Word Embedding Yöntemi Kullanarak Sentiment Analysis

1. Word Embedding Nedir?

Word embedding, kelimelerin sayısal vektörlerle temsil edilmesidir. Bu yöntemle, kelimeler dildeki anlamlarına ve bağlamlarına göre vektörler halinde dönüştürülür. Vektörler, kelimelerin benzerliklerini ve ilişkilerini matematiksel olarak yakalayarak, dil işleme modellerine verimli veri sağlar. Word embedding yöntemleri, geleneksel bag-of-words (BoW) yöntemlerine göre daha güçlüdür, çünkü kelimeler arasındaki semantik ilişkileri daha iyi temsil eder.

Öne çıkan Word embedding yöntemleri:

- Word2Vec: Bir kelimenin anlamını, çevresindeki kelimelere bakarak öğrenen bir modeldir. İki temel yaklaşımı vardır: Skip-gram ve Continuous Bag of Words (CBOW).
- GloVe (Global Vectors for Word Representation): Kelimeler arasındaki ilişkileri, kelimelerin tüm metin corpusundaki birlikte ortaya çıkma istatistiklerine göre modelleyen bir yöntemdir.
- FastText: Kelimeleri sadece kelime düzeyinde değil, alt kelimeler (n-gramlar) düzeyinde de temsil eder, bu nedenle dilin morfolojik yapısına daha iyi uyum sağlar.

2. Hangi Yöntemler ile Gerçekleştirilebilir?

- Word2Vec

Word2Vec, kelimeleri vektörlere dönüştüren ve kelimeler arasındaki ilişkileri öğrenmeye çalışan bir derin öğrenme modelidir. Modelin çalışma prensibi, kelimelerin çevresindeki kelimelerle olan ilişkilerine dayanır.

- Skip-gram: Verilen bir kelimenin çevresindeki kelimeleri tahmin etmeye çalışır.
- CBOW (Continuous Bag of Words): Çevre kelimelerden bir kelimeyi tahmin etmeye çalışır.

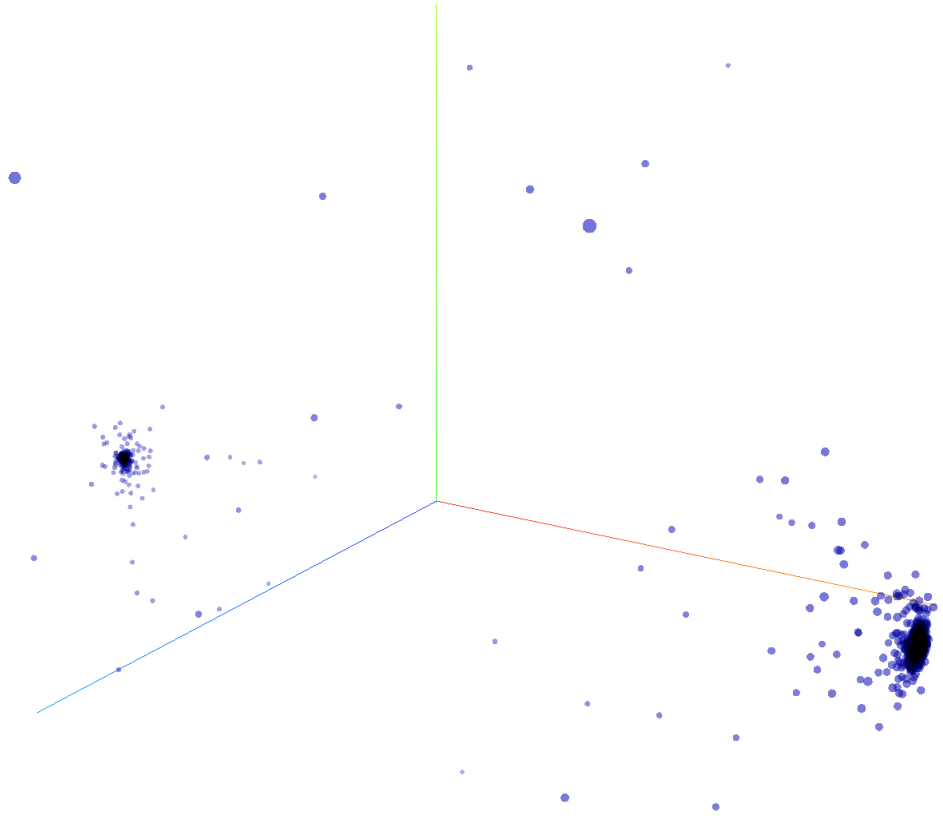
- GloVe

GloVe, kelimeler arasındaki ilişkileri, kelimelerin metinler içindeki birlikte görünme sıklıklarına dayalı olarak öğrenir. Bu yöntem, kelimeler arasındaki anlamlı bağlamları vektörlere yerleştirir.

