**T.C.**

**ATATÜRK ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BİLGİSAYAR MÜHENDİSLİĞİ BİTİRME PROJESİ DERSİ**

**YAPAY ZEKA İLE OTOMATİK SORU HAZIRLAMA**

**HAZIRLAYANLAR**

**Hilal Zehra AYDIN 190707078**

**PROJE DANIŞMANI**

**Dr.Öğr.Üyesi Mete YAĞANOĞLU**

**ERZURUM, OCAK 2023**

İçindekiler

[KISALTMALAR 4](#_Toc124106383)

[ŞEKİLLER DİZİNİ 5](#_Toc124106384)

[TABLOLAR DİZİNİ 6](#_Toc124106385)

[ÖZET 7](#_Toc124106386)

[1.GİRİŞ 8](#_Toc124106387)

[1.1.NLP 8](#_Toc124106388)

[1.2. NLP’NİN TARiHÇESİ 10](#_Toc124106389)

[1.3.NLP İle Otomatik Soru Hazırlama Projesine Giriş 10](#_Toc124106390)

[2.LİTERATÜR ARAŞTIRMASI 11](#_Toc124106391)

[3.PROJE 12](#_Toc124106392)

[A. KURAMSAL TEMELLER 12](#_Toc124106393)

[A.1 Yapay Sinir Ağları 12](#_Toc124106394)

[A.2.Uzun- Kısa Süreli Bellek (Long Short-Term Memory) 14](#_Toc124106395)

[A.3 Transformatörler 15](#_Toc124106396)

[A.5 BERT Algoritması 16](#_Toc124106397)

[A.6. T5: Metinden Metne Aktarım Transformatörü 17](#_Toc124106398)

[A.7. C4-Corpus 17](#_Toc124106399)

[A.8. Tokenization 18](#_Toc124106400)

[A.9. Projede Kullanılan Kütüphaneler 18](#_Toc124106401)

[A.9.1.Numpy 18](#_Toc124106402)

[A.9.2. Random 19](#_Toc124106403)

[A.9.3 Os 19](#_Toc124106404)

[A.9.4 PyTorch 19](#_Toc124106405)

[A.9.5 nltk 20](#_Toc124106406)

[A.9.7. String 21](#_Toc124106407)

[A.9.8 pke 21](#_Toc124106408)

[*A.9.10. flastext* 21](#_Toc124106409)

[A.9.11 gradio 22](#_Toc124106410)

[A.4.3 Sense2Vec 22](#_Toc124106411)

[A.4.3 SenseceTransformer 23](#_Toc124106412)

[A.4.3 NormalizedLevenshtein 23](#_Toc124106413)

[Textwrap3 24](#_Toc124106414)

[B. YÖNTEM 24](#_Toc124106415)

[B.1.Temel Cümleleri/Kavramları Belirlenmesi 24](#_Toc124106416)

[B.2.Anahtar Kelimelerin Belirlenmesi 24](#_Toc124106417)

[B.3.Çoktan Seçmeli Sorular Oluşturulması 25](#_Toc124106418)

[B.4.Bir NLP Probleminin Belirlenmesi 25](#_Toc124106419)

[B.4.1.Adım: Soyutlayıcı/Açıklayıcı Özetleme 25](#_Toc124106420)

[B.4.2.Adım: Cümlelerin Açıklanması 25](#_Toc124106421)

[B.4.3.Adım: Anahtar Kelimeler 25](#_Toc124106422)

[B.4.4.Adım: Soru Üretimi 26](#_Toc124106423)

[B.4.5.Adım: Yanlış Seçimler/Dikkat Dağıtıcılar Üretimi 26](#_Toc124106424)

[C. PROJENİN GERÇEKLENMESİ 26](#_Toc124106425)

[C.1.Programda modelin eğitimi için kullanılması istenen örnek metin girdileri 26](#_Toc124106426)

[C.2. Metin Özetleme için Gerekli veri setlerinin İndirilmesi 26](#_Toc124106427)

