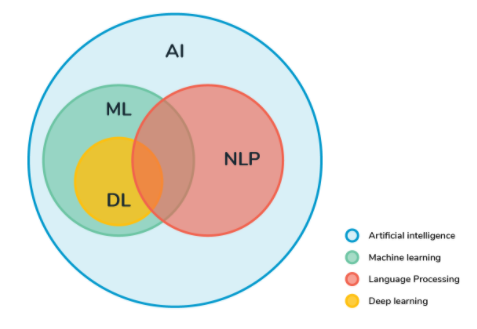
## **NATURAL LANGUAGE PROCESSING**

### **Introduction [Giriş]**

NLP or Natural Language Processing is a subfield of Artificial Intelligence that gives machines the ability to understand and extract meaning from human languages. [NLP veya Doğal Dil İşleme, makinelere insan dillerini anlama ve anlam çıkarma yeteneği veren Yapay Zekanın bir alt alanıdır.]



NLP is a field that focuses on the interaction between data science and human language. [NLP, veri bilimi ve insan dili arasındaki etkileşime odaklanan bir alandır.] It allows data scientists to derive meaningful results in areas like media, healthcare, finance, and human resources, etc. [Veri bilimcilerinin medya, sağlık, finans ve insan kaynakları gibi alanlarda anlamlı sonuçlar elde etmelerini sağlar.] Nowadays NLP is booming thanks to the huge improvements in acquiring the data and the sharp increase in computational power. [Günümüzde NLP, verilerin elde edilmesindeki büyük gelişmeler ve hesaplama gücündeki keskin artış sayesinde patlama yaşıyor.]

Q:What do you understand by Natural Language Processing? [S: Doğal Dil İşlemeden ne anlıyorsunuz?]  
A: Natural Language Processing is a field of computer science that deals with communication between computer systems and humans. [C: Doğal Dil İşleme, bilgisayar sistemleri ve insanlar arasındaki iletişimle ilgilenen bir bilgisayar bilimi alanıdır.] It is a technique used in Artificial Intelligence and Machine Learning. [Yapay Zeka ve Makine Öğreniminde kullanılan bir tekniktir.] It is used to create automated software that helps understand human spoken languages to extract useful information from the data it gets in the form of audio. [Ses biçiminde aldığı verilerden yararlı bilgiler çıkarmak için insan konuşulan dilleri anlamaya yardımcı olan otomatik bir yazılım oluşturmak için kullanılır.] Techniques in NLP allow computer systems to process and interpret data in the form of natural languages. [NLP'deki teknikler, bilgisayar sistemlerinin verileri doğal diller biçiminde işlemesine ve yorumlamasına izin verir.]

- Interview Q&A

### **NLP Applications Worldwide**



NLP can help people with many tasks. [NLP, insanlara birçok görevde yardımcı olabilir.] Some examples are given below. [Bazı örnekler aşağıda verilmiştir.]

**Diagnosing:** Prediction of diseases based on the patient’s own speech and electronic health records. [Teşhis: Hastanın kendi konuşma ve elektronik sağlık kayıtlarına dayanarak hastalıkların tahmini.]

**Sentiment Analysis:** Organizations can determine what customers are feeling about a product or service by extracting information from sources like social media. [Duygu Analizi: Kuruluşlar, sosyal medya gibi kaynaklardan bilgi çıkararak müşterilerin bir ürün veya hizmet hakkında ne hissettiğini belirleyebilir.]

**Translator:** Online translators have never been so successful before NLP was used in that field. [Çevirmen: NLP bu alanda kullanılmadan önce çevrimiçi çevirmenler hiç bu kadar başarılı olmamıştı.]

**ChatBot:**To communicate with the customer like an actual employee. [ChatBot: Müşteri ile gerçek bir çalışan gibi iletişim kurmak.]

**Classifying emails:**  To classify emails as spam or ham and stop spams before they even enter the inbox. [E-postaları sınıflandırma:  E-postaları spam veya jambon olarak sınıflandırmak ve spam'leri daha gelen kutusuna girmeden durdurmak için.]

**Detecting Fake News:** To determine if a source is politically biased or accurate, detecting if a news source can be trusted or not. [Sahte Haber Tespiti: Bir kaynağın politik olarak taraflı mı yoksa doğru mu olduğunu belirlemek, bir haber kaynağına güvenilip güvenilmeyeceğini tespit etmek.]

**Intelligent Voice-Driven Interfaces:** Apple’s Siri or Android's Iris are examples of intelligent voice-driven interfaces that use NLP to respond to humans. [Akıllı Sese Dayalı Arabirimler: Apple'ın Siri'si veya Android'in Iris'i, insanlara yanıt vermek için NLP kullanan akıllı sesle çalışan arabirimlere örnektir.]

**Trading Algorithms:** Tracking news, reports, comments about financing to sell or buy the stocks automatically. [Ticaret Algoritmaları: Hisse senetlerini otomatik olarak satmak veya satın almak için finansmanla ilgili haberleri, raporları, yorumları takip edin.]

**Recruiting Assistant:** Both the search and selection phases of new employees and identifying the skills of potential hires. [İşe Alım Asistanı: Hem yeni çalışanların arama ve seçim aşamaları hem de potansiyel işe alımların becerilerinin belirlenmesi.]

**Litigation Tasks:** To automate routine litigation tasks and help courts save time. [Dava Görevleri: Rutin dava görevlerini otomatikleştirmek ve mahkemelerin zaman kazanmasına yardımcı olmak için.]

In this lesson, you will study email classification with NLP. [Bu derste, NLP ile e-posta sınıflandırmasını öğreneceksiniz.]

Q:List any two real-life applications of Natural Language Processing. [S:Doğal Dil İşleme'nin gerçek hayattaki iki uygulamasını listeleyin.]  
A: Two real-life applications of Natural Language Processing are as follows: [C: Doğal Dil İşleme'nin gerçek hayattaki iki uygulaması aşağıdaki gibidir:]

**Google Translate**: Google Translate is one of the famous applications of Natural Language Processing. [Google Çeviri: Google Çeviri, Doğal Dil İşleme'nin ünlü uygulamalarından biridir.] It helps convert written or spoken sentences into any language. [Yazılı veya sözlü cümleleri herhangi bir dile dönüştürmeye yardımcı olur.] Also, we can find the correct pronunciation and meaning of a word by using Google Translate. [Ayrıca Google Translate'i kullanarak bir kelimenin doğru telaffuzunu ve anlamını bulabiliriz.] It uses advanced techniques of Natural Language Processing to achieve success in translating sentences into various languages. [Cümleleri çeşitli dillere çevirmede başarı elde etmek için gelişmiş Doğal Dil İşleme tekniklerini kullanır.]

**Chatbots:** To provide a better customer support service, companies have started using chatbots for 24/7 service. [Chatbot'lar: Daha iyi bir müşteri destek hizmeti sağlamak için şirketler 7/24 hizmet için chatbot'ları kullanmaya başladı.] Chatbots helps resolve the basic queries of customers. [Chatbotlar, müşterilerin temel sorularını çözmeye yardımcı olur.] If a chatbot is not able to resolve any query, then it forwards it to the support team, while still engaging the customer. [Bir sohbet robotu herhangi bir sorguyu çözemezse, müşteriyle iletişim kurmaya devam ederken bunu destek ekibine iletir.] It helps make customers feel that the customer support team is quickly attending them. [Müşterilere, müşteri destek ekibinin hızla onlara katıldığını hissettirmeye yardımcı olur.] With the help of chatbots, companies have become capable of building cordial relations with customers. [Chatbotların yardımıyla şirketler, müşterilerle samimi ilişkiler kurma yeteneğine sahip oldular.] It is only possible with the help of Natural Language Processing. [Bu ancak Doğal Dil İşleme yardımıyla mümkündür.]

- Interview Q&A

### **NLP Theory [NLP Teorisi]**

In this lesson, you will study the following concepts; [Bu derste aşağıdaki kavramları çalışacaksınız;]

* Document and Corpus, [Belge ve Derlem,]
* Vectorization, [vektörleştirme,]
* Bag of Words, [Kelime Torbası,]
* TF-IDF. [TF-IDF.]