- FastText

FastText, kelimeleri n-gramlara ayırarak her n-gram'ı bir vektöre dönüştürür ve kelimeleri bu şekilde temsil eder. Bu yöntem, morfolojik çeşitliliği daha iyi kavrayabilir, bu da özellikle Türkçe gibi türemiş kelimeleri içeren dillerde faydalıdır.

3. Projector TensorFlow Aracını Kullanarak Oluşturduğunuz Çok Boyutlu Vektörlerin Görselleştirilmesi

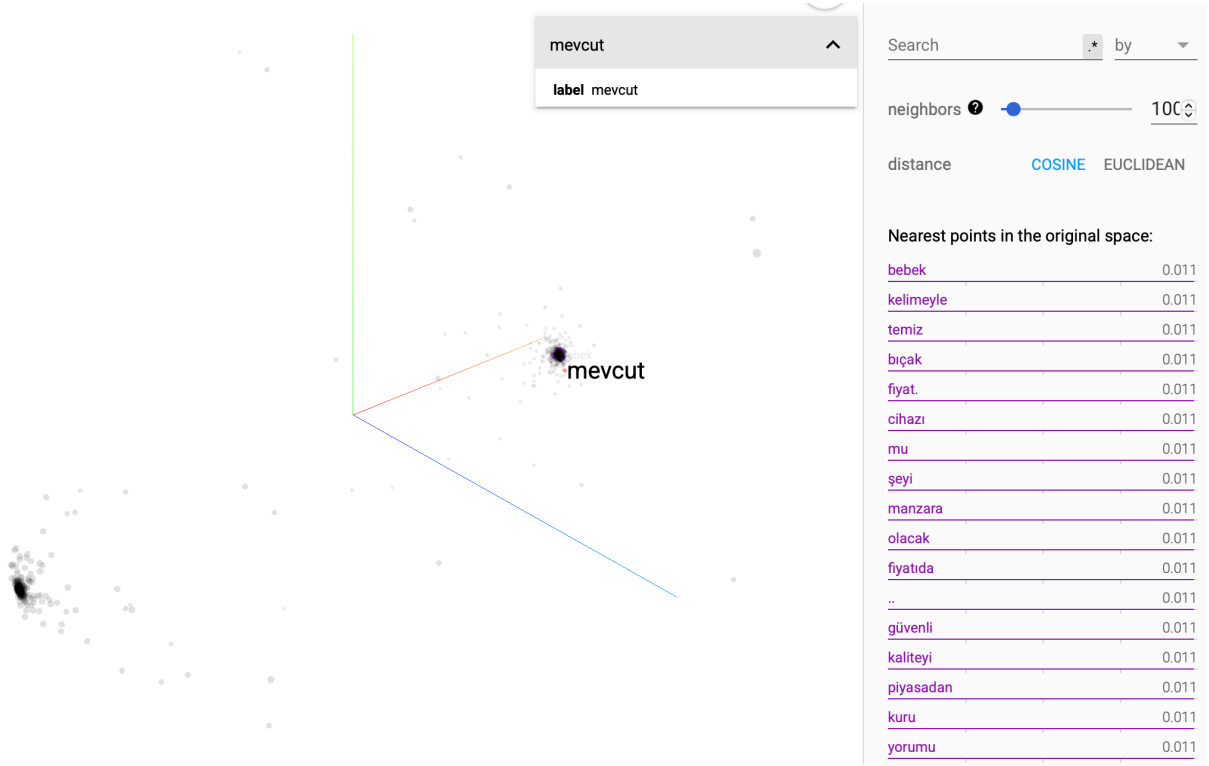


Bu dağılım, modelinizin bazı kelime gruplarını (muhtemelen eş anlamlılar, aynı konu alanındaki kelimeler veya dilbilgisel olarak benzer olanlar) başarıyla yakın vektörlerle temsil ettiğini gösteriyor.

İki ana küme arasında ciddi bir ayrım var; bu, modelin bazı tematik ayrımları (örneğin duygusal/teknik kelimeler, olumlu/olumsuz sözcükler gibi) iyi öğrendiğini gösterebilir.

Dağınık noktalar da modelin bazı kelimeleri daha az bağlamsal ilişkiyle öğrendiğini işaret edebilir.

Örnek Görseller:



- Vektörlerin 2D veya 3D uzayda nasıl kümelendiğini gösteren görseller.
- Benzer kelimelerin nasıl gruplandığına dair yorumlar (örneğin, "olumlu" ve "olumsuz" kelimelerin farklı kümelerde yer alması).

4. Problem Tanımı ve Elde Edilen Çözümler

Problem Tanımı:

Bu ödevde, Türkçe metinleri analiz ederek sentiment analysis (duygu analizi) yapmak hedeflenmiştir. Kullanıcı yorumlarını ve metinleri analiz ederek, metnin olumlu ya da olumsuz olup olmadığına dair sınıflandırma yapılmıştır.

