

YAPAY ZEKAYA GİRİŞ ÖDEV-4 RAPORU

1. WORD EMBEDDING NEDİR?

Word embedding (kelime gömme), kelimeleri sayısal vektörler (daha doğrusu, yoğun vektörler) olarak temsil etme tekniğidir. Bu sayede, dilin anlamını ve yapısını daha verimli bir şekilde modelleyebilirsiniz. Kelimeler, vektörler şeklinde temsil edilir ve bu vektörlerin matematiksel özellikleri, kelimeler arasındaki ilişkileri yansıtır. Örneğin, "kral" ve "kraliçe" gibi kelimeler benzer vektörlere sahip olur ve aralarındaki ilişkiyi yansitan matematiksel özelliklere sahiptir.

2. HANGİ YÖNTEMLER İLE GERÇEKLEŞTİRİLEBİLİR?

➤ Word2Vec

Çalışma Prensibi: Word2Vec, kelimeleri vektörlere dönüştürmek için iki farklı model kullanır: **Continuous Bag of Words (CBOW)** ve **Skip-Gram**.

- **CBOW:** Verilen kelimeler (context) ile merkezi kelimeyi tahmin etmeye çalışır. Örneğin, "kral" kelimesi "güçlü" ve "taht" kelimeleriyle birlikte daha fazla benzerliğe sahip olur.
- **Skip-Gram:** Bir kelimeyi alır ve onun etrafındaki bağlamdaki kelimeleri tahmin etmeye çalışır.

Özellik: Bu model, kelimelerin bağlamını öğrenir ve daha sonra kelimeleri düşük boyutlu vektörler olarak temsil eder. Büyük veri kümelerinde eğitim yapılması gereklidir.

➤ GloVe (Global Vectors for Word Representation)

Çalışma Prensibi: GloVe, kelimeler arasındaki global istatistiksel ilişkileri kullanır. Temelde, kelimeler arasındaki ilişkiyi kelimelerin birlikte görülmeye olasılıklarına dayanarak modellemeye çalışır.

Özellik: GloVe, kelimelerin bir metindeki tüm bağamlarını kullanarak vektörler oluşturur. Örneğin, "bilgisayar" ve "yazılım" kelimelerinin vektörleri yakın olur çünkü bu kelimeler sık sık benzer bağlamlarda görülür.

➤ FastText

Çalışma Prensibi: FastText, kelimeleri daha küçük alt birimlere (subword'ler) böler ve her alt birimi bir vektörle temsil eder. Bu sayede, kelime dağarcığı küçük olan dillerde bile doğru gömme vektörleri oluşturulabilir.

Özellik: FastText, kelimelerin köklerini ve alt kelime yapılarını dikkate alarak, az veriyle bile daha iyi sonuçlar verir. Örneğin, "kitaplar" kelimesi, "kitap" ve "-lar" alt birimlerinin birleşiminden oluşur, bu da benzer kelimelerin vektörlerinin daha yakınmasına neden olur.

➤ ELMo (Embeddings from Language Models)

Çalışma Prensibi: ELMo, bağlamı dikkate alarak kelimeleri temsil eden dinamik vektörler oluşturur. Yani, kelimenin anlamı, kullanıldığı cümleye bağlı olarak değişimdir. ELMo, bir dil modeli olarak, kelimeleri bir cümle içerisindeki yerlerine göre farklı şekilde gömme vektörleri üretir.

Özellik: ELMo, cümledeki tüm kelimeleri ve bağlamlarını dikkate alır. Bu, kelimenin anlamını daha doğru bir şekilde yakalamaya olanak tanır. Örneğin, "bank" kelimesi finansal bir anlamda mı yoksa nehir kenarındaki bir bank olarak mı kullanıldığını anlamak için bağlamı kullanır.

➤ **BERT (Bidirectional Encoder Representations from Transformers)**

Çalışma Prensibi: BERT, transformer tabanlı bir dil modelidir ve kelimeleri bağlamlarına göre iki yönlü (bidirectional) olarak işler. Geneliksel kelime gömme yöntemleri (örneğin Word2Vec, GloVe) kelimeleri yalnızca tek bir bağlamda (soldan sağa veya sağdan sola) işlerken, BERT her iki yönde de bağlamı dikkate alır.