**Document vs Corpus: [Belge ve Corpus:]**

Corpus refers to a collection of texts. [Derlem, bir metin koleksiyonunu ifade eder.]   It is described as "a large and structured set of texts" in Wikipedia. [Wikipedia'da 'geniş ve yapılandırılmış bir metin kümesi' olarak tanımlanmaktadır.] The term document has a very restricted meaning when compared with the corpus. [Belge terimi, külliyatla karşılaştırıldığında çok sınırlı bir anlama sahiptir.] For example, you are implementing some NLP tasks using a book. [Örneğin, bir kitap kullanarak bazı NLP görevlerini uyguluyorsunuz.] Let's consider each paragraph of the book is a row in your dataset. [Kitabın her paragrafını veri kümenizdeki bir satır olarak düşünelim.] The book itself is the corpus and each paragraph is a document. [Kitabın kendisi külliyattır ve her paragraf bir belgedir.] Another example can be given about the dataset you will be working on during this course. [Bu derste üzerinde çalışacağınız veri seti ile ilgili başka bir örnek verilebilir.] Each SMS message will be a document and the whole dataset (whole SMS messages) will form the corpus. [Her SMS mesajı bir belge olacak ve tüm veri seti (tüm SMS mesajları) tümceyi oluşturacaktır.]

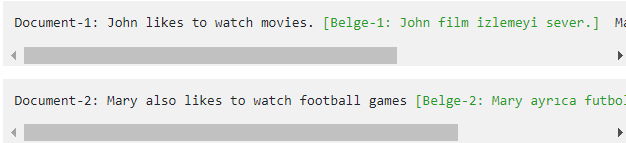
**Vectorization : [Vektörleştirme :]**

Word vectorization is a methodology in NLP to map words from vocabulary to a corresponding numeric vector. [Kelime vektörleştirme, kelimeleri kelime dağarcığından karşılık gelen bir sayısal vektöre eşlemek için NLP'de bir metodolojidir.] In other words, it is the process of converting words into numbers. [Başka bir deyişle, kelimeleri sayılara dönüştürme işlemidir.]

There are several methods to implement vectorization. [Vektörleştirmeyi uygulamak için birkaç yöntem vardır.] But maybe the most famous methods are listed below: [Ama belki de en ünlü yöntemler aşağıda listelenmiştir:]

* Count Vectors (CountVectorizer) [Sayma Vektörleri (CountVectorizer)]
* TF-IDF Vectors (TfidfVectorizer) [TF-IDF Vektörleri (TfidfVectorizer)]
* Word Embeddings (Word2Word, GloVe, etc.) [Kelime Gömmeleri (Word2Word, GloVe, vb.)]

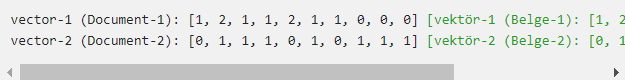
In this course, you will learn CountVectorizer and TF-IDF methods. [Bu derste CountVectorizer ve TF-IDF yöntemlerini öğreneceksiniz.] Maybe CountVectorizer is the simplest method to vectorize the text. [Belki CountVectorizer, metni vektörleştirmenin en basit yöntemidir.]   The logic behind that method is to count each word in each document. [Bu yöntemin arkasındaki mantık, her belgedeki her kelimeyi saymaktır.] Let's have a look following two documents. [Şimdi iki belgeyi takip edelim.]



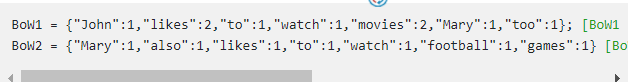
The CountVectorizer will convert a collection of text documents to a matrix of token counts. [CountVectorizer, bir metin belgesi koleksiyonunu belirteç sayısı matrisine dönüştürür.]  We can imagine this as a 2-Dimensional matrix. [Bunu 2 Boyutlu bir matris olarak hayal edebiliriz.] Where the 1-dimension is the entire vocabulary (1 row per word) and the other dimension is the actual documents, in this case, a column per document. [1 boyutun tüm kelime dağarcığı (kelime başına 1 satır) ve diğer boyutun asıl belgeler olduğu durumda, bu durumda belge başına bir sütun.]



Vectors of each document can be demonstrated as follows. [Her belgenin vektörleri aşağıdaki gibi gösterilebilir.]

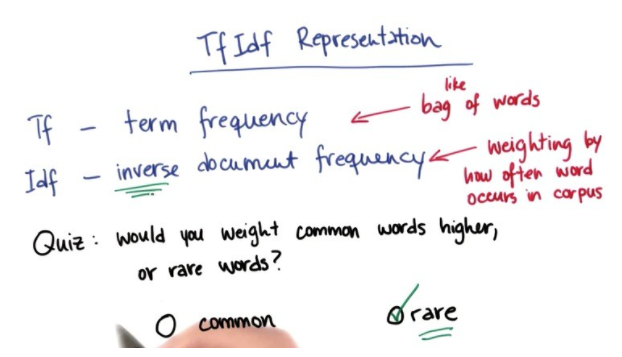


its Jason object representation will be as follows. [Jason nesne gösterimi aşağıdaki gibi olacaktır.] This method is called the **Bag of Words (BoW)**modeling. [Bu yönteme Kelime Torbası (BoW) modellemesi denir.]



**What is TF-IDF? [TF-IDF nedir?]**

We can say that this method is more advanced than just counting the words. [Bu yöntemin sadece kelimeleri saymaktan daha gelişmiş olduğunu söyleyebiliriz.]  TF-IDF stands for term frequency-inverse document frequency, and the tf-idf weight is a weight used in NLP. [TF-IDF, frekans-ters belge frekansı terimini ifade eder ve tf-idf ağırlığı, NLP'de kullanılan bir ağırlıktır.]   This weight is a statistical measure used to evaluate how important a word is to a document in a corpus. [Bu ağırlık, bir kelimenin bir bütüncedeki bir belge için ne kadar önemli olduğunu değerlendirmek için kullanılan istatistiksel bir ölçüdür.]  The importance increases proportionally to the number of times, a word appears in the document but is offset by the frequency of the word in the corpus. [Önem, bir sözcüğün belgede görünme sayısıyla orantılı olarak artar, ancak tümcedeki sözcüğün sıklığıyla dengelenir.]



**What is TF (Term Frequency)? [TF (Dönem Frekansı) nedir?]**

TF measures how frequently a term occurs in a document. [TF, bir belgede bir terimin ne sıklıkta geçtiğini ölçer.] Since every document is different in length, it is possible that a term can appear much more times in long documents than shorter ones. [Her belgenin uzunluğu farklı olduğundan, bir terimin uzun belgelerde kısa olanlara göre çok daha fazla görünmesi mümkündür.] Thus, the term frequency is divided by the document length as a way of normalization: [Böylece, terim sıklığı, normalleştirmenin bir yolu olarak belge uzunluğuna bölünür:]

*TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document). [TF(t) = (Bir belgede t teriminin görünme sayısı) / (Belgedeki toplam terim sayısı).]*

**What is IDF (Inverse Document Frequency)? [IDF (Ters Belge Frekansı) Nedir?]**

IDF measures how important a term is. [IDF, bir terimin ne kadar önemli olduğunu ölçer.] While computing TF, all terms are considered equally important. [TF hesaplanırken, tüm terimler eşit derecede önemli kabul edilir.] However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. [Bununla birlikte, 'is', 'of' ve 'that' gibi belirli terimlerin birçok kez görünebileceği ancak çok az öneme sahip olduğu bilinmektedir.] Thus we need to weigh down the frequent terms while scaling up the rare ones, by computing the following: [Bu nedenle, aşağıdakileri hesaplayarak nadir terimleri ölçeklendirirken sık kullanılan terimleri tartmamız gerekir:]

*IDF(t) = log(Total number of documents  in a corpus/ Number of documents with term t in it). [IDF(t) = log(Bir bütüncedeki toplam doküman sayısı/ İçinde t terimi olan dokümanların sayısı).]*

**Example: [Örnek:]**

Consider a document containing 100 words wherein the word **cow** appears 3 times. [İnek kelimesinin 3 kez geçtiği 100 kelimelik bir doküman düşünün.]

