International Symposium on Advanced Engineering Technologies (ISADET)
16-18 June 2022
Kahramanmaraş

TÜRKÇE METİNLERİN SINIFLANDIRILMASINDA ÖZNİTELİK OLARAK EKLERİN DE KULLANILMASI

Ahmet Erkan Çelik [1*]

[1]Next4biz Bilgi Teknolojileri AR-GE Merkezi, İstanbul, Türkiye

*Corresponding author: *erkan.celik@next4biz.com*

ÖZET

Günümüzde büyük boyutlu ve farklı kaynaklardan toplanan dokümanların analizinde yapay öğrenme yaklaşımları metin sınıflandırmasında kullanılmaktadır. Pek çok doğal dil işleme çalışmasında metinlerin önişlemi aşamasında kök bulma analizi yapılmaktadır. Burada amaç, ekler yoluyla çeşitli varyasyonlara sahip kelimelerin tekilleştirilmesi, böylece yapay öğrenme aşamasında her bir özniteliğin değerinin arttırılmasıdır. Fakat aslen İngilizce metinlerin işlenmesinden uyarlanan bu yöntem Türkçe için tam olarak uygunluğu tartışılmaktadır. Türkçe içerikler için kelimelerin ekleriyle beraber değerlendirildiğinde daha değerli olabileceği konusu bu çalışmada kapsamında incelenmiştir. Çalışmada, "Mobil uygulaması aracılığıyla, restoran yemek teslimatları ve market ürünleri için teslimat hizmeti" sunan bir kuruma ait Next4biz CSM ürünün veri tabanında bulunan şikâyet ve talep verileri kullanılmıştır. Veri dengesiz ve çokça hatalı sınıflandırma içermekte olduğu göz ardı edilerek çalışmalar uygulanmıştır. Önerilen model Next4biz CSM ürününe entegre olarak bulut ortamda çalışacak bir servise dönüştürülmüş ve çeşitli Next4biz müşterileri tarafından kullanılmaya başlanmıştır.

Anahtar Kelimeler: Derin Öğrenme, Doğal Dil İşleme, Gözetimli Öğrenme, Metin Sınıflandırma, Türkçe kök bulma

USING SUFFIX AS FEATURES CLASSIFICATION OF TURKISH TEXTS

ABSTRACT

Today, machine learning approaches are used in text classification in the analysis of large-sized documents collected from different sources. In many natural language processing, stemming analysis is performed in the preprocessing stage of texts. The aim here is to singularize words with various variations through suffixes, thus increasing the value of each attribute in the machine learning stage. However, this method, which was originally adapted from the processing of English texts, is being discussed as fully suitable for Turkish. For Turkish content, the subject that words can be more valuable when evaluated together with their suffixes has been examined within the scope of this experimental study. In the experimental study, complaint and request data in the database of the Next4biz CSM product belonging to an institution that provides "delivery services for restaurant food deliveries and grocery products via mobile application" was used. The deeplearning were applied by ignoring the fact that the data is unbalanced and contains many misclassifications. The proposed model has been integrated into the Next4biz CSM product and has been converted into a service that will run in the cloud environment and has been used by various clients of Next4biz.

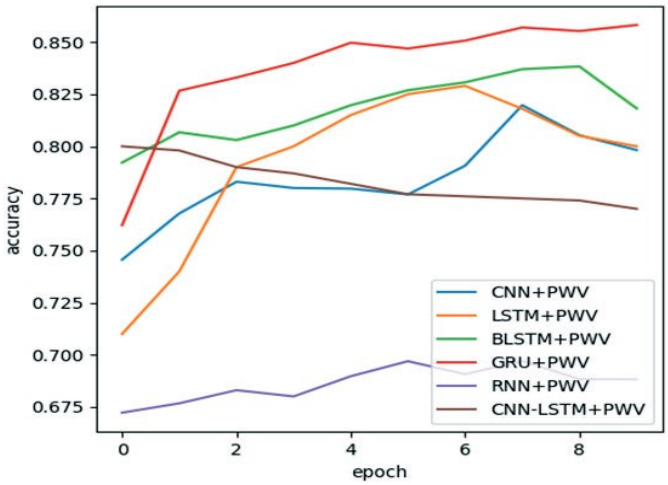
Keywords: Deep Learning, Natural Language Processing, Supervised Learning, Stemming Turkish Words, Text Classification

