ELEŞTİRİ SINIFLANDIRMA

Anıl KARABULUT
Mühendislik Fakültesi
Trakya Üniversitesi

Azad KÖL
Mühendislik Fakültesi
Trakya Üniversitesi

Samet AKIŞIK
Mühendislik Fakültesi
Trakya Üniversitesi

Muhamed CURRİ
Mühendislik Fakültesi
Trakya Üniversitesi

Olumlu-Olumsuz Film Eleştirisi Sınıflandırmasında Terim Ağırlıklandırma

Öz

Günümüzde çevrimiçi platformlar aracılığıyla kullanıcılar, izledikleri filmler hakkında yoğun biçimde içerik üretmektedir. Bu durum, özellikle IMDB gibi geniş film eleştirisi veri tabanlarında olumlu veya olumsuz olarak etiketlenmiş incelemelerin doğru bir biçimde sınıflandırılmasını önemli hale getirmektedir. Doğru sınıflandırma hem film yapımcıları hem de izleyiciler için değerli içgörüler sağlayarak karar verme süreçlerini iyileştirebilir. Bu çalışma kapsamında, IMDB film eleştirisi veri setinde üç farklı popüler terim ağırlıklandırma yönteminin sınıflandırma performansı, dört farklı sınıflandırıcı ve bir sinir ağı modeli yardımıyla incelenmiştir. Elde edilen sonuçlar, özellikle belirli bir terim sayısı eşiğinden (örneğin 50 terim) sonra terim ağırlıklandırma şemalarının potansiyel performansının daha iyi yansıtılabildiğini ve metin içeriklerinin akılcı biçimde ağırlıklandırılmasının olumlu-olumsuz ayrımında önemli olduğunu göstermiştir.

Anahtar Kelimeler: Duygu analizi, terim ağırlıklandırma, IMDB veri seti, film eleştirisi sınıflandırma, metin madenciliği, sinir ağları

1. Giriş

Günümüzde dijital platformlar, kullanıcıların izledikleri filmler hakkındaki görüşlerini rahatlıkla paylaşabildiği alanlar haline gelmiştir. Özellikle IMDB gibi geniş kullanıcı kitlesine sahip platformlarda biriken milyonlarca film eleştirisi, potansiyel izleyicilerin film tercihlerini ve üreticilerin pazarlama stratejilerini etkilemektedir. Bu eleştirilerin makine öğrenmesi ve metin madenciliği yöntemleriyle olumlu veya olumsuz olarak otomatik sınıflandırılması, film öneri sistemlerinde kalite artışını sağlarken aynı zamanda kullanıcılara doğru bilgiyi sunarak zaman kazandırır.

Ancak, doğal dil işleme (NLP) alanındaki zorluklar, metinlerin bağlamını anlamayı, özellikle de kısa veya uzun eleştirilerdeki olumlu-olumsuz ifadeleri tespit etmeyi gerektirir. Bu nedenle, ham metin verisinden anlamlı özellik çıkarma, bu özellikleri ağırlıklandırma ve sınıflandırma aşamaları kritik önem taşır.

Literatürde, e-posta spam filtrelemeden ürün yorumlarının (review) sınıflandırılmasına kadar pek çok alanda terim ağırlıklandırma yöntemlerinin metin sınıflandırma performansına katkıda bulunduğu gösterilmiştir . Film eleştirisi verileri üzerinde yapılan çalışmalar da benzer şekilde TF-IDF, TF-RF, TF-IGM gibi popüler terim ağırlıklandırma yöntemlerinin olumlu-olumsuz duygu analizindeki rolünü vurgulamaktadır . Bununla birlikte, farklı dil ve veri setlerinin kendine özgü özellikleri, terim ağırlıklandırma yöntemlerinin etkililiğini değiştirebilmekte ve daha fazla analize ihtiyaç duymaktadır.