The TF for cow is then (3 / 100) = 0.03. [İnek için TF bu durumda (3/100) = 0.03'tür.] Now, let's assume the corpus has 10 million documents and the word cow appears in one thousand of these. [Şimdi, derlemin 10 milyon belgesi olduğunu ve bunların bininde inek kelimesinin geçtiğini varsayalım.] Then, the IDF is calculated as log(10,000,000 / 1,000) = 4. [Daha sonra IDF log(10.000.000 / 1.000) = 4 olarak hesaplanır.] Thus, the Tf-idf weight is the product of these quantities: 0.03 \* 4 = 0.12. [Böylece, Tf-idf ağırlığı şu miktarların çarpımıdır: 0.03 \* 4 = 0.12.]

# Natural Language Processing In 5 Minutes | What Is NLP And How Does It Work? | Simplilearn

<https://www.youtube.com/watch?v=CMrHM8a3hqw>

### **Data Preparation With NLP**

In this lesson, you will study text pre-processing. [Bu derste, metin ön işlemeyi öğreneceksiniz.] The text pre-processing simply means to bring your text into a form that is analyzable and predictable for your task. [Metin ön işleme, metninizi göreviniz için analiz edilebilir ve tahmin edilebilir bir forma getirmek anlamına gelir.] In this respect you will learn how to; [Bu bağlamda öğreneceğiniz;]

* Remove punctuation, [noktalama işaretlerini kaldır,]
* Remove stopwords, [Durdurulan kelimeleri kaldır,]
* Tokenize the text, [Metni tokenize edin,]
* Vectorize the words. [Kelimeleri vektörleştirin.]

**Removing Punctuation: [Noktalama İşaretlerini Kaldırma:]**

There might be some punctuation such as commas, quotes, apostrophes, question marks, and more. [Virgül, tırnak işareti, kesme işareti, soru işareti ve daha fazlası gibi bazı noktalama işaretleri olabilir.]  We cannot feed a machine learning model from raw text. [Bir makine öğrenimi modelini ham metinden besleyemeyiz.] We need to clean the text first. [Önce metni temizlememiz gerekiyor.] Removing the punctuation is usually one of the first steps of cleaning the text. [Noktalama işaretlerini kaldırmak genellikle metni temizlemenin ilk adımlarından biridir.] Because the model does not need them. [Çünkü modelin onlara ihtiyacı yok.]

**Removing Stopwords: [Durdurulan Kelimeleri Kaldırmak:]**

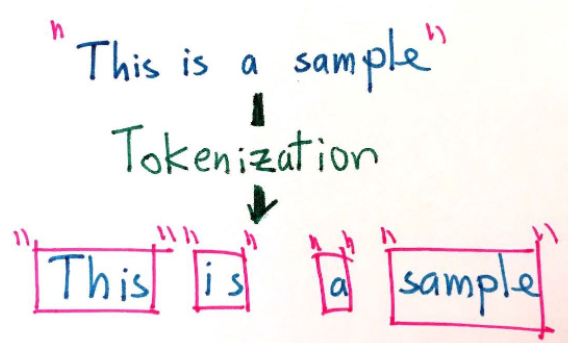
Stopwords do not contribute to the meaning of the text deeply. [Durdurulan kelimeler metnin anlamına derinlemesine katkıda bulunmaz.] These words introduce much noise because they appear more frequently than other words. [Bu kelimeler, diğer kelimelerden daha sık göründükleri için çok gürültü çıkarır.] Some examples of stopwords are given below: [Bazı stopwords örnekleri aşağıda verilmiştir:]

*"and",  "the",  "how", "all",  "about", "on", "under", "up",  "after", "i", "me", "myself", "we", our", ours", your", "yours"... ['ve',  'the',  'nasıl', 'hepsi',  'hakkında', 'açık', 'altında', 'yukarı',  'sonra', 'i', 'ben', 'kendim', 'biz ', bizim', bizim', sizin', 'sizin'...]*

We filter out these stopwords before doing any statistical analysis or creating a model. [Herhangi bir istatistiksel analiz yapmadan veya bir model oluşturmadan önce bu kilit kelimeleri filtreleriz.]

**Tokenization: [Tokenizasyon:]**

Tokenization means splitting a sentence, paragraph, phrase, or an entire text document into smaller units, such as individual words or terms. [Belirteçleştirme, bir cümleyi, paragrafı, tümceyi veya tüm metin belgesini tek tek kelimeler veya terimler gibi daha küçük birimlere bölmek anlamına gelir.] Each of these smaller units is called tokens. [Bu daha küçük birimlerin her birine jeton denir.] You can consider each word as a token. [Her kelimeyi bir belirteç olarak düşünebilirsiniz.]



Tokenization is significant because the meaning of the text can easily be interpreted by analyzing the words present in the text. [Belirteçleştirme önemlidir, çünkü metnin anlamı, metinde bulunan kelimeler analiz edilerek kolayca yorumlanabilir.] We can count the number of words in the text after tokenization. [Tokenizasyondan sonra metindeki kelime sayısını sayabiliriz.] You can find more information at this [link](https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/) about tokenization methods with python [Python ile belirteçleştirme yöntemleri hakkında bu bağlantıda daha fazla bilgi bulabilirsiniz.]

<https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/>

**Vectorization : [Vektörleştirme :]**

Word vectorization is a methodology in NLP to map words from vocabulary to a corresponding numeric vector. [Kelime vektörleştirme, kelimeleri kelime dağarcığından karşılık gelen bir sayısal vektöre eşlemek için NLP'de bir metodolojidir.] In other words, it is the process of converting words into numbers. [Başka bir deyişle, kelimeleri sayılara dönüştürme işlemidir.]

There are several methods to implement vectorization. [Vektörleştirmeyi uygulamak için birkaç yöntem vardır.] But maybe the most famous methods are listed below: [Ama belki de en ünlü yöntemler aşağıda listelenmiştir:]

* Count Vectors (CountVectorizer) [Sayma Vektörleri (CountVectorizer)]
* TF-IDF Vectors (TfidfVectorizer) [TF-IDF Vektörleri (TfidfVectorizer)]
* Word Embeddings (Word2Word, GloVe, etc.) [Kelime Gömmeleri (Word2Word, GloVe, vb.)]

In this part, you will study CountVectorizer. [Bu bölümde CountVectorizer çalışacaksınız.] Maybe it is the simplest method to vectorize the text. [Belki de metni vektörleştirmenin en basit yöntemidir.]   The logic behind that method is to count each word in each document. [Bu yöntemin arkasındaki mantık, her belgedeki her kelimeyi saymaktır.]

# Text Data Cleaning In Python | How to clean text data in python

<https://www.youtube.com/watch?v=KhXU7KOxQcg>

Q:What are stopwords? [S: Durdurulan kelimeler nelerdir?]  
A: Stopwords are said to be useless data for a search engine. [C: Durdurulan kelimelerin bir arama motoru için gereksiz veriler olduğu söylenir.] Words such as articles, prepositions, etc. are considered as stopwords. [Makaleler, edatlar vb. kelimeler, stopwords olarak kabul edilir.] There are stopwords such as was, were, is, am, the, a, an, how, why, and many more. [Oldu, was, is, am, the, a, an, how, Why ve daha pek çok sözcük vardır.] In Natural Language Processing, we eliminate the stopwords to understand and analyze the meaning of a sentence. [Doğal Dil İşleme'de, bir cümlenin anlamını anlamak ve analiz etmek için engellenen kelimeleri ortadan kaldırıyoruz.] The removal of stopwords is one of the most important tasks for search engines. [Durdurulan kelimelerin kaldırılması, arama motorları için en önemli görevlerden biridir.] Engineers design the algorithms of search engines in such a way that they ignore the use of stopwords. [Mühendisler, arama motorlarının algoritmalarını, duraklama kelimelerinin kullanımını göz ardı edecek şekilde tasarlar.] This helps show the relevant search result for a query. [Bu, bir sorgu için ilgili arama sonucunun gösterilmesine yardımcı olur.]  
- Interview Q&A

### **Practice With NLP In ML [ML'de NLP ile Pratik Yapın]**

In this lesson, you will study; [Bu derste çalışacaksınız;]

* Document and Corpus concepts, [Belge ve Derlem kavramları,]
* TF-IDF, [TF-IDF,]
* Training a model, [Model yetiştirmek,]
* Model evaluation, [Model değerlendirme,]
* Creating a pipeline. [Bir boru hattı oluşturma.]