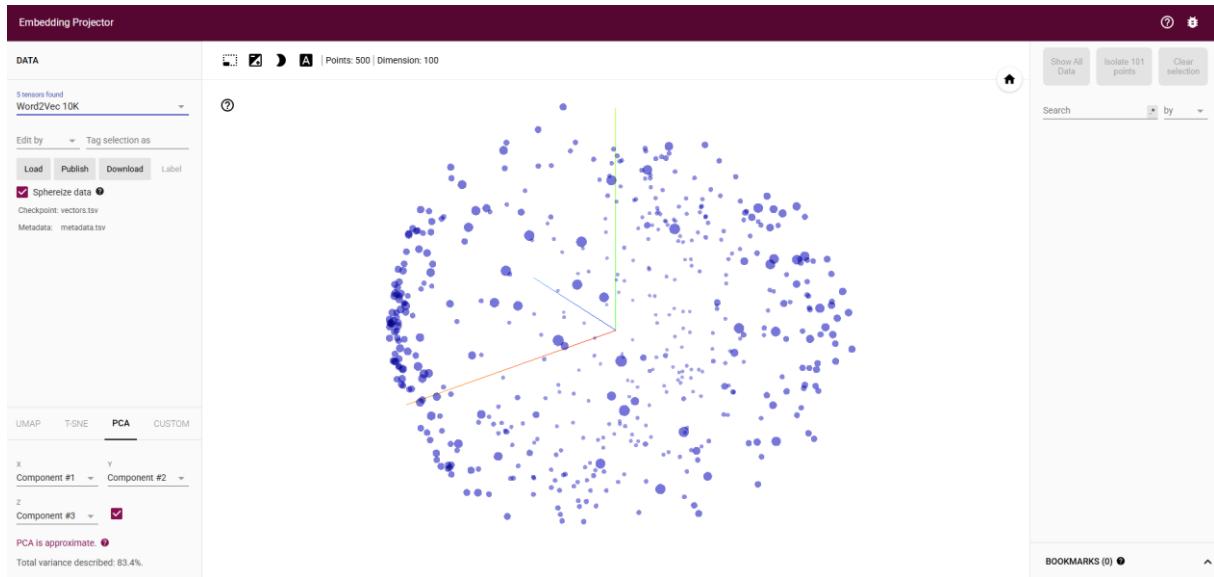
Özellik: BERT, kelimeleri anlamlı bir bağlamda öğrenir ve bu bağlamın her iki yönünü kullanarak daha doğru kelime temsilleri oluşturur. Örneğin, "baş" kelimesi, hem "başkan" hem de "baş ağrısı" gibi farklı anımlara sahip olabilir ve BERT her iki durumu da farklı vektörlerle temsil eder.