Bu çalışmada, IMDB film eleştirisi veri seti üzerinde popüler terim ağırlıklandırma yöntemlerinin, olumlu-olumsuz sınıflandırma problemine etkisi analiz edilmiştir. Üç temel odak noktası bulunmaktadır: (i) Terim ağırlıklandırma yöntemlerinin sınıflandırma başarımına katkısı, (ii) Düşük ve yüksek sayıda öznitelikle (terim) çalışmanın etkileri ve (iii) Farklı sınıflandırıcılar (ör. SVM, KNN) üzerinden elde edilen performans farklarının incelenmesi ve sinir ağlarıyla kıyaslanması. Deneysel bölümde seçilen IMDB veri seti, 3 farklı terim ağırlıklandırma yöntemi ve 5 farklı sınıflandırıcı kullanılarak sınıflandırma performansları Accuracy cinsinden hesaplanmış ve sonuçlar yorumlanmıştır.

2. Deneysel Çalışma

Bu bölümde IMDB veri seti, ön işleme adımları, öznitelik seçimi, terim ağırlıklandırma yöntemleri ve sınıflandırma deneyleri ayrıntılı biçimde ele alınmaktadır.

2.1. Veri Seti

Deneyler için kullanılan IMDB film eleştiri veri seti, önceden etiketlenmiş olan olumlu ve olumsuz film yorumlarından oluşmaktadır. Veri seti geniş bir doküman sayısına sahip olup kullanıcıların filmler hakkındaki kapsamlı görüşlerini içermektedir. Eğitim ve test bölümü olarak veri seti genellikle %80 eğitim, %20 test oranında bölünerek kullanılmıştır (Bu oranlar kullanıcı tarafından güncellenebilir).

2.2. **Ön İşleme ve Öznitelik Seçimi**

Veri setindeki her bir film eleştirisi, metin işleme adımlarından geçirilmiştir. Bu adımlar, metinlerin doğru ve etkili bir şekilde modellenebilmesi için gereklidir. Aşağıda, bu ön işleme adımlarının detayları verilmiştir:

- **Büyük/Küçük Harfe Dönüştürme:** Metindeki tüm harfler küçük harfe dönüştürülerek tutarsızlıklar ortadan kaldırılır. Örneğin, "Film" ve "film" aynı kelime olarak değerlendirilir.
- **Noktalama İşaretlerinin Temizlenmesi:** Noktalama işaretleri (örneğin, virgül, nokta, soru işareti) çıkarılır, çünkü çoğu zaman anlam taşımazlar ve modelin doğruluğunu olumsuz etkileyebilirler.
- **Rakamların Kaldırılması:** Sayılar, modelin anlamını etkilemeyen bilgiler olabilir. Bu yüzden, sayılar metinden çıkarılır veya bazı durumlarda özel bir etiketle değiştirilir.
- **Anlamı Olmayan Kelimelerin (Stop-Words) Çıkarılması:** İngilizce durdurma kelimeleri (örneğin, "the", "and", "is") metinden çıkarılır. Bu kelimeler genellikle sınıflandırma işlemi için fazla bilgi taşımazlar.
- **Gerekirse Gövde Bulma (Lemmatization) İşlemleri:** Lemmatizasyon, kelimelerin köklerine indirgenmesidir. Örneğin, "running" kelimesi "run" köküne dönüştürülür. Bu işlem, dilin anlamını kaybetmeden kelimeleri normalleştirir.

Öznitelik Seçimi: Ön işleme adımlarından sonra elde edilen kelimeler, "Bag of Words" (BoW) modeli ile temsil edilir. Bu modelde, metinler sırasıyla terimler (kelimeler) üzerinden sayısal vektörlere dönüştürülür. Her terim, metinde yer aldığı frekansa göre bir ağırlık değeri taşır. Ancak, terimler arasındaki ilişkileri anlamak ve en ayrıştırıcı terimleri seçmek için **Ki-Kare (Chi-Squared)** gibi istatistiksel öznitelik seçim yöntemleri kullanılır.