1. GİRİŞ

Bir kurumun yüzlerce çağrı merkezi çalışanı, her gün binlerce müşteri ile görüşmekte ve onlardan gelen şikayetleri ya da talepleri doğru kategorilere yönlendirerek bildirimlerin ilgili iş birimlerinde işleme alınabilmesini sağlamaktadırlar. Ayrıca her gün müşteri talep ya da şikayetlerini içeren yüzlerce e-posta ve mesaj benzer şekilde kurum çalışanları tarafından işlenmektedir. Next4biz CSM ürünü e-posta, çağrı merkezi ya da iletişim formu gibi çeşitli kanallardan müşterilerden talep ve şikayetleri toplayarak bu bildirimlerin ilgili iş birimlerinin önüne düşmesini sağlar. Ne var ki müşterilere ait şikayet ya da taleplerin ne hakkında olduğunu belirlemek için de azımsanmayacak ölçüde insan gücü gerekmektedir. Bu süreci otomatikleştirmek amacıyla başlattığımız projede derin öğrenme yoluyla müşterin şikayet ya da taleplerini sınıflandıran bir servis geliştirdik. Çözmeye çalıştığımız problemi "gözetimli öğrenme yoluyla metinlerin sınıflandırılması" olarak nitelendirdik. Bu servisin gelişimi sırasında en iyi yöntemi uygulayabilmek adına Geçitli Tekrarlayan Birim (Gated Recurrent Unit-GRU), Konvolüsyonel Sinir Ağı (Convolutional Neural Network-CNN), Uzun Kısa Süreli Bellek (Long Short-Term Memory-LSTM), Çift yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short-Term Memory-BLSTM), CNN katmanının çıktısının birkaç LSTM katmanını beslediği hibrit yaklaşım (CNN-LSTM) modellerini başarımlarını karşılaştıran deneyler de yaptık ve Şekil1'de de görüleceği üzere, en iyi sonuçları aldığımız, GRU tabanlı modeli bu deneyde de kullandık. Bu çalışma, metinlerin hazırlanması aşamasında, kelimeleri kökleri ile değiştirme işlemi yapılırken, alternatif bir yöntem olarak, Türkçe eklerin de ayrı birer kelime gibi cümle

International Symposium on Advanced Engineering Technologies (ISADET)
16-18 June 2022
Kahramanmaraş

içerisinde kullanılması durumunda sınıflandırıcı modelin başarısının nasıl etkileneceğini incelediğimiz empirik çalışmaları sunar.