[C.3. Metinden Random Seçimler Yapma 27](#_Toc124106428)

[C.4. Kodlama - Kod Çözücülerin Kullanımı ile Veri Seti Eğitimi 27](#_Toc124106429)

[C.5. Yanlış Şık oluşturabilmek için Sözcük türetimi 28](#_Toc124106430)

[C.6. T5 ile Soru Üretimi 28](#_Toc124106431)

[C.7. Gradio ile Tasarım ve Projenin Web Sunumu İçin Kodlar 29](#_Toc124106432)

[D. PROJEDEN ELDE EDİLEN SONUÇLAR 30](#_Toc124106433)

[D.EKRAN ÇIKTILARI 30](#_Toc124106434)

[KAYNAKÇA 32](#_Toc124106435)

# KISALTMALAR

NLP: Natural Language Processing

WSD: Windows Vista Web Services on Devices

GPT-2: Generative Pre-trained

AllenNLP: Allen Natural Language Processing

BERT: Bidirectional Encoder Representations from Transformers

MCQ: Multiple Choice Questions

# ŞEKİLLER DİZİNİ

**Şekil1.1:** Doğal Dil İşleme Genel Yapısı…………………………………………………8

**Şekil 3.1:** Bir Yapay Sinir Ağı……………………………………………………………12

**Şekil 3.2:** Üç Girişli Yapay Nöron…………………………………………………………12

**Şekil 3.3:** Uzun Kısa Süreli Bellek Ünitesi…………………………………………………13

**Şekil 3.3:** Kodlayıcı-Kod Çözücü…………………………………………………………14

**Şekil A.5:** Dönüştürücü Model Mimarisi…………………………………………………15

**Şekil A.6:** **T5:** Metinden Metne Aktarım Transformatörü………………………………17

**Şekil A.8:** **T5:** Tokenizasyon Örneği………………………………………………………17

**Şekil A.9.6:**  Gereksiz Kelime Örneği……………………………………………………19

**Şekil A.9.10:**  Kelime Ayrıştırma …………………………………………………………20

**Şekil A.4.3 :** SentenceTransformers kitaplığının kullanım örnekleri 23

# TABLOLAR DİZİNİ

**Tablo C.1**.Örnek Metin Girdileri………………………………………………25

**Tablo C.2**.Metin Özetleme ……………………………………………………25

**Tablo C.3**.Random Secim………………………………………………………26

**Tablo C.4**.Metin Özetleme ……………………………………………………26

**Tablo C.5**.Sözcük Türetimi ………………………………………………………27

**Tablo C.6**.Soru Üretimi …………………………………………………………27

**Tablo C.7**.Gradio İle Proje Web Sunumu …………………………………………28

**Tablo D.1**. Gradio ile Web için Hazırlık……………………………………………28

**Tablo D.2.** Web Sayfasında Metin Girdisi ve Çıktı Sonucu-1…………………………29

**Tablo D.3.**Web Sayfasında Metin Girdisi ve Çıktı Sonucu-2(Devamı) …………………29

# ÖZET

Bu proje de metin üzerinden Yapay Zeka ile çoktan seçmeli sorular hazırlanmaktadır. İlk olarak doğal diş işleme teknikleri kullanılarak metinden sözdizimsel anlam analizine odaklanıldı. Tasarlanan sistemin temel amacı soru üretimi olsa da amaca ulaşmak için ilk olarak cümleler anlamlandırıldı ve anahtar kelimeleri bulundu. Bu anlam ve anahtar kelimeleri esas alarak metin dil bilgisi kurallarına göre özetlendi. Metin özetleme aşamasında herhangi bir cümleyi bölmek için AllenNLP' den faydalanıldı. Cümleler oluşturmak için GPT-2 modeli kullanıldı. Cümleler BERT ile filtrelendi. Özetlenen metin sayesinde programda soru oluşturma kalitesi artırıldı ve anahtar kelimelerden faydalanılarak özetlenmiş metinden sorular oluşturuldu.

