BYM308 - Yapay Zekaya Giriş

Ödev 4 - Word Embedding Rapor

Grup Bilgileri, Katkıları ve İş Bölümü:

- **Grup Numarası:** 6
- Yunus Emre Yıldırım – 220601014 - 36%
  - Word2Vec, GloVe, FastText yöntemlerinin karşılaştırmalı analizi
  - NLP problem türünün belirlenmesi (duygu analizi, metin sınıflandırma vb.)
  - Çözüm metodolojisinin tasarlanması
  - Değerlendirme metriklerinin belirlenmesi
  - Temel metin ön işleme (tokenization, lemmatization vb.)
  - Geleneksel metin temsil yöntemlerinin uygulanması (Bag of Words, TF-IDF)
  - Embedding olmayan model eğitimi ve değerlendirilmesi
- Fatma Cüre – 220601023 - 36%
  - Word2Vec, GloVe, FastText yöntemlerinin karşılaştırmalı analizi
  - Embedding'li Model eğitimi ve parametrelerin optimize edilmesi
  - Sonuçların değerlendirilmesi
  - Embedding vektörlerinin projector.tensorflow.org'a aktarılması
  - Farklı görselleştirme konfigürasyonlarının denenmesi
  - Görselleştirmelerin analizi ve yorumlanması
  - Word embedding kullanan ve kullanmayan modellerin performans karşılaştırması
  - İstatistiksel analiz ve grafiklerin hazırlanması
- Yusuf Özpamuk - 220601012 - 28%
  - Raporlama
  - Potansiyel veri setlerinin taranması (Kaggle, GitHub, Huggingface)
  - Seçilen veri setinin incelenmesi ve analizi
  - Veri setinin ön işleme için hazırlanması

İçerikler

Word Embedding Nedir? 3
Hangi Yöntemler ile Gerçekleştirilebilir 3
1. Word2Vec 3
2. GloVe (Global Vectors for Word Representation) 3
3. FastText 4
Çok Boyutlu Vektörlerin Görselleştirmesi 4
Problem Tanımı ve Elde Edilen Çözümler 6
Embedding Kullanılmayan Model 6
Word Embedding Kullanılan Model 7
Değerlendirme 7
Github Linki: 8
Kaynakça 8

Word Embedding Nedir?

Word embedding, kelimeleri bilgisayarların anlayabileceği şekilde sayısal vektörlere dönüştürme yöntemidir. Bu vektörler, kelimeler arasındaki anlamsal ilişkileri ve bağlam bilgilerini yakalamayı amaçlar.

Örneğin, "kral" ve "kraliçe" gibi kelimeler birbirine yakın vektörlerle temsil edilirken, "masa" gibi alakasız bir kelimenin vektörü çok daha uzakta olur. Bu şekilde, model sadece kelimenin kendisini değil, anlamını ve kullanıldığı bağlamı da öğrenebilir.

Word embedding, kelimeler arasında benzerlikleri yakalamaya olanak tanır ve bu da özellikle duygu analizi, metin sınıflandırma, çeviri gibi NLP görevlerinde büyük avantaj sağlar.

Hangi Yöntemler ile Gerçekleştirilebilir

Word embedding işlemi, kelimeleri vektör haline getirmenin farklı yöntemleriyle gerçekleştirilebilir. İşte en yaygın kullanılan word embedding yöntemleri:

1. Word2Vec

Word2Vec, Google tarafından geliştirilen ve kelimeleri anlamlarına göre sayısal vektörlerle temsil eden bir yöntemdir. İki ana mimarisi vardır:

- CBOW (Continuous Bag of Words): Çevresindeki kelimelere bakarak bir hedef kelimeyi tahmin eder.
- Skip-Gram: Hedef kelimeye bakarak çevresindeki kelimeleri tahmin eder.

Word2Vec, kelimeler arasındaki anlamsal benzerlikleri yüksek doğrulukla yakalayabilir. Örneğin, "kral - adam + kadın = kraliçe" gibi vektörel çıkarımlar yapılabilir.

2. GloVe (Global Vectors for Word Representation)

GloVe, Stanford Üniversitesi tarafından geliştirilen bir word embedding yöntemidir. Word2Vec'ten farklı olarak hem yerel (lokal) hem de global istatistikleri dikkate alır. Kelimelerin birlikte görülme sıklıklarına (co-occurrence) dayalı olarak kelime vektörleri üretir.

Avantajı, hem sık geçen kelimeleri hem de daha az görülen ama anlamlı ilişkileri modelleyebilmesidir.