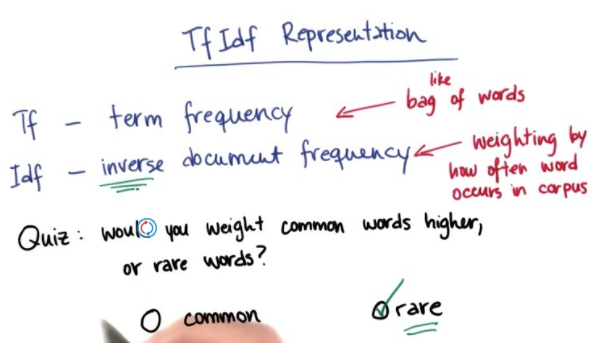
You have already learned the document, corpus, and TF-IDF concepts. [Belge, derlem ve TF-IDF kavramlarını zaten öğrendiniz.] But maybe it is useful to review these terms here. [Ama belki bu terimleri burada gözden geçirmekte fayda var.]

**Document vs Corpus: [Belge ve Corpus:]**

Corpus refers to a collection of texts. [Derlem, bir metin koleksiyonunu ifade eder.]   It is described as "a large and structured set of texts" in Wikipedia. [Wikipedia'da 'geniş ve yapılandırılmış bir metin kümesi' olarak tanımlanmaktadır.] The term document has a very restricted meaning when compared with the corpus. [Belge terimi, külliyatla karşılaştırıldığında çok sınırlı bir anlama sahiptir.] For example, you are implementing some NLP tasks using a book. [Örneğin, bir kitap kullanarak bazı NLP görevlerini uyguluyorsunuz.] Let's consider each paragraph of the book is a row in your dataset. [Kitabın her paragrafını veri kümenizdeki bir satır olarak düşünelim.] The book itself is the corpus and each paragraph is a document. [Kitabın kendisi külliyattır ve her paragraf bir belgedir.] Another example can be given about the dataset you are working on it now. [Şu anda üzerinde çalıştığınız veri seti ile ilgili başka bir örnek verilebilir.] Each SMS message is a document and the whole dataset is the corpus. [Her SMS mesajı bir belgedir ve tüm veri kümesi bütüncedir.]

**What is TF-IDF? [TF-IDF nedir?]**

TF-IDF stands for term frequency-inverse document frequency, and the tf-idf weight is a weight used in NLP. [TF-IDF, frekans-ters belge frekansı terimini ifade eder ve tf-idf ağırlığı, NLP'de kullanılan bir ağırlıktır.]   This weight is a statistical measure used to evaluate how important a word is to a document in a corpus. [Bu ağırlık, bir kelimenin bir bütüncedeki bir belge için ne kadar önemli olduğunu değerlendirmek için kullanılan istatistiksel bir ölçüdür.]  The importance increases proportionally to the number of times a word appears in the document (in a SMS message) but is offset by the frequency of the word in the corpus (whole SMS messages). [Önem, bir kelimenin belgede (bir SMS mesajında) görünme sayısı ile orantılı olarak artar, ancak kelimenin bütündeki (tüm SMS mesajları) sıklığı ile dengelenir.]



# Countvectorizer and TF IDF in Python|Text feature extraction in Python

<https://www.youtube.com/watch?v=lBO1L8pgR9s&t=2s>

Q: What is TF-IDF? [S: TF-IDF nedir?]  
A: TF-IDF or Term Frequency-Inverse Document Frequency indicates the importance of a word in a set. [A: TF-IDF veya Terim Frekansı-Ters Belge Frekansı, bir kümedeki bir kelimenin önemini belirtir.] It helps in information retrieval with numerical statistics. [Sayısal istatistiklerle bilgi alınmasına yardımcı olur.] For a specific document, TF-IDF shows a frequency that helps identify the keywords in a document. [Belirli bir belge için TF-IDF, bir belgedeki anahtar sözcükleri tanımlamaya yardımcı olan bir sıklığı gösterir.] The major use of TF-IDF in NLP is the extraction of useful information from crucial documents by statistical data. [TF-IDF'nin NLP'deki başlıca kullanımı, önemli belgelerden istatistiksel verilerle faydalı bilgilerin çıkarılmasıdır.] It is ideally used to classify and summarize the text in documents and filter out stop words. [İdeal olarak, belgelerdeki metni sınıflandırmak ve özetlemek ve durak sözcüklerini filtrelemek için kullanılır.]

TF helps calculate the ratio of the frequency of a term in a document and the total number of terms. [TF, bir belgedeki bir terimin sıklığının ve toplam terim sayısının oranını hesaplamaya yardımcı olur.] Whereas, IDF denotes the importance of the term in a document. [Oysa IDF, bir belgedeki terimin önemini belirtir.]

The formula for calculating TF-IDF: [TF-IDF hesaplama formülü:]

**TF(W) = (Frequency of W in a document)/(The total number of terms in the document) [TF(W) = (Bir belgedeki W'nin sıklığı)/(Belgedeki toplam terim sayısı)]**

**IDF(W) = log\_e(The total number of documents/The number of documents having the term W) [IDF(W) = log\_e(Toplam belge sayısı/W terimine sahip belge sayısı)]**

Google uses TF-IDF to decide the index of search results according to the relevancy of pages. [Google, sayfaların alaka düzeyine göre arama sonuçlarının dizinine karar vermek için TF-IDF'yi kullanır.] The design of the TF-IDF algorithm helps optimize the search results in Google. [TF-IDF algoritmasının tasarımı, Google'daki arama sonuçlarını optimize etmeye yardımcı olur.] It helps quality content rank up in search results. [Kaliteli içeriğin arama sonuçlarında yükselmesine yardımcı olur.]

- Interview Q&A

### **Word Embedding [Kelime Gömme]**

Word embeddings are numeric vectors that show semantic relationships between words. [Kelime yerleştirmeleri, kelimeler arasındaki anlamsal ilişkileri gösteren sayısal vektörlerdir.] To see how close the two words are to each other semantically, the cosinus similarity of these word vectors is checked. [İki kelimenin anlamsal olarak birbirine ne kadar yakın olduğunu görmek için bu kelime vektörlerinin kosinüs benzerliği kontrol edilir.] The closer the value is to 1, the closer the words are semantically. [Değer 1'e ne kadar yakınsa, kelimeler anlamsal olarak o kadar yakındır.]

Cosine similarity is a measurement that quantifies the similarity between two or more vectors. [Kosinüs benzerliği, iki veya daha fazla vektör arasındaki benzerliği ölçen bir ölçümdür.] The cosine similarity is the cosine of the angle between vectors. [Kosinüs benzerliği, vektörler arasındaki açının kosinüsüdür.] The cosine similarity is a value that is bound by a constrained range of 0 and 1. [Kosinüs benzerliği, 0 ve 1'lik sınırlı bir aralıkla sınırlanan bir değerdir.] The similarity measurement is a measure of the cosine of the angle between the two non-zero vectors A and B. [Benzerlik ölçümü, sıfır olmayan iki vektör A ve B arasındaki açının kosinüsünün bir ölçüsüdür.]