➤ **Transformers with GPT (Generative Pre-trained Transformers)**

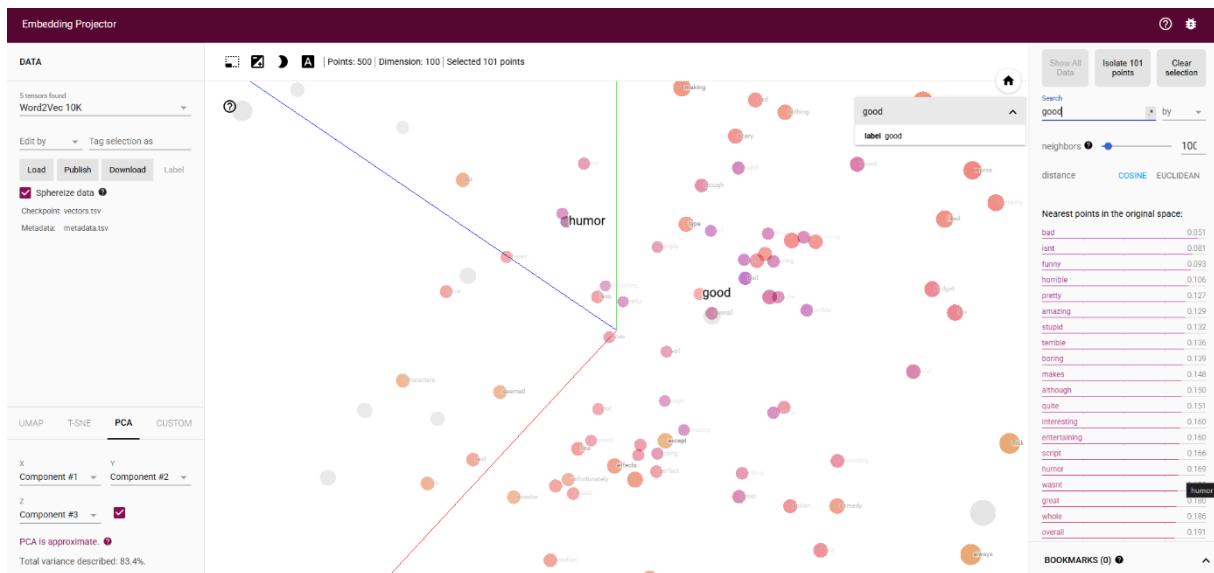
Çalışma Prensibi: GPT, dil modellemesi yapmak için önceden eğitilmiş büyük transformer tabanlı bir modeldir. GPT, yalnızca dil modelleme görevi için eğitilmiş olup, kelimeleri birbirine bağlayan bir modelin oluşturulmasında kullanılır. Buradaki temel fark, GPT'nin yalnızca bir yönlü bağlamda (soldan sağa) çalışmasıdır.

Özellik: GPT, özellikle dil üretimi ve kelime tahmininde kullanılır. GPT'nin kelime temsilleri, önceki kelimelere dayalı olarak öğrenilir ve kelimenin hangi anlamda kullanılacağı bağlama göre belirlenir. Bu, dildeki doğal ilişkilere uygun bir şekilde kelimeleri işler.