3. FastText

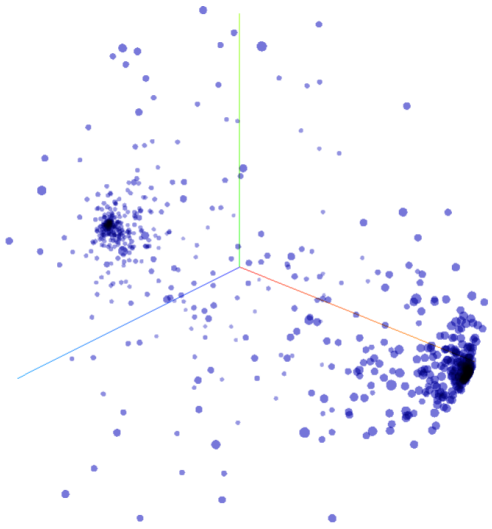
Facebook tarafından geliştirilen FastText, kelimeleri sadece kelime olarak değil, aynı zamanda alt birimlerine (subword) ayırarak işler. Örneğin, “güzellik” kelimesini “güz”, “üzel”, “ellik” gibi parçalarına bölerek temsil eder.

Bu sayede:

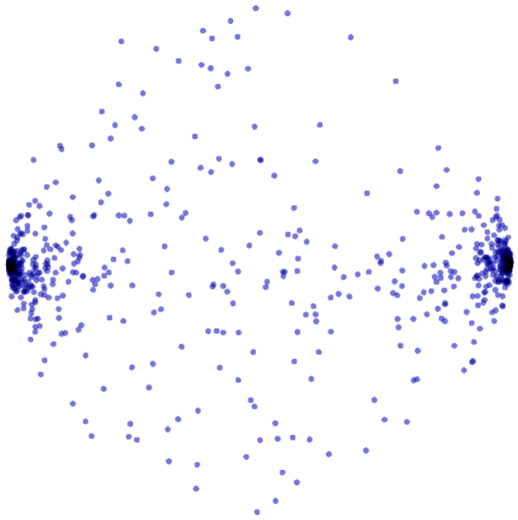
- Nadiren geçen kelimeler için bile vektör oluşturulabilir.
- Aynı kökten türemiş kelimeler benzer vektörlere sahip olur.

Bu yöntemler sayesinde kelimeler, metin içinde anlamlarına uygun şekilde çok boyutlu uzaylarda konumlandırılır ve bu sayede NLP problemlerinde daha başarılı sonuçlar elde edilir.

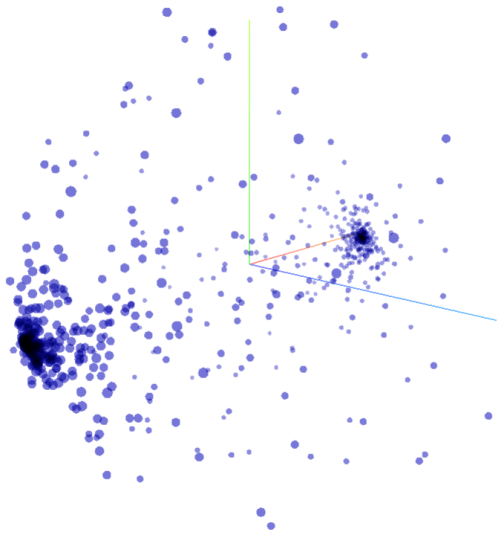
Çok Boyutlu Vektörlerin Görselleştirmesi



Bu görselleştirmede, iki ana küme göze çarpıyor - soldaki daha yoğun ve koyu mavi bir küme ile sağ üstte daha küçük bir küme. Yeşil ve kırmızı/mavi çizgiler muhtemelen temel koordinat sistemini veya boyut projeksiyonlarını temsil ediyor. Veri noktaları farklı yoğunluklarda dağılmış durumda, bazı noktalar merkezden uzaklaşıp saçılırken, çoğunluk iki ana küme etrafında yoğunlaşmış.



Bu görüntüde simetrik bir yapı görülüyor - hem sol hem de sağ tarafta benzer büyüklükte iki yoğun küme var. Bu yapı, verilerin iki ana kategoriye ayrıldığını veya çift modlu bir dağılım gösterdiğini düşündürüyor. Noktaların ortada daha seyrek olması, bu iki küme arasında bir ayrım olduğunu gösteriyor.



