Veri Artırımı ve BERT ile Türkçe Duygu Analizi

Talha Bacak*

16011038

* *Bilgisayar Mühendisliği, Elektrik-Elektronik Fakültesi*
Yıldız Teknik Üniversitesi, İstanbul, Turkiye

Özet — Verilerin augmentation işlemleri ile artırarak eğitilen Türkçe BERT modeli ve augmentation uygulanmamış veri ile eğitilen modelin karşılaştırılması.

Anahtar Kelimeler — BERT, duygu analizi, veri artırımı, doğal dil işleme

Abstract — Comparison of the Turkish BERT model trained by augmenting the data with augmentation operations and the model trained with the non-augmented data.

Keywords — BERT, sentiment analysis, augmentation, natural language process

I. KONUSU

Türkçe metinlerden faydalanılarak olumlu ya da olumsuz duygu analizi yapan iki BERT modeli geliştirilmiştir. Modellerden biri augmentation işlemleri uygulanmış bir veri ile eğitilmişken, diğer model augmentation işlemi uygulanmamış veri ile eğitildi. Bu iki model incelenmiştir.

II. İZLENEN YOL

İlgili modeli eğitmek için öncelikle gerekli veri seti oluşturuldu. Bu veri seti açık kaynaklı olarak paylaşılan birçok veri setinin optimize edilerek birleştirilmesi sonucu elde edildi. Yaklaşık 125.000 Türkçe cümleden oluşan etiketlenmiş veri elde edildi. Bu veriler üzerine de yapay artırım yapmanın faydalı olabileceği düşünüldüğü için augmentation yöntemleri kullanıldı. Bu sayede bulunan veri 375.000 Türkçe cümleden oluşan etiketlenmiş veriye çıkartıldı. 2 farklı augmentation işlemi yapıldı. Augmentation işlemlerinden biri cümleden random bir şekilde kelime silmek, diğeri ise random bir şekilde kelimelerin yerlerini değiştirmek oldu. Elde edilen verileri eğitmek için BERT modeli kullanıldı. Bu model google'un çıkarttığı ve embedding layer kısmında cümledeki bir kelimeyi maskeleme yaptıktan sonra onu tahmin etmeye çalışması ile başarısını artırmayı amaçlamaktadır. Bu özelliği ve elde edilen veri setinin büyüklüğü nedeni ile eğitim oldukça yavaş sürmüştür. Ayrıca Türkçe için BERT modelinden oluşturulmuş bir modeli geliştiren Stefan Schweter'ın modeli (https://github.com/stefan-it/turkish-bert) kullanıldı. Model zaman kısıtından dolayı 2 epoch eğitilmiştir. 1 epoch süresi augmentation yapılmış veri ile eğitilen modelin yaklaşık 32 saat (Nvidia GTX1050), augmentation işlemi yapılmamış veri ile eğitilen modelin ise yaklaşık 11 saat (Nvidia GTX1050) sürmüştür.

III. KULLANILAN ARAÇLAR, YÖNTEMLER

- Programlama dili olarak Python dili kullanıldı.
- Ortam olarak Jupyter Notebook kullanıldı.
- Google Colab'ın süre aşımından dolayı eğitim için Nvidia GTX 1050 ekran kartı kullanıldı.
- Veri için Kaggle'dan faydalanıldı.

• Augmentation işlemi için textaugment (https://github.com/dsfsi/textaugment) kütüphanesi kullanıldı.

• Textaugment kütüphanesinde bulunan EDA (Easy Data Augmentation) yöntemleri tercih edildi. Bu yöntemin 2 metodu kullanıldı:

1. Random Deletion metodu,
2. Random Swap metodu

Textaugment Kütüphanesi:

TextAugment, doğal dil işleme uygulamaları için metni genişletmeye yönelik bir Python 3 kitaplığıdır. TextAugment, NLTK, Gensim ve TextBlob kullanmaktadır.

- Manuel çaba harcamadan model performansını iyileştirmek için sentetik veriler oluşturur.
- Basit, hafif, kullanımı kolay.
- Herhangi bir makine öğrenimi frameworkünü destekler. (ör. PyTorch, TensorFlow, Scikit-learn)
- Metin verilerini destekler.

Kütüphanenin içerdiği augmentation yöntemleri:

1. Word2vec-based augmentation
2. WordNet-based augmentation
3. RTT-based augmentation
4. EDA teknikleri:
    - Synonym Replacement
    - Random Deletion
    - Random Swap
    - Random Insertion
5. Mixup augmentation

Kullanılan Yöntemler

Random Deletion metodu:

Bu metodun tercih edilme sebeplerinden biri kullanılan dilden bağımsız olarak kullanılabilmesidir. Diğer sebep ise bu yöntemin basit ve etkili olmasıdır. Bu metod cümlelerden belli olasılıkla kelimeleri silmektedir.

Random Swap metodu:

Bu metodun tercih edilme sebebi random deletion metodunda olduğu gibi kullanılan verinin dilinden bağımsız olması ve basit olmasıdır. Ayrıca etkili olduğunun önceki çalışmalarla bilinmesidir. Bu metod belli bir olasılıkla cümlelerdeki kelimelerin yerlerini değiştirmektedir.

IV. ALINAN BAŞARI, SONUÇ

2 epoch eğitilen 2 modelimizin de sonuçları şu şekildedir (Model1: Augmentation uygulanmış model):

F1 / Model	Model1	Model2
F1 Score	% 94.74	% 94.44

Tablo 4.1

Accuarcy / Model	Model1	Model2
Accuarcy (Olumlu)	% 91.47	% 89.57
Accuarcy (Olumsuz)	% 96.84	% 97.13
Accuarcy (Toplam)	% 94.94	% 94.16

Tablo 4.2

Loss / Model	Model1	Model2
Training Loss	0.2124	0.2569
Validation Loss	0.2687	0.3044

Tablo 4.3

Zaman / Model	Model1	Model2
Training Süresi (per epoch)	32 saat	11 saat

Tablo 4.4

Sonuçlara göre augmentation uygulanan modelin belli bir düzeyde sonuçları daha iyi gelmiştir. Elde bulunan veri setinin daha optimize edilmesi ve epoch değerinin artırılması ile daha başarılı sonuçlar elde edilmesi beklenmektedir.