Bu anahtar sözcükleri ayıklamak için Python Anahtar Sözcük ayıklama ve flashtext kitaplığı kullanıldı. Sözcüklerde ki anlam belirsizliğini gidermek (WSD) için BERT kullanıldı.Çoktan seçmeli sorular da (MCQ seçenekleri) çeldiriciler (yanlış şıklar) Wordnet, ConceptNet ve Sense2vec gibi çeşitli yaklaşımlar kullanılarak oluşturuldu. Doğru ve yanlış şıkları oluşturmak için yine BERT algoritması ve OpenAI, GPT-2 gibi önceden eğitilmiş modeller kullanıldı.MCQ için çeldiriciler (yanlış seçimler) oluşturma aşamasında iken HuggingFace kitaplığını kullanarak ve bir T5 trafo modelini eğiterek herhangi bir içerikten MCQ'lar oluşturuldu. T5 dönüştürücü algoritması, HuggingFace ve Transformers kitaplıklarıyla Pytorch Lightning soru oluşturma modelini eğitmek için SQUAD veri seti projede kullanıldı. T5 modeliyle ONNX formatına dönüştürülerek program yayınlandı. Transformatör modeli için FastAPI kullanılarak projenin web bağlantısı sağlandı.

Bu bağlantıyla web sayfasında girilen İngilizce metin Sense2vec veya Wordnet yaklaşımlarından herhangi biri seçilerek çoktan seçmeli soruların oluşturulması sağlandı.

Farklı modellerin kullanılmasına olanak sağlayan proje soru şıkları metin içeriğindeki anahtar kelimelere odaklanarak doğru seçeneğin bulunmasına yöneliktir.

Yeni metin içeriği ile farklı sorular oluşturmak için ise metin temizleme butonundan ve ayrıca eklenen metnin etiketlenip projenin veri listesine eklenebilmesi ve tekrar kullanılması için etiketle butonundan faydalanıldı. Sorular oluşturulurken anahtar sözcüklerin rastgele seçilmesi metinden farklı soruların oluşturulmasına olanak sağladı. Bu sayede daha dinamik ve kullanıcıya yönelik yapay zeka ile soru oluşturma otomatı projesi hazırlandı.

# 1.GİRİŞ

# 1.1.NLP

İnsanların kullandığı Türkçe, İngilizce, Rusça, Almanca gibi diller doğal dil olarak tanımlanır. Bilgisayarlar ise makine dili olarak bilinen programlama dilleri kullanır. Programlama dilleri kelimelerden değil “0” ve “1” sayılarının binlerce farklı kombinasyonundan oluşur. İnsan dilini bilgisayarın anlayabileceği formata dönüştürerek insan ile bilgisayar arasında bağ kurulmasını sağlayan Doğal Dil İşleme, yapay zekanın bir alt dalı olarak tanımlanır. Yapay Zeka Doğal Dil İşleme tekniğini kullanarak bilgisayarın insan dilini anlayıp yorum yapabilmesini sağlar. Makinelerin insanları anlamaları, NLP’nin hedeflerinden biridir. Geleceğin en önemli sektörlerinden biri olarak görülen yapay zekânın insanla verimli iletişim kurması için dil vazgeçilmez unsurdur. Bu noktada ilgili teknolojinin yapay zekâya büyük katkıları olduğu söylenebilir. NLP, zekânın üst düzeyde kullanımını ve kaliteyi konu alır. İnsanların ve kuruluşların, nasıl olup da sıra dışı sonuçlar almayı başarabildiklerini inceler.