2.3. **Terim Ağırlıklandırma**

Bu çalışmada, film eleştirisi verisinin sınıflandırılmasında beş popüler terim ağırlıklandırma yöntemi kullanılmıştır:

1. **TF-IDF (Term Frequency-Inverse Document Frequency):**

    - **TF-IDF**, kelimenin bir dokümanda ne kadar önemli olduğunu belirlemek için yaygın olarak kullanılan bir yöntemdir.

    - **TF (Term Frequency)**, kelimenin bir dokümanda kaç defa geçtiğini ölçer. Bu, kelimenin o dokümanda ne kadar önemli olduğunu gösterir.

    - **IDF (Inverse Document Frequency)**, kelimenin tüm dokümanlar arasında yaygın olup olmadığını ölçer. Yaygın kelimeler (örneğin, "the", "is") düşük IDF değerine sahiptir, çünkü bu kelimeler her dokümanda bulunur ve ayırt edici değeri düşer.

    - **TF-IDF**: Bu ikisinin çarpımı, kelimenin hem belirli bir dokümanda hem de genel olarak ne kadar önemli olduğunu gösterir. Yüksek TF-IDF değeri, kelimenin hem sık kullanıldığı hem de az rastlanan bir kelime olduğunu belirtir.

Formül:

$$TF - IDF\ (t, d) = TF(t, d) \times IDF(t)$$

Burada:

- **TF(t, d)**: Belirli bir terim **t**'nin doküman **d** içindeki frekansı.
- **IDF(t)**: Tüm dokümanlarda terim **t**'nin önemini ölçen terim ağırlığı.

TF-IDF Kullanımı:

- Bu yöntem, özellikle metin sınıflandırma ve bilgi arama gibi uygulamalarda yaygın olarak kullanılır, çünkü sık karşılaşılan ama az anlam taşıyan kelimelerin ağırlığını azaltır.

2. **Bag of Words (BoW):**

- **BoW**, metinlerin sayısal bir formata dönüştürülmesi için kullanılan basit bir modeldir. Her metin, içindeki her bir kelimenin sıklığına dayalı olarak bir vektörle temsil edilir.
- Bu vektörlerin her bir boyutu, belirli bir kelimenin metindeki sıklığını gösterir.
- BoW, metindeki kelimelerin sırasını dikkate almaz, sadece kelimelerin varlığını ve sıklığını inceler.
- Bu modelde, kelimeler arasında ilişkiler ve bağlamlar göz ardı edilir. Bu nedenle, bazı durumlarda BoW modelinin doğruluğu sınırlı olabilir.

TF-IDF ve Bag-of-Words Karşılaştırması:

- **TF-IDF**'nin en büyük avantajı, metindeki kelimelerin önem derecelerini dikkate almasıdır. Yani, yaygın kelimeler düşük ağırlık alırken nadir bulunan ve dokümanda belirli bir konuyu temsil eden kelimeler yüksek ağırlık alır.
- **BoW** modelinde, kelimeler sadece sıklıklarına göre değerlendirilir, dolayısıyla bazı kelimeler modelin kararlarını olumsuz yönde etkileyebilir. Örneğin, çok yaygın olan kelimeler her zaman önemliymiş gibi değerlendirilir.

Her iki model de metinlerin sayısal temsillerini oluşturur ancak farklı yöntemlerle ağırlıklandırma yapar. **TF-IDF** genellikle metin sınıflandırma ve bilgi alma sistemlerinde daha başarılı sonuçlar verir çünkü kelimelerin sıklığından çok, onların ne kadar özel olduğuna daha fazla odaklanır.

2.4. **Sınıflandırma ve Değerlendirme**

KNN (K-Nearest Neighbors):

KNN algoritması, verilen bir veri noktasının en yakın komşularına bakarak sınıfını belirler. Bu algoritma, metin sınıflandırma problemlerinde TF-IDF veya BoW özellikleriyle birlikte kullanıldığında anlamlı performans gösterebilir. Ancak KNN'in hesaplama maliyeti yüksek olabileceği için veri setinin büyüklüğü önemlidir.