Bu görselleştirmede, biri sol üstte diğeri sağ altta olmak üzere iki belirgin küme görünüyor. İlk görüntüdeki gibi, koordinat eksenlerini temsil eden renkli çizgiler (yeşil, kırmızı ve mavi) var. Noktaların dağılımı iki küme arasında bir geçiş yolu oluşturuyor gibi görünüyor, ancak noktalar yine de ana kümelerde yoğunlaşmış durumda.

Bu görselleştirmeler, vektör uzayında dağılmış dilsel veya semantik özelliklerin matematiksel temsilidir. Veri noktalarının belirli bölgelerde kümelenme eğilimi, anlamsal veya fonksiyonel benzerlik ilişkilerini göstermektedir. Sözcüklerin veya kavramların bu tür organizasyonu, dil modelleri ve doğal dil işleme sistemlerinin içsel yapısını ortaya

koymaktadır. İstatistiksel paternler, benzer semantik içeriğe sahip sözcüklerin benzer vektörel konumlarda gruplanmasına yol açar. Bu dağılımlar, kullanılan eğitim verilerindeki dilsel ilişkilerin matematiksel izdüşümüdür. Kümelerin arasındaki boşluklar ise kavramsal farklılıkları temsil eder ve semantik ayrımları belirginleştirir. Bu tür görselleştirmeler, yüksek boyutlu vektör uzayındaki karmaşık ilişkileri anlamak için önemli bir analitik araç sağlamaktadır.

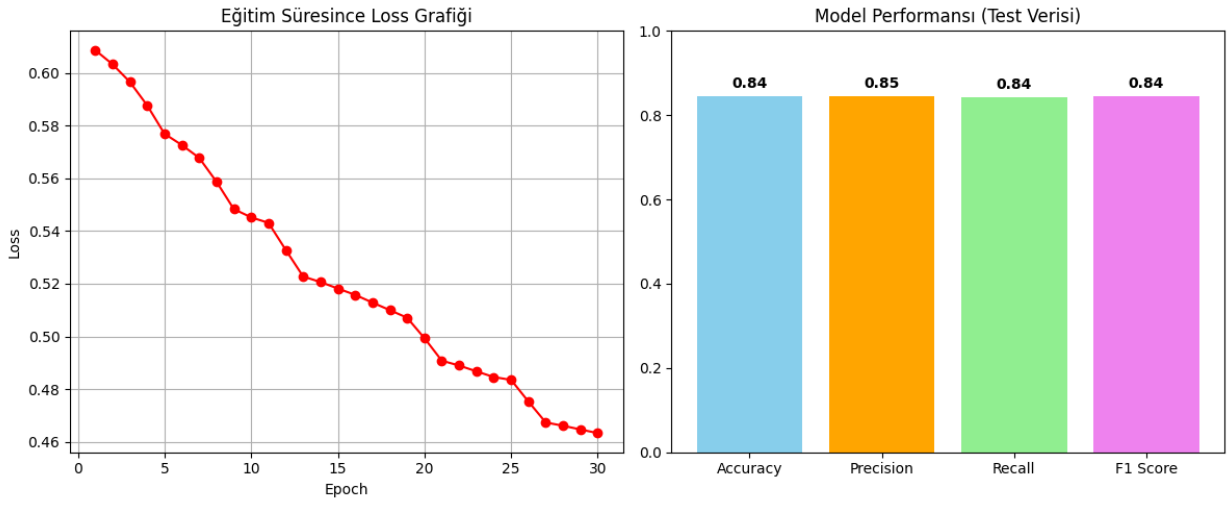
Problem Tanımı ve Elde Edilen Çözümler

Bu çalışmada, doğal dil işleme (NLP) alanındaki temel problemlerden biri olan duygu analizi (sentiment analysis) üzerine odaklanılmıştır. Problem, bir metindeki (örneğin ürün yorumu) ifadenin olumlu mu yoksa olumsuz mu olduğunu otomatik olarak tespit etmeyi amaçlamaktadır. Ödev kapsamında, bu problemi hem word embedding kullanmadan hem de word embedding kullanarak çözmek hedeflenmiştir.

Veri seti olarak, Amazon ürün yorumlarından oluşan etiketli bir veri seti tercih edilmiştir. Bu veri setinde her yorumun karşısında ilgili duygusal sınıf etiketi (pozitif/negatif) yer almaktadır.