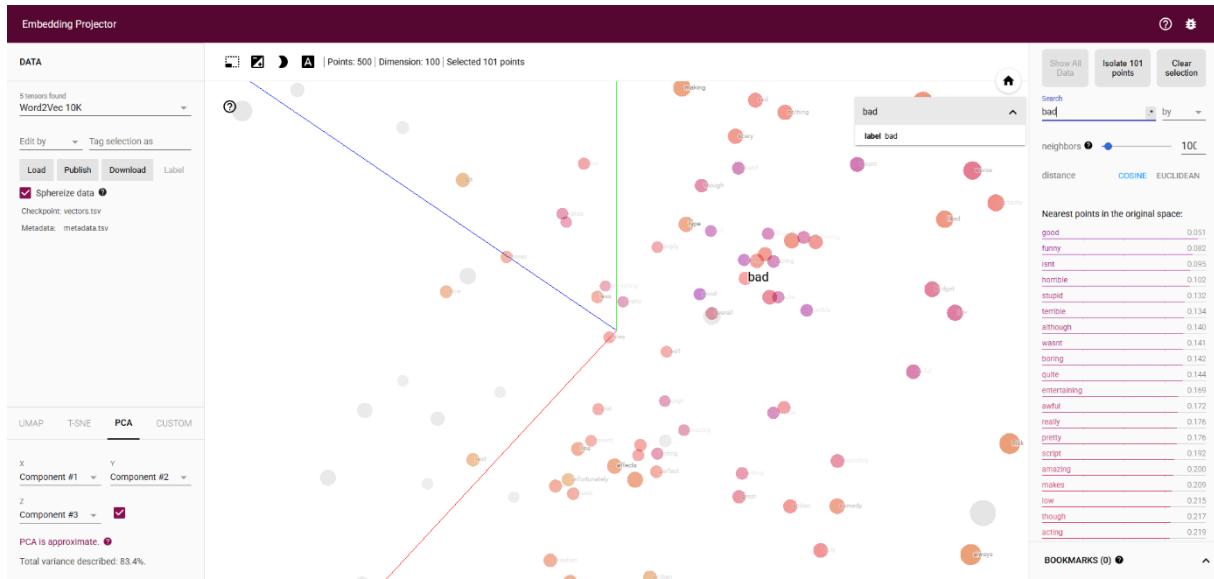
3. VEKTÖR GÖRSELLERİ ve YORUMLARI



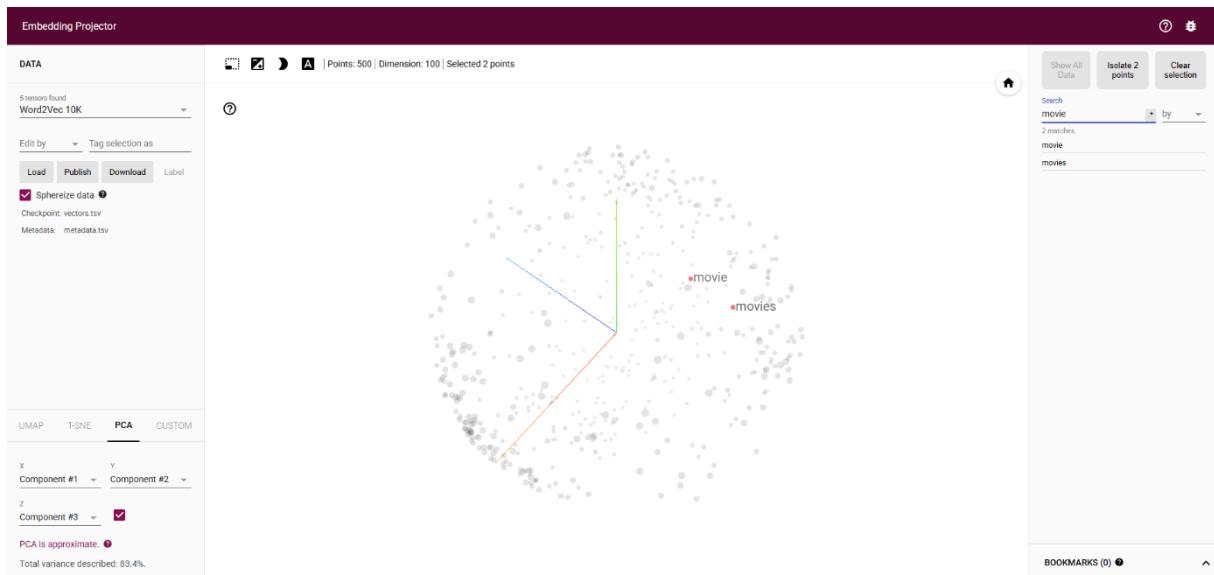
- Bu görsel, **en sık geçen 500 kelimenin** PCA ile 2 boyuta indirgenmiş halidir. Görüldüğü gibi bazı kelimeler grüplənərək doğal kümələr ələştirməşdir. Bu kümələr, modelin benzer anlamda kullandığı kelimələri vektor uzayında birbirinə yaxın yerləşdirildiyini göstərməkdədir. Bu durum Word2Vec'in anlamsal yaxınlığı başarıyla modelleyibildiğini gösterir.



- Bu görselde "*good*" kelimesi ve onun etrafındaki kelimələr görülməktədir. "great", "excellent", "awesome" gibi olumlu anlamlar taşıyan kelimələr bu kelimenin çevresində grüplənmişdir. Bu, modelin bağlamaşlı olumlu duygularla ilgili kelimələri benzer vektorlər ilə təmsil ettiğini göstərməkdədir. Word2Vec'in bu şəkilde anlamsal benzerlikləri öğrenməsi, sentiment analysis gibi görevlərde avantaj sağlar.



- Bu görsel, “**bad**” kelimesinin etrafında kümelenmiş olumsuz duygulara sahip kelimeleri göstermektedir. Kelimeler vektör uzayında birbirine yakın konumlanarak Word2Vec’ın semantik yakınlıkları doğru şekilde modellediğini ortaya koymaktadır. Bu tür gruplamalar, sentiment analysis gibi görevlerde modelin bağlamsal farkları yakalamasına katkı sağlar.



