileştirmiş, bazı durumlar da ise düşürmüştür.

Her ne kadar başarımları arasındaki fark çok az olsa da beklenildiği gibi karakter n-gramların kullanılmasıyla test verisi daha iyi genelleştirilmiş ve en iyi başarımlı değeri FastText algoritması ile ulaşılmıştır. Genel olarak NB ve KA sınıflandırıcıları geri kalmış, DVM, YSA ve LR daha iyi sonuçlar üretmiştir.

Gelecek çalışmalarda Word2Vec, Doc2Vec ve FastText algoritmaları ile elde edilen kelime vektörleri farklı varyasyonlar ile birleştirilecek ve haber metinlerinin sınıflandırılması aşamasında topluluk öğrenmesi (ensemble learning) tekniklerinin etkileri analiz edilecektir.

KAYNAKLAR

- [1] Mikolov T, Chen K, Corrado G, Dean J. (2013), "Efficient estimation of word representations in vector space". Proceedings of Workshop at ICLR. Scottsdale, Arizona 2-4 Mayıs 2013.
- [2] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML 2014), pages 1188–1196, Beijing, China.
- [3] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.
- [4] Arabacı, M. A., Esen, E., Atar, M. S., Yılmaz, E., & Kalthoğlu, B. (2018, May). Detecting similar sentences using word embedding. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
- [5] Ay Karakuş, B., Talo, M., Hallaç, İ. R., & Aydın, G. (2018). Evaluating deep learning models for sentiment classification. *Concurrency and Computation: Practice and Experience*, e4783.
- [6] Seyfioğlu, M. S., & Demirezen, M. U. (2017, September). A hierarchical approach for sentiment analysis and categorization of Turkish written customer relationship management data. In *Computer Science and Information Systems (FedCSIS)*, 2017 Federated Conference on (pp. 361-365). IEEE.
- [7] Ozer, Z., Ozer, I., & Findik, O. (2018). Diacritic restoration of Turkish tweets with word2vec. *Engineering Science and Technology, an International Journal*.
- [8] Karasoy, O., & Ballı, S. (2017, October). Classification Turkish SMS with deep learning tool Word2Vec. In *Computer Science and Engineering (UBMK)*, 2017 International Conference on (pp. 294-297). IEEE.
- [9] Güngör, O., & Yıldız, E. (2017, May). Linguistic features in Turkish word representations. In *Signal Processing and Communications Applications Conference (SIU)*, 2017 25th (pp. 1-4). IEEE.
- [10] Şen, Mehmet Umut, and Berrin Yanıkoğlu. "Document classification of SuDer Turkish news corpora." (2018).
- [11] <https://github.com/hakkiyagiz/SIU2019>