Şekil 1. Modelin tespiti için yapılan deneydeki doğruluk değerleri

1.1 Konuya İlişkin Önceki Çalışmalar

Türkçe gibi sondan eklemeli dillerde sözcükler morfem adı verilen parçaların birbirine eklenmesi sayesinde oluşur. Kök bulma işlemi iki farklı yaklaşımla yapılmaktadır. İlki Yapım ekleri de dahil olmak üzere sözcüğün en yalın halini elde etme işlemidir. Buna "kök bulma" denir. İkincisi ise sözcün yapım ekli halini bırakarak yapılmaktadır. Buna "gövdeleme" denir ancak her iki yaklaşım da "stemming" olarak adlandırılır. Stemming için yapılan çalışmalar genel olarak kural tabanlı, hibrit ve istatistiksel olmak üzere üç başlık altında toplanabilir. [1] Metin sınıflandırılması problemlerinde birçok kelimenin aslında aynı kök ile ifade edilebilmesine bağlı olarak modelin kullanacağı öznitelik sayısının sadeleştirilebileceği yaklaşımı ön plana çıkmaktadır. [2] Farklı stemming yöntemlerinin, çalışmaların başarı oranını değiştirdiğine dair çalışmalar literatürde mevcuttur.[3] Metin sınıflandırma probleminde öznitelik belirleme aşamasında kullanılan bir başka yaklaşım ise kelimeleri değil ngramları kullanmak yönündedir. Ngramlar kelimeyi oluşturan karakterlerin belli sayıda yan yana getirilmesi yöntemine verilen genel isimdir. Her bir öbekte kaç karakter kullanıldığı ngram'ın ismini de belirler. Örneğin 2 karakterli ngramlara bigram, 3 karakterli ngramlara ise trigram adı verilir. Bu yöntem dilden bağımsız öznitelik belirlemede fayda sağlarken çok sayıda öznitelik ortaya çıkaracaktır. [4] Öte yandan ngram ile öznitelik belirlenmeden önce de stemming yapılması yaygın kullanılan bir yöntemdir. Ngramlar genellikte istatistiksel tabanlı sığ öğrenme yöntemlerinde kullanılmaktadırlar.[5] Bu çalışmada kullanılan, çekim ekleri de dahil olmak üzere eklerin herhangi bir yöntemle cümle içerisinde tutulmasına yönelik yaklaşımımız özgün olarak değerlendirilmektedir.

2. DENEYSEL ÇALIŞMA

2.1 Önişleme

Metinlerin derin öğrenme tekniği ile işlenebilmesi için doğal dil işleme kütüphaneleri ile bir ön işlemden geçirilmesi gerekmektedir. Ön işlem için kullanılan yöntemler sıralayacak olursak; İlk aşamada metnin cümlelere ayrılması: Bu işlem için Zemberek NLP [6] kütüphanesinin TurkishSentenceExtractor sınıfı kullanılmıştır. Sonra cümledeki yazım yanlışlarının otomatik olarak düzeltilmeye çalışılması: Bu işlem için de Zemberek NLP kütüphanesinin TurkishSpellChecker sınıfı kullanılmıştır. Son olarak özel bilgilerin temsili gösterimi: Engerek kütüphanesi [7] yardımıyla sayılar #number, e-posta adresleri #email bağlantıları ise #web ifadeleri ile değiştirilmiştir. Ancak Türkçe gibi eklemeli dillerde, Avrupa dillerinden (Germen dilleri, Roman dilleri vb.) farklı olarak eklerin cümlenin anlamı açısından önemli bir yer tuttuğu varsayımı ile eklerin de bir şekilde kullanılmasını sağlayan bir çalışma da gerçekleştirilmiştir. Kelime kökünü bulmak (Stemming) için[8] Snowball stemmer [9] kütüphanesi kullanılmıştır. Yeni önerdiğimiz yaklaşım için ise Zemberek kütüphanesinden faydalanılmıştır.

International Symposium on Advanced Engineering Technologies (ISADET)
16-18 June 2022
Kahramanmaraş

Örnek İçerik:

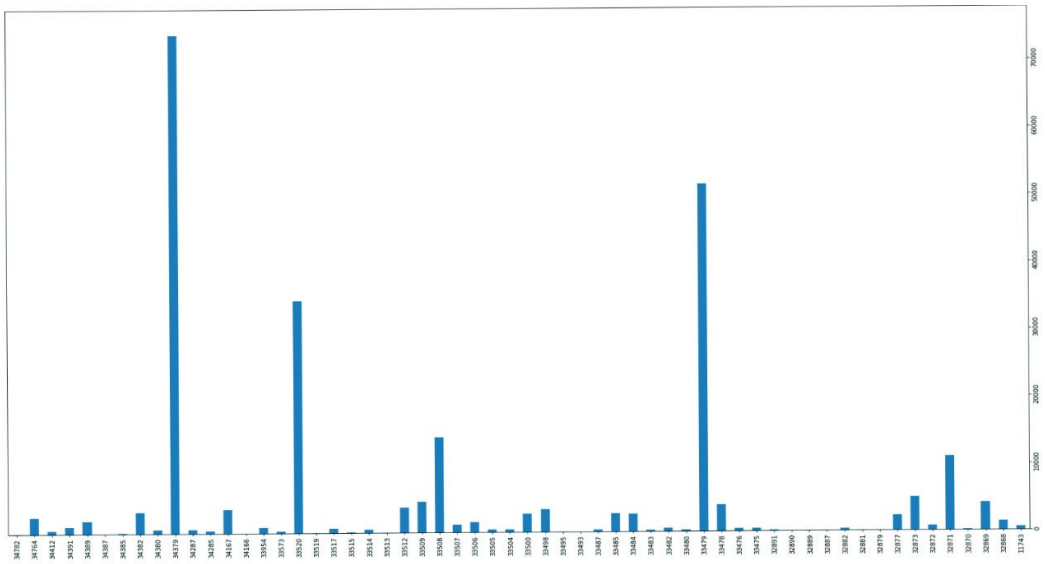
"Müşterimiz 5452541211 sipariş numaralı siparişindeki 2 adet beklediği eti lifalif yulaf ezmesi ürününün 1 adet teslim edilmesinden dolayı 25 TL İNDİRİM KUPONU TANIMLANMASI talebi yapmaktadır. Ülke: Türkiye"

Cümlesi ön işlemde snowball kullanıldığında aşağıdaki gibi olurken:
"müşter #number sipariş numaral sipariş #number adet bekledik et lifalif yulaf ezmes ürün #number adet tesl edilme dola #number tl indir kupo tanımlanmas talep yapmak ülke türki."
Zemberek ile kök, ek ve eklerin türleri tespit edilerek ekler ayrı bir kelime olarak cümleye dahil edildiğinde şöyle olmaktadır:
"müşteri miz:p1pl #number sipariş numara lı:with sipariş in:p2sg de:loc ki:rel #number adet bekle diğ:pastpart i:p3sg et i:acc lifalif yulaf ez me:inf2 si:p3sg ürün ün:p2sg ün:gen #number adet teslim et ed:verb il:pass me:inf2 si:p3sg nden:abl dolayı #number tl indirim kupon u:acc tanımla n:pass ma:inf2 sı:p3sg talep taleb:noun i:acc yap makta:prog2 dır:cop ülke türki ye:dat."

Burada ek ayrı bir kelime olarak yazılmış iki nokta üstüste (:) karakteri ile ekin türünü ifade eden kısaltma da yanına yazılmıştır. Burada kullanılan teknik aşağıdaki gerekçelerle belirlenmiştir:

i) Türkçede kullanılan eklerin bir kısmının İngilizcede ayrı bir kelime olarak kullanılıyor olması
ii) Derin öğrenme için kelimelerin vektörler olarak ifade edilmesini sağlayan Word2Vec [10] vb. kütüphanelerin kelime odaklı çalışıyor olması.
iii) Türkçedeki bazı eklerin aynı şekilde yazılan farklı türdeki eklerle ya da bağlaçlarla benzerlik gösterebiliyor olması (de, ki gibi).

2.1 Materyal



Şekil 2. Verilerin Sınıflar Üzerindeki Dağılımı

Çalışmada "Mobil uygulaması aracılığıyla, restoran yemek teslimatları ve market ürünleri için teslimat hizmeti" sunan bir kuruma ait müşteri şikâyet ve talepleri kullanılmıştır. Bu metinler, telefonla çağrı merkezi çalışanı tarafından doğrudan yazılmış ve ilişkili olduğu kategori seçilmiş olabileceği gibi, müşteriler tarafından, e-posta ya da mesaj yoluyla iletilmiş de olabilir. İkinci şekilde iletilen metinler farklı çalışanlarca okunarak ilişkili olduğu kategori belirlenmiştir. Veri seti yaklaşık 250.000 şikâyet ve talep verisi içermektedir. Çokça yazım yanlışı ve yanlış sınıflandırma içerdiği gibi aynı zamanda oldukça dengesizdir. Verideki 58 sınıfa ait veri dağılımı Şekil 2'de gösterilmiştir. Sınıfların isimleri yerine id bilgisi kullanılmıştır. Veri, önceden eğitim ve doğrulama parçalarına ayrılmış ve hem stemming, hem de kök ve ekin birlikte kullanılması yöntemleri için aynı eğitim ve doğrulama verileri ile yapılmıştır. Amacımız iki farklı ön işleme yöntemini karşılaştırmak olduğu için, ön işlem dışında tüm