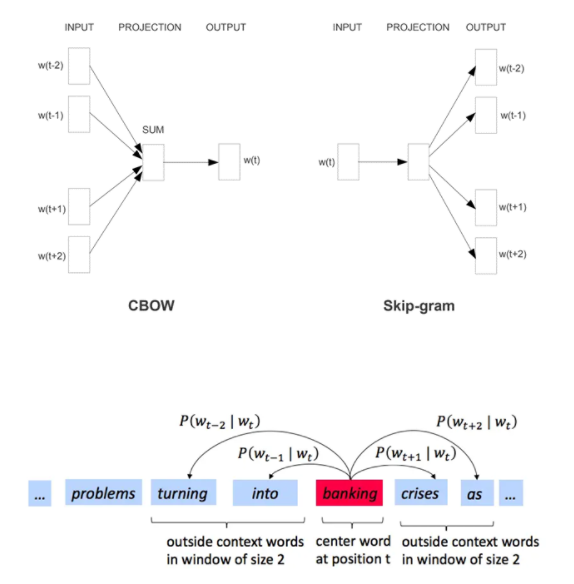
#### **Word Embedding Algorithms [Kelime Gömme Algoritmaları]**

**Embedding Layer**: It is a word embedding that is learned in conjunction with a neural network model for the purpose of performing a particular natural language processing job, such as language modeling or document classification. [Gömme Katmanı: Dil modelleme veya belge sınıflandırma gibi belirli bir doğal dil işleme işini gerçekleştirmek amacıyla bir sinir ağı modeliyle bağlantılı olarak öğrenilen bir kelime gömmedir.]

**Word2Vec**: It is a statistical method for efficiently learning a word embedding from a text corpus. [Word2Vec : Bir metin korpusundan gömülü bir kelimeyi verimli bir şekilde öğrenmek için istatistiksel bir yöntemdir.] It was developed by Google in 2013. [2013 yılında Google tarafından geliştirilmiştir.] There are 2 different algorithms for training in Word2vect. [Word2vect'te eğitim için 2 farklı algoritma vardır.] These are the **Continuous Bag-of-Words (CBOW model)** and the **Skip-Gram Model**. [Bunlar Sürekli Kelime Torbası (CBOW modeli) ve Skip-Gram Modelidir.]

In the **CBOW model**, a word (middle word) in the sentence is tried to be predicted based on the words around that word. [CBOW modelinde, cümledeki bir kelime (orta kelime), o kelimenin etrafındaki kelimelerden hareketle tahmin edilmeye çalışılır.] (By giving the words around it as input to the model, the middle word is tried to be predicted) [(Etrafındaki kelimeler modele girdi olarak verilerek ortadaki kelime tahmin edilmeye çalışılır)]

In **the Skip-Gram model**, based on a word (middle word) in the sentence, the words around the middle word are tried to be predicted. [Skip-Gram modelinde cümledeki bir kelimeden (ortadaki kelime) yola çıkılarak orta kelimenin etrafındaki kelimeler tahmin edilmeye çalışılır.] (The words around the middle word are tried to be predicted by giving the middle word as an input to the model.) [(Ortadaki kelime modele girdi olarak verilerek, ortadaki kelimenin etrafındaki kelimeler tahmin edilmeye çalışılır.)]



**Global Vectors (Glove):**It was developed by Stanford University in 2014. [Küresel Vektörler (Eldiven): 2014 yılında Stanford Üniversitesi tarafından geliştirilmiştir.] It is a model that takes into account the statistics of using all the words in the corpus with other words. (in the corpus, the word red is used together with the word car 100 times, etc.) [Derlemdeki tüm kelimeleri başka kelimelerle kullanma istatistiklerini dikkate alan bir modeldir. (bütüncede kırmızı kelimesi araba kelimesi ile birlikte 100 defa kullanılır vb.)]

**Embeddings from Language Models (ELMo):**The word2vec and glove models are unfortunately unsuccessful in capturing homophones with different meanings. [Dil Modellerinden Gömmeler (ELMo): Word2vec ve eldiven modelleri ne yazık ki farklı anlamlara sahip sesteş sözcükleri yakalamada başarısızdır.] However, the Elmo model is quite successful in capturing the semantic differences between words. [Ancak Elmo modeli kelimeler arasındaki anlam farklılıklarını yakalamada oldukça başarılıdır.]

I love the**apple**that my mother bought from the market. [Annemin pazardan aldığı elmayı seviyorum.]

I love the**apple,**the biggest company in the world. [Dünyanın en büyük şirketi olan elmayı seviyorum.]

Word2vec and Glove modules can't capture the difference in semantics between the apple words in the two different sentences above. [Word2vec ve Glove modülleri, yukarıdaki iki farklı cümlede elma kelimeleri arasındaki anlam farkını yakalayamaz.] However, the ELMo model will capture this difference. [Ancak ELMo modeli bu farkı yakalayacaktır.]

# Converting words to numbers, Word Embeddings | Deep Learning Tutorial 39 (Tensorflow & Python)

<https://www.youtube.com/watch?v=sZGuyTLjsco>

# What is Word2Vec? A Simple Explanation | Deep Learning Tutorial 41 (Tensorflow, Keras & Python)

<https://www.youtube.com/watch?v=hQwFeIupNP0>

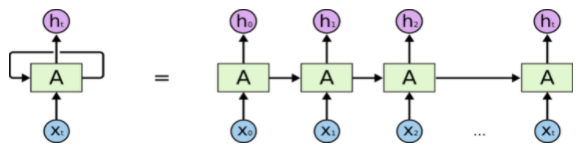
Q: What is Word embeddings? [S: Word yerleştirmeleri nedir?]

##### **A: Word embeddings are numeric vectors that show semantic relationships between words. [C: Kelime yerleştirmeleri kelimeler arasındaki anlamsal ilişkileri gösteren sayısal vektörlerdir.] To see how close the two words are to each other semantically, the cosinus similarity of these word vectors is checked. [İki kelimenin anlamsal olarak birbirine ne kadar yakın olduğunu görmek için bu kelime vektörlerinin kosinüs benzerliği kontrol edilir.] The closer the value is to 1, the closer the words are semantically. [Değer 1'e ne kadar yakınsa, kelimeler anlamsal olarak o kadar yakındır.]**

### **Recurrent Neural Networks (RNN)**

#### **Recurrent Neural Networks (RNN)**

The RNNs are a kind of Deep Learning construct that is generally used to predict the next step. [RNN'ler, genellikle bir sonraki adımı tahmin etmek için kullanılan bir tür Derin Öğrenme yapısıdır.] The biggest difference between them and other deep learning structures is that they remember. [Onlarla diğer derin öğrenme yapıları arasındaki en büyük fark, hatırlamalarıdır.] Another difference is that, while in other neural networks, each input is independent of the other, in RNNs the inputs are related to each other. [Diğer bir fark ise, diğer sinir ağlarında her giriş diğerinden bağımsız iken, RNN'lerde girişler birbiriyle ilişkilidir.] RNNs make associations between inputs to follow the next step and remember all their associations while they are being trained. [RNN'ler, bir sonraki adımı takip etmek için girdiler arasında ilişkilendirmeler yapar ve eğitilirken tüm çağrışımlarını hatırlar.]



But, vanishing and exploding gradients are two frequent issues that arise during RNN backpropagation. [Ancak, kaybolan ve patlayan gradyanlar, RNN geri yayılımı sırasında ortaya çıkan iki sık sorundur.] In the vanishing gradient problem, the gradient values decrease rapidly, and the training stops. [Kaybolan gradyan probleminde gradyan değerleri hızla düşer ve eğitim durur.] The model can't capture the relationships between the beginning and the end of long sentences. [Model, uzun cümlelerin başı ve sonu arasındaki ilişkileri yakalayamaz.] But in the exploding gradients, the gradient values grow to infinity exponentially rapidly and model weight values become NaN due to the unstable process. [Ancak patlayan gradyanlarda, gradyan değerleri katlanarak sonsuza kadar büyür ve kararsız süreç nedeniyle model ağırlık değerleri NaN olur.] The model can't learn anything from training data. [Model, eğitim verilerinden hiçbir şey öğrenemez.]

#### **Short Term Memory Problem. [Kısa Süreli Bellek Sorunu.]**

Short-term memory is a problem for recurrent neural networks due to vanishing gradient issues. [Kısa süreli bellek, kaybolan gradyan sorunları nedeniyle tekrarlayan sinir ağları için bir sorundur.] They'll have difficulty transferring information from earlier time steps to later ones if the sequence is lengthy enough. [Eğer dizi yeterince uzunsa, bilgiyi önceki zaman adımlarından sonraki adımlara aktarmakta zorluk çekeceklerdir.] If you're attempting to predict anything from a paragraph of text, RNN’s may leave out essential information at the beginning of sequences. [Bir metin paragrafından herhangi bir şey tahmin etmeye çalışıyorsanız, RNN'ler dizilerin başında önemli bilgileri dışarıda bırakabilir.] The vanishing gradient problem affects recurrent neural networks during backpropagation. [Kaybolan gradyan sorunu, geri yayılım sırasında tekrarlayan sinir ağlarını etkiler.]

