

Doğal Dil İşleme

(Natural Language Processing)

Muhammed Ensar Çıtak

Doğal Dil İşleme

Doğal Dil İşleme, insanların kendi aralarında anlaşmak için kullandıkları doğal dilin insan-bilgisayar etkileşimini en üst düzeye çıkarmak amacıyla işlenmesi ve kullanılması hakkında araştırma yapan bilim dalıdır.

Bilgisayarların doğal diller üzerinde işlemler yapabilmesi için doğal dillerin çeşitli aşamalardan geçirilerek makinelerin anlayacağı şekilde işlenmesi gerekir.



Made with Gamma

Metin İşleme Aşamaları

Tokenization, Stop Words Removal

Tokenization metin içerisindeki kelimelerin, cümlelerin ayrıştırılması sürecini ifade eder.

Stop-words removal ise dilde yaygın olarak kullanılan ve tek başına anlam ifade etmeyen kelimelerin (ile, ve, şu) metinden çıkarılmasıdır.

Vectorization

Doğal Dil İşleme çalışmalarında modeller her zaman vektörler ve matrisler ile çalışmaktadır. Bu da bizim dış dünyadan aldığımız girdileri, bir şekilde bu formata uygun bir hale dönüştürmemizi zorunlu kılmaktadır. Vektörizasyon yöntemleri de bu noktada devreye girer ve girdilerimizi makinenin anlayacağı formata çevirmeye çalışır.

1

Ön İşleme

Özel durumlar dışında, metin tabanlı olmayan veriler yani noktalama işaretleri, özel karakterler dokümandan kaldırılır ve bütün harfler küçük harfe dönüştürülür.

2

Morfolojik Kök Bulma (Lemmatization)

Lemma, kelimenin morfolojik kökü demektir. Örneğin alınan kelimesinin lemması almaktır. Yani lemma için kelimenin yalın halinin sözlükteki karşılığı diyebiliriz.

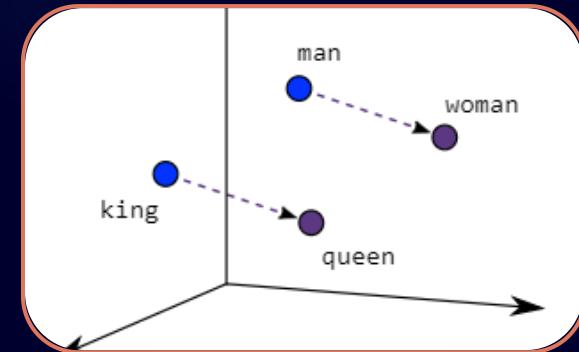
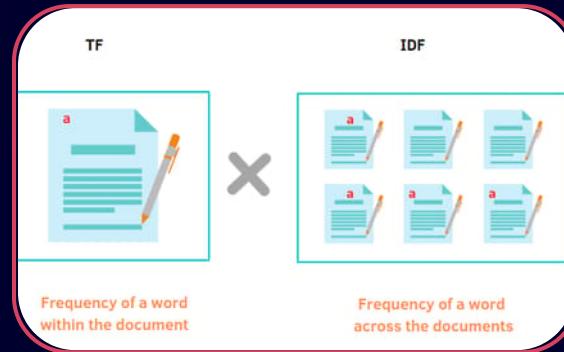
3

4



Vektörizasyon Yöntemleri

	"Bug"	"Poison"	"Ghost"	"Water"	"Electric"	"Ground"
0	0	1	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1
6	0	0	0	1	0	0
7	1	0	0	0	0	0



One-Hot Encoding

Bu yöntemde sözel ifadelerin toplam sayısı boyutunda vektör oluşturulur. Vektörlerin değeri, her bir sözel ifadenin kendi indeksine ait olan değeri 1 diğerleri 0 olacak şekilde atanır.

TF-IDF

TF-IDF, bir terimin bir doküman içerisindeki önemini gösteren istatistikî yöntem ile hesaplanmış bir ölçüdür.

Vektörizasyon işlemi One Hot Encoding'e benzer şekilde gerçekleşir. Farklı olarak sözel ifadeye karşılık gelen değere 1 yerine TF-IDF değeri atanır.

Word2Vec

Word2Vec , kelimeleri vektör uzayında ifade etmeye çalışan tahmin temelli bir **vektörizasyon modelidir**.

Word2Vec vektörleştirme işleminden önce kendisine verilen dokümanı tarayarak hedef kelimenin daha çok hangi kelimelerle birlikte geçtiğini tespit eder. Bu sayede kelimelerin anlamsal olarak birbirleri ile yakınlıklarını kaybetmemiş olur. Her kelimenin, kendisine anlamsal olarak en yakın kelimeleri içeren bir vektörü vardır.

Diger yöntemlerden farklı olarak vektör boyutu doküman içerisindeki eşsiz kelimelerin sayısı kadar olmaz.

TF-IDF (Term Frequency-Inverse Document Frequency)

Document 1		Document 2	
Term	Count	Term	Count
This	1	This	1
is	1	is	2
about	2	about	1
Messi	4	Tf-idf	1

TF (Terim Sıklığı) : Sözcüğün belge içerisinde geçme sıklığıdır.

TF = (t teriminin bir belgede geçme sayısı)/(Belgedeki terim sayısı)

IDF (Ters Belge Frekansı): Bir kelimenin nadir olup olmadığını belirten bir terimdir. Eğer bir kelime diğer belgelerde daha az sıklıkla kullanılıyorsa, o kelimenin IDF değeri yüksek olur. Bu da kelimenin o belgedeki önemini artırır.