International Symposium on Advanced Engineering Technologies (ISADET)
16-18 June 2022
Kahramanmaraş

diğer değişkenleri sabit tutmak tercih edilmiştir. Eğitim için toplam verinin %90'lık parçası kullanılmış, doğrulama için ise %10'luk parçası ayrılmıştır. Bu oranın belirlenmesi tamamen sezgiseldir. Her iki şekilde işlenmiş veri Word2Vec kütüphanesi ile ayrı ayrı eğitilmiş [11,12] ve ayrı ayrı vektör sözlükleri elde edilmiştir. Deneylerin gerçekleştirildiği bilgisayar ve kullanılan temel kütüphaneler den de özetle bahsedecek olursak; Ubuntu Linux 20.04 LTS, işletim sistemi, Intel i7 12 core CPU işlemci, 64 GB bellek, Nvidia Quadro P4000 8GB GPU grafik işlemci, 1 TB SSD disk, Python v3.8.7 programlama dili, CUDA v11.6 paralel programlama ara yüzü ve Tensorflow v2.8.0 yapay öğrenme çerçevesi.

2.3 Model

Deney, Keras çatısı ile iki adet GRU kullanan bir RNN modeli üzerinde gerçekleştirilmiştir. Kayıp fonksiyonu olarak Keras çatısına ait Kategorik Çapraz Düzensizlik (Categorical Crossentropy) fonksiyonu, optimizer olarak da yine Keras çatısına ait ADAM optimizasyonu tercih edilmiştir.
Modele ait Keras çatısının verdiği özet bilgiler aşağıdaki gibidir:

Snowball			**Kök ve ekin birlikte kullanımı**		
Model:	"sequential"		Model:	"sequential"	
Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 250, 100)	4991800	embedding (Embedding)	(None, 250, 100)	3531300
gru (GRU)	(None, 250, 1024)	3459072	gru (GRU)	(None, 250, 1024)	3459072
gru_1 (GRU)	(None, 256)	984576	gru_1 (GRU)	(None, 256)	984576
dense (Dense)	(None, 58)	14906	dense (Dense)	(None, 58)	14906
Total params:	9,450,354		Total params:	7,989,854	
Trainable params:	9,450,354		Trainable params:	7,989,854	
Non-trainable params:	0		Non-trainable params:	0	

2.4 Başarının Ölçülmesi

Her iki veri 7 tur eğitilmiş, her tur sonunda doğrulama verisi ile başarı ölçülmüştür. Dengesiz sınıflarda başarı ölçümü için sadece Accuracy kullanmak sağlıksız olduğu için MCC (Matthews Correlation Coefficient), F1 Score, Doğruluk (Accurracy) ve Kayıp (Loss) değerleri ayrı ayrı izlenmiştir. F1 score hesaplaması, Denklem (1)'deki Precision ve Denklem (2)'deki Recall değerlerinden faydalanılarak Denklem (3)'teki gibi yapılır. F1 Score genellikle ikili sınıflandırmada kullanılan bir ölçme yöntemi olduğu için çok sınıflı sınıflandırmada, her bir sınıf için ayrı ayrı F1 score hesaplaması yapılır ve elde edilen her bir F1 score değerinin ortalaması, en büyüğü ya da en küçüğü kullanılır. Bu çalışmada ortalama alama yöntemi kullanılmıştır.[13]