Kişilerin ve kuruluşların kullandıkları yöntemler, başkalarına da öğretilebilir ve eğer öğretilen kişiler öğrendiklerinin ne olduğunun farkına varırlarsa, benzer sonuçları alabilirler. Bu öğrenme, uygulama ve benzer sonuçlar alma sürecine “modelleme” adını veriyoruz. Herhangi bir başarının modellemesini yapılabilmemiz için, NLP, kendimize ait olan deneyimleri nasıl yapılandırdığımızı inceler. Değerlerimizi ve inançlarımızı nasıl algıladığımız, onlar hakkında neler düşündüğümüz, duygusal hâllerimizi ve tutumumuzu nasıl şekillendirdiğimiz üzerinde durur. Yaşadığımız deneyimlerle iç dünyamızı nasıl oluşturduğumuzu ve nasıl anlamlandırdığımızı anlamaya gayret eder. Hiçbir olayın kendi başına bir anlamı yoktur. Olayları anlamlandıran bizleriz. Dolayısıyla, aynı olay, çeşitli kişiler için farklı anlamlar taşıyabilir. Bu yüzden NLP, deneyimleri içeriden anlamaya ve onları iç dünyamızda nasıl anlamlandırdığımızı keşfetmeye çalışır.

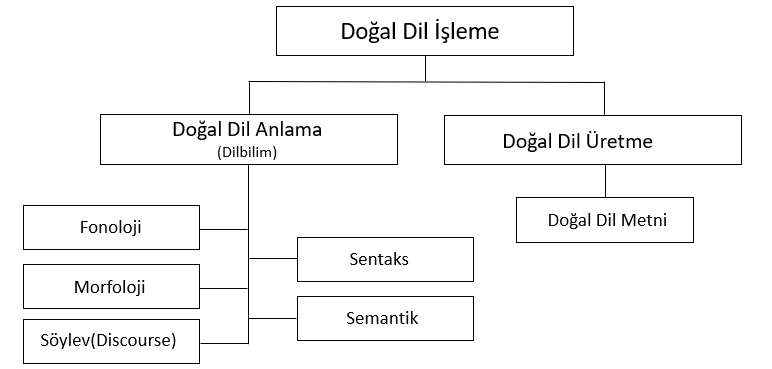
NLP işe, dünya üzerindeki en iyi iletişimcilerin çalışmalarının incelenmesiyle başlamış, sonrasında insanlar arası iletişimi sistematize edilmiş bir bilgi dalı hâline getiren bir çalışmaya dönüşmüş ve bu çalışmaya sıra dışı kişileri modelleyerek ve pratik araçlar ile bazı yöntemler ekleyerek gelişmesine devam etmiştir. Söz konusu araçlar, dünyanın her yerinde; sporda, iş dünyasında, askerî eğitimlerde, satış sektöründe, hukuk ve eğitim alanlarında kullanılmaktadırlar. NLP sadece bir teknikler demeti değil, aynı zamanda bir düşünce tarzıdır; merak duygusuna ve eğlenceye dayanan zihinsel bir çerçevedir. NLP, vizyonunuz, amacınız ve değerleriniz hakkında netlik kazandırır. Genellikle kişisel gelişim araçları ve kursları, yalnızca hedeflerin başarılı bir şekilde nasıl belirleneceğine odaklanır. NLP ise, dilde ve duyularınızla ifade ettiğiniz bilinçaltı kalıplarınızdan haberdar olmanıza ve gerektiğinde bu kalıpları değiştirmenize yardımcı olur.