The vanishing gradient problem affects recurrent neural networks during backpropagation. [Kaybolan gradyan sorunu, geri yayılım sırasında tekrarlayan sinir ağlarını etkiler.] Gradients are values that are used to update the weights of a neural network. [Gradyanlar, bir sinir ağının ağırlıklarını güncellemek için kullanılan değerlerdir.] When a gradient diminishes as it backpropagates through time, this is known as the vanishing gradient issue. [Bir gradyan zamanla geri yayılırken azaldığında, bu kaybolan gradyan sorunu olarak bilinir.] When a gradient value falls below a certain threshold, it no longer contributes much to learning. [Bir gradyan değeri belirli bir eşiğin altına düştüğünde, artık öğrenmeye fazla bir katkı sağlamaz.]

**updated new weight = weight – learning rate \* gradient [güncellenmiş yeni ağırlık = ağırlık – öğrenme oranı \* gradyan]**

**1.01  = 1.01001 – 0.00001 [1.01 = 1.01001 – 0.00001]**

**1.00999 = 1.01 – 0.00001 [1.00999 = 1.01 – 0.00001]**

**1.00998 = 1.00999 – 0.00001 [1.00998 = 1.00999 – 0.00001]**

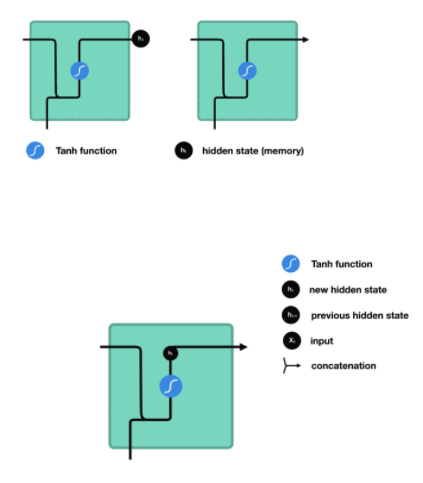
**(As can be seen above, when the gradient value is too small, the update in the weights almost comes to a halt.) [(Yukarıda görüldüğü gibi gradyan değeri çok küçük olduğunda ağırlıklardaki güncelleme neredeyse durma noktasına gelmektedir.)]**

Layers that get a tiny gradient update in recurrent neural networks stop learning. [Tekrarlayan sinir ağlarında küçük bir gradyan güncellemesi alan katmanlar öğrenmeyi durdurur.] Those are generally the first layers to appear. [Bunlar genellikle görünen ilk katmanlardır.] RNNs can forget what they've seen in longer sequences since these layers don't learn, resulting in short-term memory. [RNN'ler, bu katmanlar öğrenmediği için daha uzun dizilerde gördüklerini unutabilir ve bu da kısa süreli belleğe neden olur.]

RNNs work very well in short sequences (like short sentences) and have less computational cost. [RNN'ler kısa dizilerde (kısa cümleler gibi) çok iyi çalışır ve daha az hesaplama maliyetine sahiptir.] However, they can’t show this performance in long strings. [Ancak uzun dizilerde bu performansı gösteremezler.] Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), which are improved variants of RNN, are used as a solution to this problem. [Bu soruna çözüm olarak RNN'nin geliştirilmiş varyantları olan Uzun Kısa Süreli Bellek (LSTM) ve Kapılı Tekrarlayan Birimler (GRU) kullanılmaktadır.]

#### **How does RNN work? [RNN nasıl çalışır?]**

In an RNN, the words are converted into numeric vectors. [Bir RNN'de kelimeler sayısal vektörlere dönüştürülür.] Then the RNN processes each vector in the sequence sequentially. [Daha sonra RNN dizideki her vektörü sırayla işler.] It transmits the previous concealed state from one step to the next during processing. [İşleme sırasında önceki gizli durumu bir adımdan diğerine iletir.] The hidden state behaves as the neural network's memory. [Gizli durum, sinir ağının belleği gibi davranır.] It stores prior network data. [Önceki ağ verilerini depolar.]



# Illustrated Guide to Recurrent Neural Networks: Understanding the Intuition

<https://www.youtube.com/watch?v=LHXXI4-IEns>

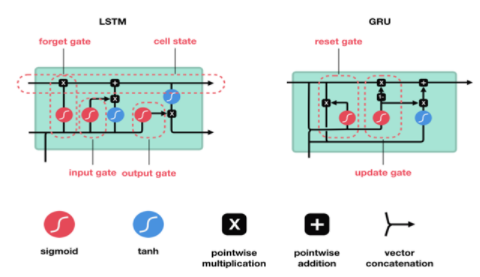
Q: What is Short Term Memory Problem? [S: Kısa Süreli Bellek Sorunu Nedir?] The vanishing and exploding gradients

A: **Short-term memory** is a problem for recurrent neural networks due to vanishing gradient issues. [C: Kısa süreli bellek, kaybolan gradyan sorunları nedeniyle tekrarlayan sinir ağları için bir sorundur.] They'll have difficulty transferring information from earlier time steps to later ones if the sequence is lengthy enough. [Eğer dizi yeterince uzunsa, bilgiyi önceki zaman adımlarından sonraki adımlara aktarmakta zorluk çekeceklerdir.] If you're attempting to predict anything from a paragraph of text, RNN’s may leave out essential information at the beginning of sequences. [Bir metin paragrafından herhangi bir şey tahmin etmeye çalışıyorsanız, RNN'ler dizilerin başında önemli bilgileri dışarıda bırakabilir.] The vanishing gradient problem affects recurrent neural networks during backpropagation. [Kaybolan gradyan sorunu, geri yayılım sırasında tekrarlayan sinir ağlarını etkiler.]

**The vanishing gradient problem**affects recurrent neural networks during backpropagation. [Kaybolan gradyan sorunu, geri yayılım sırasında tekrarlayan sinir ağlarını etkiler.] Gradients are values that are used to update the weights of a neural network. [Gradyanlar, bir sinir ağının ağırlıklarını güncellemek için kullanılan değerlerdir.] When a gradient diminishes as it backpropagates through time, this is known as the vanishing gradient issue. [Bir gradyan zamanla geri yayılırken azaldığında, bu kaybolan gradyan sorunu olarak bilinir.] When a gradient value falls below a certain threshold, it no longer contributes much to learning. [Bir gradyan değeri belirli bir eşiğin altına düştüğünde, artık öğrenmeye fazla bir katkı sağlamaz.]

- Interview Q&A

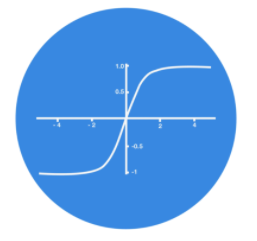
### **Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU)**



Internal processes called gates in the LSTM and GRU, regulate the flow of information. [LSTM ve GRU'da kapılar olarak adlandırılan dahili süreçler bilgi akışını düzenler.] These gates figure out which data in a sequence should be kept and which should be discarded. [Bu kapılar, bir dizideki hangi verilerin tutulması ve hangilerinin atılması gerektiğini belirler.] They can then send important information along long sequences to create predictions. [Ardından, tahminler oluşturmak için önemli bilgileri uzun diziler boyunca gönderebilirler.]

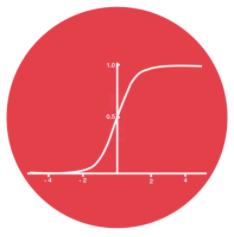
**Tanh Activations: [Tanh Aktivasyonları:]**

When vectors pass through a neural network, they go through a number of changes as a result of various math operations. [Vektörler bir sinir ağından geçtiğinde, çeşitli matematik işlemleri sonucunda bir takım değişikliklere uğrarlar.] As a result of these processes, the tanh activation is used so that some values don't become very large and other values don't become insignificant. [Bu işlemler sonucunda bazı değerlerin çok büyük, bazı değerlerin ise önemsiz hale gelmemesi için tanh aktivasyonu kullanılır.] Thus, all vector values stay between -1 and +1 values. [Böylece tüm vektör değerleri -1 ile +1 değerleri arasında kalır.]