KNN (K-Nearest Neighbors)

KNN, Euclidean mesafesi gibi metriklere dayanarak sınıflandırma yapar. İki nokta x_1 ve x_2 arasındaki mesafe şu şekilde hesaplanabilir:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1,i} - x_{2,i})^2}$$

Bu formül, KNN'nin en yakın komşuları belirlemesinde kullanılan temel hesaplama yöntemidir.

KNN'de benzerlik ölçümü için yaygın olarak kullanılan bir yöntem **Kosinüs Benzerliği (Cosine Similarity)**'dir. İki vektör A ve B arasındaki kosinüs benzerliği şu şekilde hesaplanır:

$$\text{Cosine Similarity} = \cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

SVM (Support Vector Machines):

SVM, iki sınıfı birbirinden ayıracak en iyi hiperdüzlemi bulan bir sınıflandırma algoritmasıdır. Film eleştirilerinde olumlu ve olumsuz ifadeleri ayırırken SVM algoritmasının genelleşirme kapasitesi TF-IDF ve BoW yöntemleriyle daha da geliştirilmiştir.

SVM'nin temel hedefi, iki sınıfı ayıran en iyi hiperdüzlemi bulmaktır. Bu doğrusal düzlemi tanımlayan karar sınırı şu şekildedir:

$$w \cdot x + b = 0$$

Burada w, hiperdüzlemin normal vektörü; x, veri noktası; b, bir skalar terimdir. Sınıflandırma, bu sınırın hangi tarafında olduğuna bağlıdır:

$$y = \text{sign}(w \cdot x + b)$$

Naive Bayes:

Naive Bayes, bir terimin olasılığını, diğer terimlerden bağımsız olarak hesaplayan basit ama etkili bir algoritmadır. Doğal dil işleme alanında, özellikle metin sınıflandırma problemlerinde yaygın olarak kullanılır. Bu algoritma, TF-IDF ve BoW tabanlı özelliklerle birleştirildiğinde etkili bir sınıflandırma sağlayabilir.

Naive Bayes sınıflandırıcı şu olasılık kuralına dayanır:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

Burada C, bir sınıf etiketi; X, bir özellik vektörüdür. $P(C|X)$, X gözleminin C sınıfına ait olma olasılığını temsil eder.

Performans değerlendirmesi, genel başarıyı ölçen bir metrik olan doğruluk (accuracy) kullanılarak yapılmıştır. Doğruluk, doğru sınıflandırılan örneklerin toplam örnek sayısına oranı olarak hesaplanır ve modelin genel performansını gösterir.

Accuracy, genel olarak modelin doğru sınıflandırdığı örneklerin toplam örnek sayısına oranıdır ve aşağıdaki gibi hesaplanır:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Bu formülde, TP doğru pozitif sayısını, TN doğru negatif sayısını, FP yanlış pozitif sayısını ve FN yanlış negatif sayısını temsil eder.

LSTM (Long Short-Term Memory):

LSTM, zaman serisi ve doğal dil işleme gibi çalışmalarda sıklıkla kullanılan bir yapay sinir ağı modelidir. LSTM'ler, verilerdeki uzun vadeli bağlamlara odaklanarak, metin sınıflandırma problemlerinde bağlamsal bilgiyi daha iyi yakalayabilir. Film eleştirilerindeki uzun metinlerde olumlu ve olumsuz ifadeleri algılama kapasitesi nedeniyle tercih edilmiştir.

LSTM'de hücre durumu ve kapı mekanizmaları için matematiksel ifadeler şunlardır:

1. Unutma kapısı (f_t):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. Giriş kapısı (i_t) ve aday değer ($\tilde{C}_t$):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

3. Hücre durumu (C_t):

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

4. Çıkış kapısı (o_t) ve hücre çıktısı (h_t):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

GRU (Gated Recurrent Unit):

GRU, LSTM'ye benzer bir yapıdır ancak daha az parametre kullanarak daha hafif bir hesaplama maliyeti sunar. GRU, LSTM'ye kıyasla daha az karmaşıklıkla bağlamsal bilgiyi yakalayabilir ve kısa metinlerde etkili bir performans sağlar.