Doğal Dil İşleme (Natural Language Processing), terimlerden bağımsız şekilde günlük hayatta kullanılan dili işler ve konuşulan kelimeleri yapılandırılmış verilere dönüştürür. Bu veriler genellikle metin, resim ve video gibi bilgi kaynaklarıdır. NLP, bu verileri işlemek ve anlayabilmek için özel algoritmalardan yararlanır. NLP’nin esasında kullanılamayacağı alan yoktur. Çünkü odak insanın kendisi yani zihnidir. İçinde bulunduğu hayatı, yaşadığı problemleri, sorunlarını değiştirmek isteyen herkes, hangi alanda ya da meslekte olursa olsun kullanılır. Çoğu kişi farkında olmadan günlük hayatında doğal dil işleme ile etkileşime geçmiştir. Yapay zeka, veri bilimi, bilgisayar bilimi ve dilbilimin bir araya geldiği multidisipliner bir yapıya sahip olan Doğal Dil İşleme kullanım alanları oldukça geniştir. Bankacılık, sigortacılık, sağlık, reklam, halkla ilişkiler, yayıncılık gibi insan faktörünün ön planda olduğu pek çok sektörde kullanılabilir. Örneğin Oracle Dijital Asistan (ODA), Siri, Cortana veya Alexa gibi sanal asistanların ardındaki temel teknoloji NLP'dir. Bu sanal asistanlara iletişime geçtiğimiz de asistanların hem kullanıcının talebini anlayıp hem de doğal bir dille yanıt vermesini sağlayan NLP'dir. NLP hem yazılı metinler için hem de konuşma için geçerlidir ve tüm dillerde uygulanabilir. NLP destekli araçlara ilişkin diğer örnekler arasında web arama, istenmeyen e-posta filtreleri, otomatik metin veya konuşma çevirisi, belge özetleme, duygu analizi ve dil bilgisi/yazım denetimi yer alır. Örneğin, kimi e-posta programları mesaj içeriğine göre uygun yanıt önerilerinde bulunabilir. Bu programlar, mesajınızı okumak, analiz etmek ve yanıtlamak için NLP teknolojisinden faydalanır. NLP çok çeşitli iş süreçlerinde, özellikle e-postalar, anketler, sosyal medya konuşmaları vb. gibi yüksek hacimli yapılandırılmamış metinlerde basitleştirme ve otomatikleştirme sağlar. İşletmeler NLP sayesinde verilerini daha iyi analiz ederek daha kısa sürede daha doğru kararlar alabilir.

Bilişimsel dilbilimi ile bilgisayar bilimini bir araya getirerek insan-makine iletişimini gerçekleştiren NLP, söz dizimi ve anlamsal analiz olmak üzere iki ana teknik kullanır. Bir cümle içerisinde yer alan kelimelerin dilbilgisine uygun olarak düzenlenmesine söz dizimi denir. Anlamsal analiz ise kelimelerin anlamlarının gruplanarak incelenmesidir. Doğal Dil İşleme teknolojisi, bu iki tekniği kullanarak cümlelerin yapılarını ve anlamlarını analiz eder. Ancak konuşma ve yazı dilinin analiz edilebilmesi karmaşık ve zor bir süreçtir.

Her dilin kendi içinde bir gramer kuralları ve kelime çeşitliliği vardır dolayısıyla doğal dil işlemenin karmaşıklığı buranda anlaşılabilir. Ham olan verinin makine için anlamlı olmadığı aşikardır. Metin içerisinde ki kelimelerin anlamlandırılması için çeşitli işlemlerden geçmektedir. Temelde gelinmek istenen nokta insanlar tarafından söylenen cümlelerin parçalara ayrılması, anlaşılması ve insanların anlayacağı bir şekilde cevaplanması sürecidir.

Metinler bilgisayar tarafından anlaşılmak için bir dizi süreçten geçmektedir. Örneğin gürültü giderme (noise removel), sözlük normalize edilmesi (lexicon normalization), nesne standart oluşumu (object standarization) gibi teknikler örnek verilebilir. Bu süreci daha basit şekilde anlatırsak doğal dil işlemede doğal dilin anlaşılması ve üretilmesi olarak ikiye ayrılır. Doğal dilin anlaşılması için fonoloji, morfoloji, söylev, sentaks, semantik aşamaları kullanılır.