IDF = $\log(N/n)$, burada, N belge sayısıdır ve n, t teriminin geçtiği belge sayısıdır.

Yani $IDF(\text{This}) = \log(2/2) = 0$ ve $IDF(\text{Messi}) = \log(2/1) = 0.301$

Buradan Messi kelimesinin dokğanımız için this kelimesinden daha önemli bir kelime olduğu sonucunu çıkarabiliriz.

Word2Vec



Word2Vec modeli CBOW ve Skip-Gram adlarında 2 farklı algoritma ile çalışır.

CBOW ve Skip-Gram modelleri birbirlerinden output'u ve input'u alma açısından farklılaşır. CBOW modelinde **window size**'ın merkezinde olmayan kelimeler input olarak alınıp, merkezinde olan kelimeler output olarak tahmin edilmeye çalışırken; Skip-Gram modelinde ise merkezdeki kelime input olarak alınıp merkezde olmayan kelimeler output olarak tahmin edilmeye çalışılıyor. Bu işlem cümle bitene kadar devam ediyor. Bir cümleye uygulanan bu işlemler tüm cümlelere uygulanıyor.

BERT (Bidirectional Encoder Representations from Transformers)

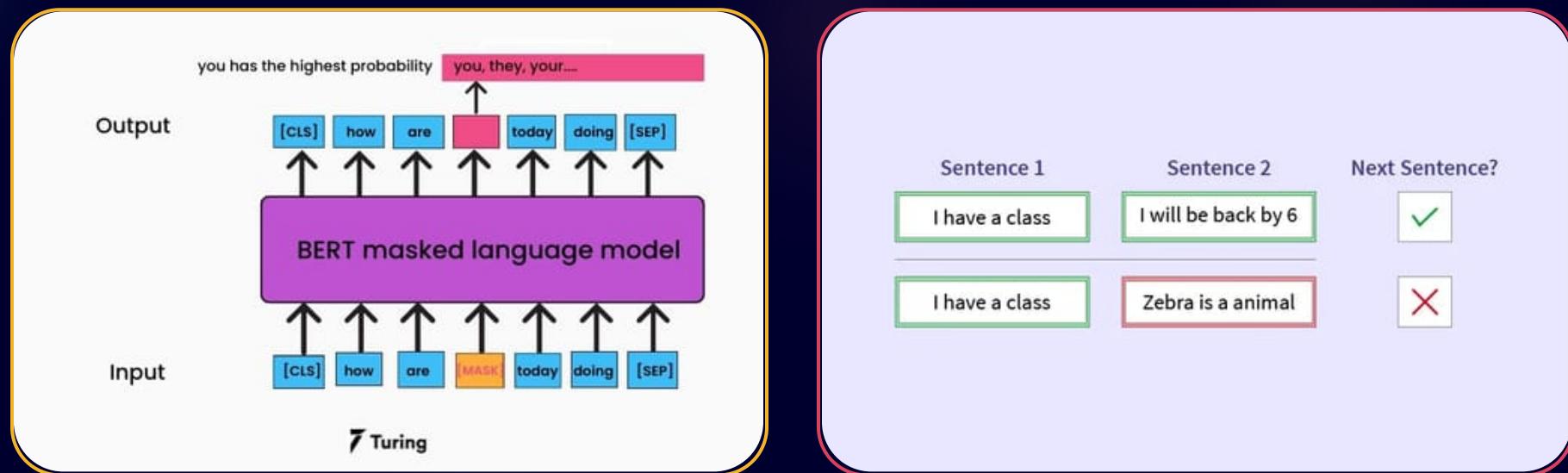


BERT, **Google tarafından** arama motoruna yazılan soruları daha iyi anlamak ve kullanıcılaraya daha doğru sonuçlar sunmak adına geliştirilmiştir. BERT algoritması, yapay zeka ve makine öğrenimi teknolojilerini bir arada kullanan bir **doğal dil işleme** tekniğidir.

- **800M kelime** hazinesine sahip olan **BookCorpus** ve **2.5B kelime** hazinesine sahip olan **Wikipedia** veriseti kullanılarak **bert_large** ve **bert_base** adı verilen 2 temel model geliştirilmiştir.

BERT Modeli Oluşturulurken Kullanılan Teknikler

BERT, Masked Language Modeling (MLM) ve Next Sentence Prediction (NSP) adı verilen iki teknikle eğitilmiştir. Bir cümle modele girdiğinde, cümledeki kelimelerin %15'inde MLM tekniği kullanılmıştır. Bu tekniğin kullanıldığı kelimelerin %80'i [MASK] token'ı ile, %10'u rastgele başka bir kelimeyle değiştirilmiş, geri kalan %10 da değiştirilmeden bırakılmıştır.



Masked Language Modeling tekniğinde, maskelenen kelime, açık şekilde beslenen kelimelerle tahmin edilmeye çalışılır. (Bu teknikte sadece maskelenen kelimeler tahmin edilmeye çalışılır, açık olan veya üzerinde işlem uygulanmayan kelimelerle ilgili herhangi bir tahmin gerçekleştirilmez. Bu sebeple Loss değeri sadece işlem uygulanan kelimeler üzerinden değerlendirilir).

Next Sentence Prediction tekniğinde ise ikili olarak gelen cümle çiftlerinde, ikinci çümlenin ilk çümlenin devamı olup olmadığı tahmin edilir. Bu teknikten önce ikinci cümlelerin %50'si rastgele değiştirilir, %50'si ise aynı şekilde bırakılır.