**Sigmoid activations: [Sigmoid aktivasyonları:]**

Sigmoid activations are at the Gates. [Sigmoid aktivasyonları Gates'de.] Sigmoid activation is comparable to Tanh activation. [Sigmoid aktivasyonu, Tanh aktivasyonu ile karşılaştırılabilir.] Instead of squishing numbers ranging from -1 to 1, it squishes values ranging from 0 to 1. [-1'den 1'e kadar olan sayıları ezmek yerine, 0'dan 1'e kadar olan değerleri eziyor.] Because every integer multiplied by 0 equals 0, values vanish or are forgotten. [0 ile çarpılan her tamsayı 0'a eşit olduğundan, değerler kaybolur veya unutulur.] This is useful for updating or forgetting data. [Bu, verileri güncellemek veya unutmak için kullanışlıdır.]

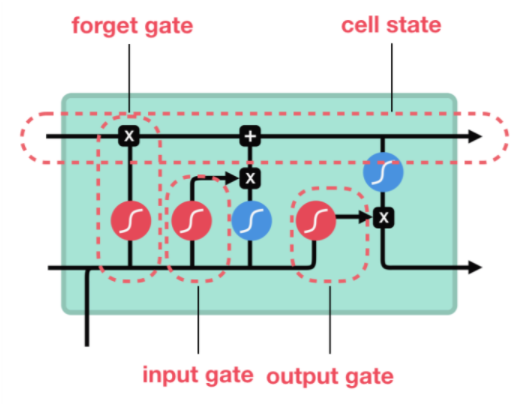


Q: What is the function of gates in LSTM and GRU? [S: LSTM ve GRU'da kapıların işlevi nedir?]

A: Internal processes called gates in the LSTM and GRU regulate the flow of information. [C: LSTM ve GRU'da kapılar olarak adlandırılan dahili süreçler bilgi akışını düzenler.] These gates figure out which data in a sequence should be kept and which should be discarded. [Bu kapılar, bir dizideki hangi verilerin tutulması ve hangilerinin atılması gerektiğini belirler.] They can then send important information along long sequences to create predictions. [Ardından, tahminler oluşturmak için önemli bilgileri uzun diziler boyunca gönderebilirler.]

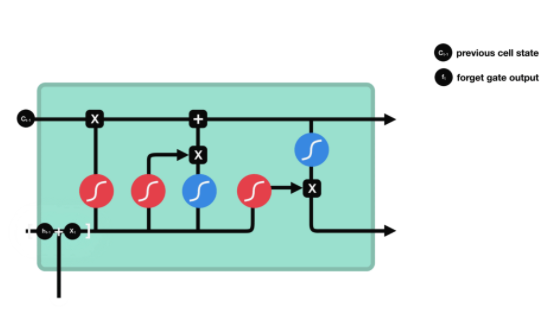
- Interview Q&A

### **Long Short-Term Memory (LSTM)**

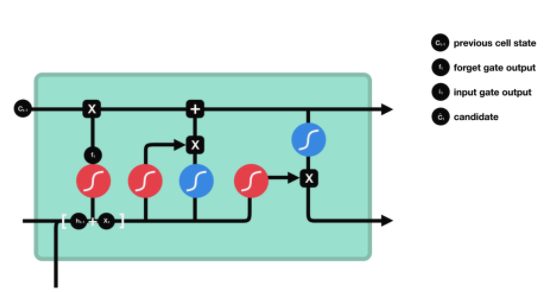


In an LSTM cell, there are three separate gates (forget gate, input gate, and output gate) that control information flow. [Bir LSTM hücresinde, bilgi akışını kontrol eden üç ayrı kapı (unut kapısı, giriş kapısı ve çıkış kapısı) vardır.] Let's get to know the forget gate. [Unutma kapısını tanıyalım.]

**Forget gate:** This gate determines whether or not to retain the information. [Kapıyı unut: Bu kapı, bilgilerin saklanıp saklanmayacağını belirler.] The sigmoid function is used for this process. [Bu işlem için sigmoid fonksiyonu kullanılır.] The closer the value is to 0, the more likely it will be forgotten, and the closer it is to 1, the more likely it will be retained. [Değer 0'a ne kadar yakınsa unutulma olasılığı o kadar yüksek ve 1'e ne kadar yakınsa akılda kalma olasılığı o kadar yüksek olacaktır.]

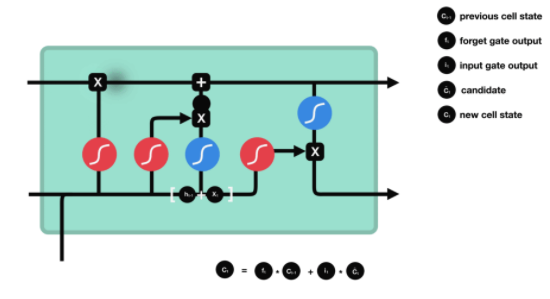


**Input gate:**In the first stage, the current input with the hidden state is passed through into the sigmoid function and then into the tanh function. [Giriş kapısı: İlk aşamada, gizli durumdaki mevcut giriş, sigmoid işlevine ve ardından tanh işlevine geçirilir.] Since the results from the sigmoid function will be between 0 and 1, it decides whether the information from the tanh function will be added to the cell state or not. [Sigmoid fonksiyonundan gelen sonuçlar 0 ile 1 arasında olacağından tanh fonksiyonundan gelen bilgilerin hücre durumuna eklenip eklenmeyeceğini belirler.]

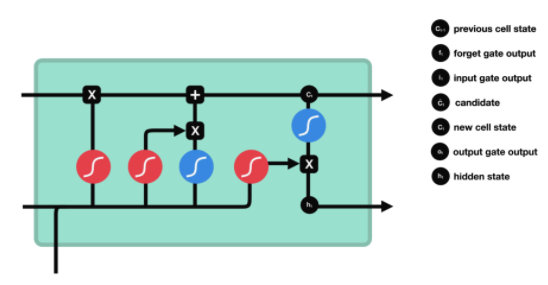


**Cell state:**The cell state carries relevant information throughout the sequence. [Hücre durumu: Hücre durumu, dizi boyunca ilgili bilgileri taşır.] Thus, previous information isn't forgotten and the model can make more accurate predictions. [Böylece önceki bilgiler unutulmamakta ve model daha doğru tahminler yapabilmektedir.] Therefore, you can think of it as the memory of the network. [Bu nedenle, bunu ağın hafızası olarak düşünebilirsiniz.]

As the cell state travels, information is added to or removed from the cell state via gates. [Hücre durumu seyahat ederken, kapılar aracılığıyla hücre durumuna bilgi eklenir veya hücre durumundan bilgi çıkarılır.] The gates are distinct neural networks that determine whether information about the cell state is permitted. [Kapılar, hücre durumuyla ilgili bilgilere izin verilip verilmediğini belirleyen farklı sinir ağlarıdır.] During training, the gates can choose which knowledge is necessary to retain and which to discard. [Eğitim sırasında kapılar, hangi bilginin muhafaza edilmesi ve hangilerinin atılması gerektiğini seçebilir.]



**Output gate:**It is used to determine the hidden state to be transferred to the next step. [Çıkış kapısı: Bir sonraki adıma aktarılacak gizli durumu belirlemek için kullanılır.] Since the hidden state contains information about previous inputs (words, letters, etc.), it is used for predictions that the model will make. [Gizli durum, önceki girdiler (kelimeler, harfler vb.) hakkında bilgi içerdiğinden, modelin yapacağı tahminler için kullanılır.] The current input and the previous hidden state are passed through into the sigmoid function. [Geçerli giriş ve önceki gizli durum, sigmoid işlevine aktarılır.] Then the current cell state is passed through into the tanh function and the results are multiplied. [Daha sonra mevcut hücre durumu tanh fonksiyonuna geçirilir ve sonuçlar çarpılır.] The output is now the new hidden state. [Çıktı şimdi yeni gizli durumdur.]