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

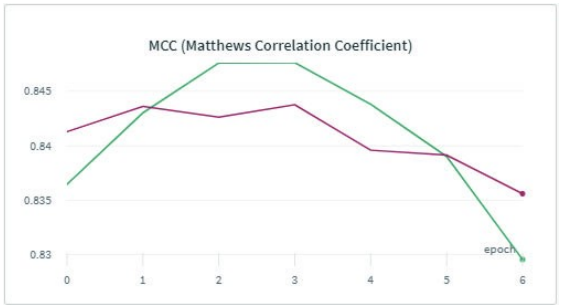
$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1\ Score = \frac{2 \text{ x } Precision \text{ x } Recall}{Precision + Recall} \tag{3}$$

International Symposium on Advanced Engineering Technologies (ISADET)
16-18 June 2022
Kahramanmaraş

MCC, özellikle çok sınıflı ve sınıf dağılımın dengesiz olduğu veri setlerinde tutarlılığını koruyabilmektedir. -1 ile +1 arasında bir çıktı üretir. Çıktı değeri 0 olduğunda sınıflandırmanın rastgele yapıldığını, -1 olduğunda ise sınıflandırıcının verdiği karar ile gerçek değerin tamamen zıt olduğunu ifade eder. 0 ile +1 aralığındaki değerler ise sınıflandırmanın ne kadar başarılı olduğunu gösterir. Hesaplanışı Denklem (4)'te görüldüğü gibidir [14]. Şekil 3'de F1 Score ölçeği için her bir epoch için her bir veri kümesi için yapılan ölçümler gözlenebilir. Burada Yeşil grafik kök ve ekin birlikte kullanıldığı veri kümesi, kırmızı grafik ise snowball ile işlenmiş veri kümesini ifade eder. Epoch 3'te her iki veri kümesi için de en yüksek f1 score değerinin ölçüldüğü grafiklerden anlaşılabilmektedir. Bu noktada kök ve ekin birlikte kullanıldığı veri kümesi için 0.8483, snowball stemmer ile işlenmiş veri kümesi için 0.8448 değerleri ölçülmüştür.

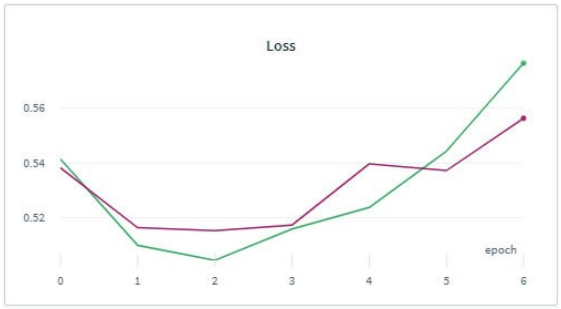
Şekil 3. Her Bir Adımda Ölçümlenen f1 Score Değeri



Şekil 4. Her Bir Adımda Ölçümlenen MCC Değeri

Şekil 4'te MCC ölçeği için her bir epoch için her bir veri kümesi için yapılan ölçümler gözlenebilir. Burada Yeşil grafik kök ve ekin birlikte kullanıldığı veri kümesi, kırmızı grafik ise snowball ile işlenmiş veri kümesini ifade eder. Yeşil grafiğin epoch 2'de kırmızı grafiğin ise epoch 3'te en yüksek MCC değerine ulaştığı grafiklerden anlaşılabilmektedir. Bu noktalarda kök ve ekin birlikte kullanıldığı veri kümesi için 0.8476, snowball stemmer ile işlenmiş veri kümesi için 0.8338 değerleri ölçülmüştür.