GRU'daki kapı mekanizmaları şu şekilde ifade edilir:

1. Güncelleme kapısı (z_t):

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

2. Yenileme kapısı (r_t):

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

3. Aday hücre durumu ($\tilde{h_t}$) ve çıktı (h_t):

$$\tilde{h_t} = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h)$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h_t}$$

3. Deneysel Sonuçlar

3.1. IMDB Veri Setinde KNN Modeli Kullanılarak Yapılan Sınıflandırma Sonuçları

Aşağıda sonuçlar tablo halinde verilmiştir.

Test için ayrılan kısım %20 olarak ayarlanmıştır ve random_state = 0 alarak deneyler gerçekleştirilmiştir.

Tablo 3.1.1 IMDB Veri Setinde KNN(k=1) ve ngram_range(1,1) ile Elde Edilen Accuracy Sonuçları

Method	Metric	Accuracy
Bag-of-Words	Cosine	0.62
	Euclidean	0.61
TF-IDF	Cosine	0.74
	Euclidean	0.74

Tablo 3.1.2 IMDB Veri Setinde KNN(k=5) ve ngram_range(1,1) ile Elde Edilen Accuracy Sonuçları

Method	Metric	Accuracy
Bag-of-Words	Cosine	0.65
	Euclidean	0.64
TF-IDF	Cosine	0.76
	Euclidean	0.76

Tablo 3.1.3 IMDB Veri Setinde KNN(k=9) ve ngram_range(1,1) ile Elde Edilen Accuracy Sonuçları

Method	Metric	Accuracy
Bag-of-Words	Cosine	0.65
	Euclidean	0.66
TF-IDF	Cosine	0.77
	Euclidean	0.77

Tablo 3.1.4 IMDB Veri Setinde KNN(k=1) ve ngram_range(1,2) ile Elde Edilen Accuracy Sonuçları

Method	Metric	Accuracy
Bag-of-Words	Cosine	0.60
	Euclidean	0.58
TF-IDF	Cosine	0.72
	Euclidean	0.72

Tablo 3.1.5 IMDB Veri Setinde KNN(k=5) ve ngram_range(1,2) ile Elde Edilen Accuracy Sonuçları

Method	Metric	Accuracy
Bag-of-Words	Cosine	0.63
	Euclidean	0.62
TF-IDF	Cosine	0.73
	Euclidean	0.73

Tablo 3.1.6 IMDB Veri Setinde KNN(k=9) ve ngram_range(1,2) ile Elde Edilen Accuracy Sonuçları

Method	Metric	Accuracy
Bag-of-Words	Cosine	0.63
	Euclidean	0.63
TF-IDF	Cosine	0.75
	Euclidean	0.75

3.2. IMDB Veri Setinde Naive Bayes Modeli Kullanılarak Yapılan Sınıflandırma Sonuçları

Tablo 3.2.1 IMDB Veri Setinde ngram_range(1,1) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.85
TF-IDF	0.86

Tablo 3.2.2 IMDB Veri Setinde ngram_range(1,2) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.88
TF-IDF	0.88

Tablo 3.2.3 IMDB Veri Setinde ngram_range(1,3) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.88
TF-IDF	0.88

Tablo 3.2.4 IMDB Veri Setinde ngram_range(1,4) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.88
TF-IDF	0.88

Tablo 3.2.5 IMDB Veri Setinde ngram_range(1,5) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.88
TF-IDF	0.88

Tablo 3.2.6 IMDB Veri Setinde ngram_range(2,3) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.88
TF-IDF	0.88

Tablo 3.2.7 IMDB Veri Setinde ngram_range(2,4) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.88
TF-IDF	0.88

Tablo 3.2.8 IMDB Veri Setinde ngram_range(3,4) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.80
TF-IDF	0.79

Tablo 3.2.9 IMDB Veri Setinde ngram_range(3,5) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.80
TF-IDF	0.79

Tablo 3.2.10 IMDB Veri Setinde ngram_range(4,5) ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.84
TF-IDF	0.83

3.3. IMDB Veri Setinde SVM Kullanılarak Yapılan Sınıflandırma Sonuçları

Tüm modellerde random_state parametresi 42 olarak belirlenmiştir.