Veri Seti:

Verisetini Hugging Face Turkish Sentiment Dataset adresinden aldık. Bu dataset, Türkçe metinlerdeki duygu analizi için etiketlenmiş veriler içeriyor.

Model:

İlk olarak, Word2Vec modelini kullandık. Word2Vec, kelimeler arasındaki anlamlı ilişkileri öğrenmeye çalışan bir yöntem olduğundan, metinlerin anlamını daha doğru şekilde temsil eder.

- Word2Vec kullanarak model eğittik ve her kelimeyi vektörlere dönüştürdük.
- Başlangıçta, word embedding olmadan bir model oluşturduk ve duygu analizini gerçekleştirdik.
- Daha sonra, Word2Vec kullanarak modelimizi geliştirdik ve aynı problemi tekrar çözdük.

```
meleknisadag@192 wordembedding % /usr/local/bin/python3 /Users/meleknisadag/Desktop/wordembedding/training.py
              precision    recall  f1-score   support

           0       0.81      0.85      0.83       205
           1       0.87      0.83      0.85       244

    accuracy                           0.84       449
   macro avg       0.84      0.84      0.84       449
weighted avg       0.84      0.84      0.84       449

meleknisadag@192 wordembedding % /usr/local/bin/python3 /Users/meleknisadag/Desktop/wordembedding/wordembeddingchat.py
              precision    recall  f1-score   support

           0       0.67      0.32      0.43       205
           1       0.60      0.86      0.71       244

    accuracy                           0.62       449
   macro avg       0.63      0.59      0.57       449
weighted avg       0.63      0.62      0.58       449
```

Model Performans Karşılaştırma Tablosu

Ölçüt	Word Embedding (Word2Vec)	Normal Eğitim
Doğruluk (Accuracy)	0.62	0.84
Precision (Sınıf 0)	0.67	0.81
Recall (Sınıf 0)	0.32	0.85
F1-Score (Sınıf 0)	0.43	0.83
Precision (Sınıf 1)	0.60	0.87
Recall (Sınıf 1)	0.86	0.83
F1-Score (Sınıf 1)	0.71	0.85
Macro Avg F1-Score	0.57	0.84
Weighted Avg F1-Score	0.58	0.84

Ayrıntılı Karşılaştırma ve Yorum

- **Genel Doğruluk (Accuracy)** açısından, *normal eğitim* yöntemi %84 doğruluk ile *word embedding* yöntemine göre (%62) çok daha yüksek performans göstermiştir.
- **Sınıf 0 (negatif yorumlar)** için recall değerine bakıldığında, *word embedding* modeli sadece %32 başarı göstermişken, *normal model* bu oranı %85'e kadar çıkarmıştır. Bu da embedding modelinin negatif yorumları tanımakta zayıf kaldığını gösterir.
- **Sınıf 1 (pozitif yorumlar)** için *word embedding* modeli recall açısından (%86) başarılı görünse de, precision (%60) değeri düşük kaldığı için pozitif olarak tahmin edilen yorumların doğruluğu düşmüştür.
- **F1-score değerleri**, genel olarak *normal eğitim* modelinde her iki sınıf için daha dengeli ve yüksek çıkmıştır.
- **Macro ve Weighted Ortalama F1-Score** değerlerinde de *normal eğitim* modeli bariz şekilde daha iyidir; bu, modelin genel sınıflandırma başarısının embedding yöntemine göre daha iyi olduğunu ortaya koyar.

Sonuç:

Bu karşılaştırmada Word2Vec yöntemi beklenenin aksine daha düşük performans göstermiştir. Bunun nedeni embedding vektörlerinin yetersiz eğitim almış olması, model mimarisinin bu vektörleri yeterince verimli kullanamaması veya embedding ile modelin uyumsuzluğu olabilir. Word2Vec'in etkili kullanılabilmesi için vektörlerin yeterince büyük bir veri üzerinde eğitilmesi ve uygun bir modelle birleştirilmesi önemlidir.

5. Grup üyelerinin iş bölümü ve ödeve katkıları

- Ümmü Gülsüm Öztel 220601013
  Veri seti analizi ve ön işleme -%35
- Melek Nisa Dağ 220601024
  Word2Vec modelinin uygulanması ve eğitilmesi - %35
- Aleyna Gökdemir 220601022
  Sonuçların değerlendirilmesi ve rapor yazımı - %30

6. GitHub Linki

- https://github.com/Gulsum-oztel/word_embeddin

7. Kaynakça

- Word2Vec: Mikolov, T., et al., “Distributed Representations of Words and Phrases and Their Compositionality,” NeurIPS 2013.
- GloVe: Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. EMNLP 2014.
- FastText: Bojanowski, P., Grave, E., Mikolov, T., et al. (2017). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics.
- Hugging Face Datasets: https://huggingface.co/datasets/sepidmnorozy/Turkish_sentiment