Şekil 5. Her Bir Adımda Ölçümlenen Accuracy (Doğruluk) Değeri



Şekil 6. Her Bir Adımda Ölçümlenen Loss (Kayıp) Değeri

Şekil 5'te accuracy ölçeği için her bir epoch için her bir veri kümesi için yapılan ölçümler gözlenebilir. Burada Yeşil grafik kök ve ekin birlikte kullanıldığı veri kümesi, kırmızı grafik ise snowball ile işlenmiş veri kümesini ifade eder. Epoch 2'de her iki veri kümesi için de en yüksek accuracy değerinin ölçüldüğü grafiklerden anlaşılabilmektedir. Bu noktada kök ve ekin birlikte kullanıldığı veri kümesi için 0.8483, snowball stemmer ile işlenmiş veri kümesi için 0.8459 değerleri ölçülmüştür. Şekil 6'da loss ölçeği için her bir epoch için her bir veri kümesi için yapılan ölçümler gözlenebilir. Burada Yeşil grafik kök ve ekin birlikte kullanıldığı veri kümesi, kırmızı grafik ise snowball ile işlenmiş veri kümesini ifade eder. Epoch 2'de her iki veri kümesi için de en düşük loss değerinin ölçüldüğü grafiklerden anlaşılabilmektedir. Bu noktada kök ve ekin birlikte kullanıldığı veri kümesi için 0.5046, snowball stemmer ile işlenmiş veri kümesi için 0.5153 değerleri ölçülmüştür.

3. SONUÇ ve DEĞERLENDİRME

Çalışma kapsamına çok sınıflı metin verilerinin sınıflandırılması probleminin çözümünde, metinlerin ön işlemi için yöntemler denenmiştir. Çalışmada geleneksel yöntemlerle stemming yapılan veri kümesi kontrol grubu olarak kullanılmış ve daha iyi bir sınıflandırma yapılacağı öngörülen kök ve ekin metinde tutulmasını sağlayan ön işlem yöntemi deney grubu olarak kullanılmıştır.

Her iki veri için de aynı eğitim ve doğrulama verileri üzerinde çalışılmış, böylece sonuçlar sağlıklı bir şekilde kıyaslanabilmiştir. Çalışma sonucunda kök ve ekin metinde tutulmasını sağlayan ön işleme sahip veri kümesinin MCC, F1 Score, accurracy ve loss ölçekleri ile en iyi sonuca 2 epoch ile ulaştığı, Snowball stemmer ile işlenen veri kümesinin ise 3 epoch ile en iyi sonuca ulaştığı anlaşılmıştır. Her iki veri kümesi için en iyi değerlere ulaşılan noktada Kök ve ekin birlikte kullanımı Snowball stemmer ile işlenen veri kümesine göre 0.0035 daha iyi F1 Score değeri, 0.0035 daha iyi MCC değeri, 0.0024 daha iyi accuracy değeri ve 0.0107 daha iyi loss değeri üretmiştir. Bu çalışma bize Türkçe kelimelerdeki eklerin de metin sınıflandırmasında değerli olduğunu göstermiştir.

4. REFERANSLAR

[1] Burcu, C. (2019). Lstm ağları ile türkçe kök bulma. Bilişim Teknolojileri Dergisi, 12(3), 183-193.

[2] Yurt, E. A. (2015). Türkçe metinlerde duygu analizi (Master's thesis, Maltepe Üniversitesi, Fen Bilimleri Enstitüsü).

[3] Düven, B. (2021). Yapay öğrenme yöntemleri ile Türkçe metinlerinde duygu analizi (Master's thesis, Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü).

[4] Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-480.

[5] ŞİRİN, Y., & KUTLUGÜN, M. A. (2020). Anlamlı ve Benzer Olmayan Türkçe Metinler Üretmek için N-Gram Yöntemi ile İstatistiksel ve Kural Tabanlı Yaklaşımın Birlikte Kullanımı. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 22(65), 331-342.

[6] Zemberek-NLP Source Code https://github.com/ahmetaa/zemberek-nlp

[7] Engerek: Turkish NLP Library for Python https://github.com/cilekagaci/engerek

[8] Doğuç, Ö., Aytaç, Ö. B., & Silahtaroğlu, G. (2020). Lemmatizer: Akıllı Türkçe kök bulma yöntemi.

[9] Snowball Stemmer Website https://snowballstem.org/

[10] Gensim Home Page https://radimrehurek.com/gensim/index.html

[11] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[12] Güngör, O., & Yıldız, E. (2017, May). Linguistic features in Turkish word representations. In 2017 25th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE

[13] Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756.

[14] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure, 405(2), 442-451.