Tablo 3.3.1 IMDB kernel='poly',C=4, degree=4,gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.64
TF-IDF	0.73

Tablo 3.3.2 IMDB kernel='rbf',C=3, degree=3,gamma='auto' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.84
TF-IDF	0.50

Tablo 3.3.3 IMDB kernel='linear',C=3, degree=3,gamma='auto' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.82
TF-IDF	0.85

Tablo 3.3.4 IMDB kernel='linear',C=2, degree=4,gamma='auto' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.82
TF-IDF	0.86

Tablo 3.3.5 IMDB kernel='linear',C=3, degree=2,gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.82
TF-IDF	0.85

Tablo 3.3.6 IMDB kernel='sigmoid',C=3, degree=2,gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.67
TF-IDF	0.85

Tablo 3.3.7 IMDB kernel='sigmoid',C=1, degree=6,gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.66
TF-IDF	0.87

Tablo 3.3.8 IMDB kernel='rbf',C=1, degree=6, gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.86
TF-IDF	0.88

Tablo 3.3.9 IMDB kernel='rbf',C=2, degree=5, gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.86
TF-IDF	0.87

Tablo 3.3.10 IMDB kernel='rbf',C=1, degree=2,gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.86
TF-IDF	0.88

Tablo 3.3.11 IMDB kernel='rbf',C=1, degree=3,gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.86
TF-IDF	0.88

Tablo 3.3.12 IMDB kernel='rbf',C=1, degree=4,gamma='scale' ile Elde Edilen Accuracy Sonuçları

Method	Accuracy
Bag-of-Words	0.66
TF-IDF	0.66

3.4. IMDB Veri Setinde Yapay Sinir Ağları Kullanılarak Yapılan Sınıflandırma Sonuçları

Tablo 3.4.1 Elde Edilen Accuracy Sonuçları

Method	GloVe(Yok)	Accuracy
LSTM	LSTM(64)	0.8812
	LSTM(128)	0.8821
GRU	GRU(64)	0.8783
	GRU(128)	0.8812

Tablo 3.4.2 Elde Edilen Accuracy Sonuçları

Method	GloVe(Var)	Accuracy
LSTM	LSTM(64)	0.8583
	LSTM(128)	0.8727
GRU	GRU(64)	0.8581
	GRU(128)	0.8804

4. Sonuç ve Öneriler

Bu çalışma, IMDB film eleştiri veri seti üzerinde popüler terim ağırlıklandırma yöntemlerinin olumlu/olumsuz sınıflandırma başarımına etkisini incelemiştir. Elde edilen bulgular, makul bir öznitelik seçimi ve uygun terim ağırlıklandırma şemalarının duygu analizi problemlerinde kritik önem taşıdığını ortaya koymuştur.

Gelecekteki çalışmalarda:

- Farklı dil ve türdeki film eleştiri veri setleri üzerinde terim ağırlıklandırma yöntemlerinin performansı incelenebilir.
- Derin öğrenme tabanlı embedding yaklaşımları ile klasik terim ağırlıklandırma yöntemleri karşılaştırılabilir.
- Daha gelişmiş öznitelik seçimi yöntemleri ile terim ağırlıklandırma yöntemlerinin etkileşimi araştırılabilir.

Referanslar

- https://pandas.pydata.org
- https://keras.io
- https://numpy.org
- https://scikit-learn.org
- https://scikit-learn.org/stable/modules/svm.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://scikit-learn.org/stable/modules/naive_bayes.html
- https://keras.io/api/layers/recurrent_layers/lstm/
- https://keras.io/api/layers/recurrent_layers/gru/

GitHub proje linki:

https://github.com/anilkrblt/criticism_classification/tree/final