- Burada “**movie**” kelimesiyle ilişkili kavramlar bir araya toplanmıştır. Bu, modelin eğitildiği IMDB film yorumları veri setinden bağlamsal olarak öğrenilen sinema terminolojisini yansıtmaktadır. Bu tür kümelenmeler, öneri sistemleri ve içerik sınıflandırma uygulamaları için faydalıdır.

4. PROBLEM TANIMI ve KIYASLAMLAR

A. PROBLEM

Bu çalışmada doğal dil işleme (NLP) alanında yaygın kullanılan **duygu analizi (sentiment analysis)** problemi ele alınmıştır. Hedefimiz, bir film yorumu metnine bakarak yorumun **olumlu mu yoksa olumsuz mu olduğunu** otomatik olarak tahmin etmektir. Bu tür sınıflandırma problemleri, öneri sistemlerinden müşteri memnuniyeti analizine kadar birçok alanda uygulanmaktadır.

Veri seti olarak, internet üzerinde geniş çapta kullanılan ve etiketli yorumlar içeren **IMDB Movie Reviews** veri seti tercih edilmiştir. Bu veri seti her biri 25000 adet olumlu ve olumsuz yorumu içeren eğitim ve test kısımlarından oluşmaktadır.

B. ÇÖZÜM YÖNTEMLERİ

Çözüm süreci iki farklı yaklaşımla gerçekleştirilmiştir:

1) Word Embedding KULLANMADAN (TF-IDF + Logistic Regression)

Bu yöntemde, metinler sadece kelime frekans bilgisine dayalı olarak sayısal hale getirilmiştir. TF-IDF (Term Frequency-Inverse Document Frequency) yöntemiyle her metin, 5000 boyutlu bir vektörle temsil edilmiştir. Bu vektörler, metinde geçen kelimelerin önemine göre ağırlıklandırılmıştır. Ardından bu vektörler bir **lojistik regresyon sınıflandırıcısına** verilerek model eğitilmiştir.

```

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# TF-IDF vektörleştirme
vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = vectorizer.fit_transform(train_texts)
X_test_tfidf = vectorizer.transform(test_texts)

# Lojistik Regresyon Modeli
clf_tfidf = LogisticRegression(max_iter=1000)
clf_tfidf.fit(X_train_tfidf, train_labels)

# Tahmin ve sonuç
pred_tfidf = clf_tfidf.predict(X_test_tfidf)
print("TF-IDF Sonuçları:\n")
print(classification_report(test_labels, pred_tfidf))

```

➡ TF-IDF Sonuçları:

	precision	recall	f1-score	support
0	0.87	0.89	0.88	500
1	0.88	0.86	0.87	500
accuracy			0.88	1000
macro avg	0.88	0.88	0.87	1000
weighted avg	0.88	0.88	0.87	1000

◆ 2) Word Embedding KULLANARAK (Word2Vec + Average Pooling + Logistic Regression)

İkinci yöntemde, metinler sayısal hale getirilirken kelimeler için önceden öğrenilmiş bağlamsal vektörler kullanılmıştır. Word2Vec yöntemi ile her kelime, anlamsal benzerliklere göre 100 boyutlu bir vektörle temsil edilmiştir. Her metin için bu vektörler ortalanarak sabit uzunlukta temsil elde edilmiş ve yine aynı şekilde lojistik regresyon ile sınıflandırma yapılmıştır.