To summarize briefly, the forget gate decides whether to continue transferring the information from the previous steps to the cell state. [Kısaca özetlemek gerekirse, unut kapısı, önceki adımlardan hücre durumuna bilgi aktarmaya devam edip etmeyeceğine karar verir.] The input gate decides whether or not to add to the cell state a piece of new information. [Giriş kapısı, hücre durumuna bir parça yeni bilgi eklenip eklenmeyeceğine karar verir.] The output gate determines the next hidden state. [Çıkış kapısı, bir sonraki gizli durumu belirler.]

# Simple Explanation of LSTM | Deep Learning Tutorial 36 (Tensorflow, Keras & Python)

<https://www.youtube.com/watch?v=LfnrRPFhkuY>

Q: What is the function of Cell State in LSTM? [S: LSTM'de Hücre Durumunun işlevi nedir?]

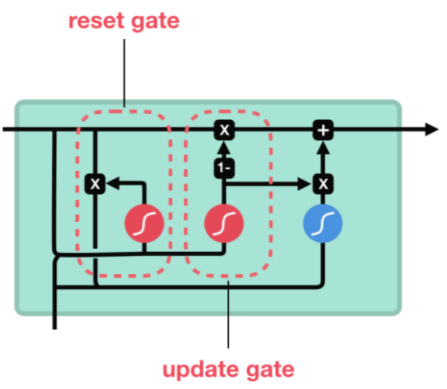
A : **The cell state**carries relevant information throughout the sequence. [A : Hücre durumu, dizi boyunca ilgili bilgileri taşır.] Thus, previous information isn't forgotten and the model can make more accurate predictions. [Böylece önceki bilgiler unutulmamakta ve model daha doğru tahminler yapabilmektedir.] Therefore, you can think of it as the memory of the network. [Bu nedenle, bunu ağın hafızası olarak düşünebilirsiniz.]

Q: What is the function of Forget Gate in LSTM? [S: LSTM'de Kapıyı Unut'un işlevi nedir?]

A: **The Forget Gate** determines whether or not to retain or discard the information. [C: Unutma Kapısı, bilgilerin saklanıp saklanmayacağını veya atılmayacağını belirler.] The sigmoid function is used for this process. [Bu işlem için sigmoid fonksiyonu kullanılır.] The closer the value is to 0, the more likely it will be forgotten, and the closer it is to 1, the more likely it will be retained. [Değer 0'a ne kadar yakınsa unutulma olasılığı o kadar yüksek ve 1'e ne kadar yakınsa akılda kalma olasılığı o kadar yüksek olacaktır.]

- Interview Q&A

### **Gated Recurrent Units (GRU)**



GRU emerged in 2014 and is the latest version of RNNs. [GRU 2014 yılında ortaya çıkmıştır ve RNN'lerin en son sürümüdür.] Its working logic is very similar to LSTM. [Çalışma mantığı LSTM'ye çok benzer.] Unlike LSTM, in GRU, cell state and hidden state are combined. [LSTM'den farklı olarak GRU'da hücre durumu ve gizli durum birleştirilir.] Also, it only has 2 gates. [Ayrıca sadece 2 kapısı var.] These are the update gate and the reset gate. [Bunlar güncelleme kapısı ve sıfırlama kapısıdır.] Let us now briefly examine them. [Şimdi bunları kısaca inceleyelim.]

**Update gate**: It decides what information to discard and what information to include, such as the forget and input gates in LSTM. [Güncelleme kapısı: LSTM'deki unutma ve giriş kapıları gibi hangi bilgilerin atılacağına ve hangi bilgilerin dahil edileceğine karar verir.]

**Reset gate**: It is the gate that decides how much of the information from other steps should be forgotten. [Reset gate: Diğer adımlardan gelen bilgilerin ne kadarının unutulacağına karar veren kapıdır.]

Although GRU is faster than LSTM, both LSTM, and GRU give very good results. [GRU, LSTM'den daha hızlı olmasına rağmen, hem LSTM hem de GRU çok iyi sonuçlar verir.] It's up to you to try both and decide. [İkisini de deneyip karar vermek size kalmış.]

# Simple Explanation of GRU (Gated Recurrent Units) | Deep Learning Tutorial 37 (Tensorflow & Python)

<https://www.youtube.com/watch?v=tOuXgORsXJ4>

Q: What is the function of Reset Gate in GRU? [S: GRU'da Kapıyı Sıfırla'nın işlevi nedir?]

A : It is the gate that decides how much of the information from other steps should be forgotten. [A : Diğer adımlardan gelen bilgilerin ne kadarının unutulacağına karar veren kapıdır.]

- Interview Q&A

### **Working Logic of LSTM and GRU [LSTM ve GRU'nun Çalışma Mantığı]**

Suppose the following comments are made on a shopping site. [Bir alışveriş sitesinde aşağıdaki yorumların yapıldığını varsayalım.]

**Negative Comments: [Olumsuz Yorumlar:]**

I've been saving for 6 months just to get this phone. [Bu telefonu almak için 6 aydır para biriktiriyorum.] I bought it very lovingly and willingly, but the product is **worthless [Çok severek ve isteyerek aldım ama ürün değersiz]**

The laptop that I dreamed about for months, arrived yesterday. [Aylardır hayalini kurduğum laptop dün elime ulaştı.] The shipping was very fast but the laptop was a **complete disappointment**. [Kargo çok hızlıydı ama laptop tam bir hayal kırıklığıydı.]

The phone is a game beast in terms of both screen quality and processor power, but the charging time is very bad. [Telefon hem ekran kalitesi hem de işlemci gücü açısından tam bir oyun canavarı ancak şarj süresi çok kötü.] You have to constantly charge. [Sürekli şarj etmeniz gerekiyor.] Frankly, I'm **not satisfied** at all [Açıkçası hiç memnun değilim]

**Positive Comments: [Olumlu Yorumlar:]**

Although the TV has minor shortcomings, I would **definitely recommend** it to everyone. [Televizyonun ufak tefek eksiklikleri olsa da kesinlikle herkese tavsiye ederim.]

**Awesome**, **a complete price performance product** that everyone should have at home [Müthiş, herkesin evinde olması gereken eksiksiz bir fiyat performans ürünü]

Shipping was very late but the product is **great. [Kargo çok geç geldi ama ürün güzel.]**

LSTM and GRU ignore all other words except underlined bold words. [LSTM ve GRU, altı çizili kalın sözcükler dışındaki tüm sözcükleri yok sayar.] The model understands that the comments made through underlined bold words are positive or negative. [Model, altı çizili kalın kelimelerle yapılan yorumların olumlu veya olumsuz olduğunu anlar.]

The model, **worthless**, **complete disappointment**, **not satisfied** etc. while learning that comments made with words are negative; **definitely recommend**, **awesome**, **a complete price performance product**, **great** etc. learns that comments made with words are positive. [Model, değersiz, tam bir hayal kırıklığı, tatmin olmama vb. kelimelerle yapılan yorumların olumsuz olduğunu öğrenirken; kesinlikle tavsiye ederim, harika, tam bir fiyat performans ürünü, harika vb. kelimelerle yapılan yorumların olumlu olduğunu öğrenir.] If we train our model  with the more comments , the more successful our model will yield. [Modelimizi daha fazla yorumla eğitirsek, modelimiz o kadar başarılı olur.]

# Illustrated Guide to LSTM's and GRU's: A step by step explanation

<https://www.youtube.com/watch?v=8HyCNIVRbSU>

Q: What is the working logic of LSTM and GRU ? [S: LSTM ve GRU'nun çalışma mantığı nedir?]

A: In NLP, LSTM and GRU determine which class the text belongs to by determining the keywords and patterns in the text. [C: NLP'de LSTM ve GRU, metindeki anahtar kelimeleri ve kalıpları belirleyerek metnin hangi sınıfa ait olduğunu belirler.] They ignore all other words, phrases, etc [Diğer tüm kelimeleri, cümleleri vb. görmezden gelirler.] .

- Interview Q&A