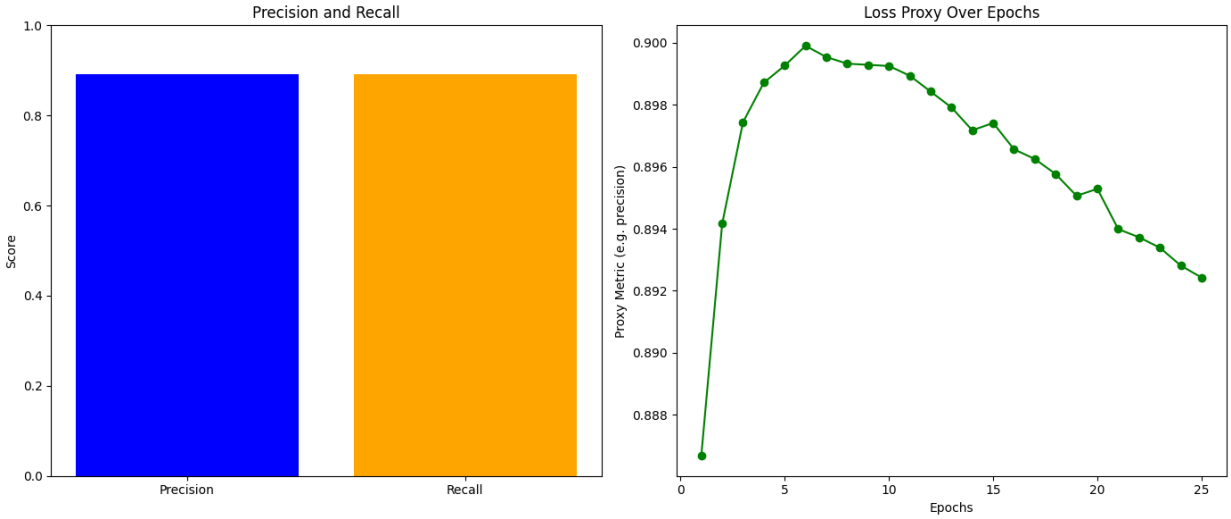
Embedding Kullanılmayan Model

İlk aşamada, yorumlar üzerinde temel metin ön işleme adımları (küçük harfe çevirme, noktalama silme, stopword temizleme vb.) uygulanmış ve ardından TF-IDF gibi geleneksel yöntemlerle kelimeler sayısal vektörlere dönüştürülerek klasik makine öğrenmesi modelleri (Logistic Regression, Naive Bayes) ile sınıflandırma gerçekleştirilmiştir.



Word Embedding Kullanılan Model

İkinci aşamada ise aynı problem, word embedding yöntemlerinden biri olan FastText kullanılarak çözülmüştür. Embedding'li temsil sonrasında, yine benzer sınıflandırma modelleri kullanılarak yorumların duygu etiketleri tahmin edilmiştir.



Değerlendirme

Word embedding **kullanmayan modelde**, loss değerleri eğitim boyunca istikrarlı şekilde düşmüş, bu da modelin her epoch'ta daha iyi hale geldiğini gösteriyor. Ancak bu modele ait test verisi üzerindeki **accuracy, precision, recall ve F1 score** değerleri 0.84–0.85 civarında kalarak bir noktada sınırlı iyileşme sunmuş. Yani, model öğreniyor ama sınırlı bir temsil gücüne sahip.

Öte yandan, **word embedding kullanılan modelde**, başlangıçta daha yüksek bir precision elde edilmiş ve modelin proxy metric değeri (örneğin precision) daha yüksek seviyelerde seyretmiş (~0.899 civarında zirve yapmış). Ancak ilerleyen epoch'larda hafif bir düşüş gözlemlenmiş, bu da modelin fazla öğrenmeye (overfitting) yatkın olabileceğini gösteriyor. Buna rağmen, genel olarak daha iyi başlangıç performansı ve yüksek doğruluk oranları sağladığı görülüyor.

Sonuç olarak, word embedding kullanılan model, daha zengin kelime temsilleri sayesinde daha yüksek performans sergilemiş. Ancak bu modelin dikkatli bir şekilde regularize edilmesi gerekebilir. Embedding kullanılmayan model ise daha stabil ama sınırlı bir başarı düzeyine ulaşmış. Uygulamanın ihtiyacına göre hız ve doğruluk dengesine göre seçim yapılabilir.

Github Linki:

https://github.com/raycure/ODEV4_GRUP6

Kaynakça

- https://projector.tensorflow.org/
- https://medium.com/bili%C5%9Fim-hareketi/word-embedding-teknikleri-word2vec-nedir-tf-idf-nedir-e2f826dd9178
- https://www.turing.com/kb/guide-on-word-embeddings-in-nlp
- Veri seti: https://www.kaggle.com/datasets/bittlingmayer/amazonreviews/data