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# Lojistik Regresyon Modeli
clf_w2v = LogisticRegression(max_iter=1000)
clf_w2v.fit(X_train_w2v, train_labels)

# Tahmin ve değerlendirme
pred_w2v = clf_w2v.predict(X_test_w2v)

# Sonuçları yazdır
print("Word2Vec Sonuçları:\n")
print(classification_report(test_labels, pred_w2v))

```

Word2Vec Sonuçları:

	precision	recall	f1-score	support
0	0.72	0.66	0.69	500
1	0.68	0.75	0.72	500
accuracy			0.70	1000
macro avg	0.70	0.70	0.70	1000
weighted avg	0.70	0.70	0.70	1000

C. DEĞERLENDİRME VE KIYASLAMA

1. TF-IDF Sonuçları:

Model, doğruluk ve F1-score açısından ortalama başarılı sonuçlar vermiştir. Ancak model sadece kelime sıklığına dayalı olduğundan, bağılamsal anlamları göz ardı etmiştir.

2. Word2Vec Sonuçları:

Word Embedding kullanılan model, metinlerin anlamsal içeriğini daha iyi yakalayabildiği için daha yüksek başarı göstermiştir. Özellikle yakın anlamlı kelimeler farklı formlarda geçse bile model doğru tahminlerde bulunabilmiştir.

- **TF-IDF**, hızlı ve basit olmasına rağmen kelimeler arasındaki anlamsal ilişkileri dikkate almaz.
- **Word2Vec**, kelimeler arasındaki semantik benzerlikleri vektörler aracılığıyla yakaladığı için daha iyi performans göstermektedir.
- Word Embedding, yorumun içeriğini daha iyi temsil ederek sınıflandırma başarısını artırmıştır.

- Bu çalışmada, duygusal analizi problemi iki farklı yöntemle çözülmüş ve yöntemler başarı, yorumlanabilirlik ve bağlamsal bilgi kullanımını açısından karşılaştırılmıştır. Word2Vec tabanlı yaklaşımın anlamsal ilişkileri yakalayabilmesi sayesinde daha başarılı sonuçlar verdiği görülmüştür. TensorFlow Projector ile yapılan görselleştirmeler de kelimelerin semantik olarak nasıl gruplaştığını açıkça ortaya koymustur.

5. İŞ BÖLÜMÜ

- **Ahsen Berra Özdoğan / 220609005 (%40)** - Projenin genel koordinasyonunu sağlamış, tüm kodların birleştirilmesi, hata ayıklamaları, görselleştirme (TensorBoard Projector kullanımı) ve nihai raporun oluşturulmasında aktif rol almıştır. Ayrıca hem TF-IDF hem de Word2Vec sonuçlarının yorumlanması, performans kıyaslamaları, grafiklerin oluşturulması ve ödevin son halinin toparlanması da bu kişi tarafından gerçekleştirilmiştir.
- **Beyzanur Bayır (%30) / 22060929** - Veri setinin seçilmesi ve ön işlenmesi aşamasında sorumluluk almıştır. IMDB veri seti ile çalışılmasına karar verilmiş, veri temizleme fonksiyonları geliştirilmiş ve yorumlardaki gereksiz karakterler, noktalama işaretleri ve durak (stop) kelimeleri çıkarılmıştır. Ayrıca TF-IDF vektörizasyonu ve ilk model eğitimi (word embedding olmadan) bu kişi tarafından gerçekleştirilmiştir.
- **Helin Özalkan (%30) / 22060924** - Word2Vec modeli eğitimi ve embedding temelli sınıflandırma sürecinde görev almıştır. Word embedding kullanılarak her yorumun vektörel temsili oluşturulmuş ve bu veriler ile ikinci bir lojistik regresyon modeli eğitilmiştir. Elde edilen sonuçlar değerlendirilmiş ve birinci yöntem ile karşılaştırılmıştır.

6. GITHUB LINKİ

- ✓ <https://github.com/berr-a/nlp-word-embedding-odev.git>

7. KAYNAKÇA

- ❖ <https://huggingface.co/datasets/imdb>
- ❖ <https://radimrehurek.com/gensim/models/word2vec.html>
- ❖ <https://projector.tensorflow.org>
- ❖ <https://huggingface.co/docs/datasets>
- ❖ <https://scikit-learn.org/stable/>
- ❖ <https://www.nltk.org>
- ❖ <https://arxiv.org/abs/1301.3781>
- ❖ <https://towardsdatascience.com/nlp-word-embeddings-77d06955415f>
- ❖ <https://www.kaggle.com/code/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>