**Şekil1.1:** Doğal Dil İşleme Genel Yapısı

# 1.2. NLP’NİN TARiHÇESİ

Nöro Linguistik Programlama (Beynimizi Doğru ve Etkin Kullanma sanatı) yani NLP,1970’li yılların başında, Richard Bandler ve John Grinder tarafından kurulmuştur. Bandler, bu yıllarda Kaliforniya Üniversitesinde, Matematik öğrencisi iken Psikolojiye ve bilgisayar mühendisliğine karşı özel bir ilgi duymaktaydı. Bandler, öğrenciliği sırasında NLP’nin diğer kurucusu dilbilim profesörü John Grinder ile tanıştı. Grinder, 1960’lı yılların başında San Fransisco Üniversitesinde Psikoloji eğitimi almıştı. Dil öğrenme yeteneğinden dolayı, bir süre Amarikan İstihbaratında ajan olarak çalıştı. 60’lı yılların sonunda Kaliforniya Üniversitesinde Dilbilim eğitimi alarak bu alanda doktora yaptı. Grinder, bir dilbilimci olarak daha çok dilin sentaktik (sözdizimsel) alanında ve Noam Chomsky’nin transformasyonel (geçişli) gramer teorileri alanında çalışmalar yaptı. Bandler ile Grinder, kendi birikimlerini ve yeteneklerini birleştirerek Gestalt terapisinin kurucusu Fritz Perls‘ün, aile terapisinin anası Virginia Satir’in ve modern Hipnozun babası Milton Ericson’un insanlar üzerinde nasıl değişimler meydana getirdiklerini inceleyerek, onların çalışma tekniklerini ortaya çıkardılar. Böylelikle, NLP ilk olarak modelleme çalışmasıyla ortaya çıkmış oldu.

NLP, bugün dünyada gitgide popülerleşen ve eğitim, iş dünyası, psikoterapi ve kişisel gelişim alanlarında kullanılan bir değişim ve gelişim sistemidir.

#### 

# 1.3.NLP İle Otomatik Soru Hazırlama Projesine Giriş

Doğal dil işleme ile otomatik yanıtlama, özetleme, benzerlik, sınıflandırma, yapay metin üretme alanlarını kullanarak hayat kolaylaştıran bir proje yapılması mümkün. Tüm bu alanları bir arada kullanarak girilen herhangi metinde ki cümlelerin ayrıştırılıp sözcüklere dönüştürülmesiyle gereksiz kelimeler atılır. Cümle içinde ki anahtar kelimelerin bulunması sağlanabilir. Buradan benzer tekniklerle cümle özetlenip, özet cümle üzerinden çalışmalar devam edilebilir. Doğal dil işleme insan dili ile uğraştığından her dilin kendine has gramerini dikkate almak zorundadır. Daha önceki veri setinden faydalanılarak istenen uygulama eğitilebilir. Bu sayede daha öncesinde hali hazırda sonradan eklenen metinlerde daha önceden eğitilmiş olan uygulama daha rahat işlem yapma gücüne sahip olur. Aynı zamanda daha önceden eğitilmiş modelleri kullanarak metin özetleme de ve alınan metnin işlenmesinde kolaylık sağlanır. Metin üzerinde ki çalışmalar doğal dil işlemenin ilgilendiği kapsamlı bir alandır. Metin işlenip anlamlandırıldığında yani bilgisayarın anladığı bir hale getirildiğinde yapılmak istenen işleme geçilebilmektedir. Otomatik olarak bu işlemlerin gerçekleşmesi ile metinlerin işlenmesi ve geri dönüt olarak yeni bir metnin oluşturulması sağlanır. Aynı zamanda dönüt metnin soru olarak oluşturulması yeni bir çalışma alanını ortaya koyulmasını sağlar. Oluşan sorunun cevaplarının doğru ve yanlış olarak oluşturulması yani çoktan seçmeli olarak soruların hazırlanması mümkün. Bu noktada otomatik olarak soru hazırlama uygulamaları üzerinde çalışmalar devam etmektedir.

Otomatik soru oluşturma projesi, Doğal Dil İşlemenin (NLP) bir parçasıdır. Bir metni girdi olarak alabilen ve okuyucunun metindeki bilgilere ilişkin bilgisini değerlendirmek için çıktı soruları üretebilen bir sistemdir. Birçok araştırmacının çalışmalarını sunduğu bir araştırma alanıdır ve daha yüksek doğruluk elde etmek için halen araştırılan bir alandır.  NLP aracılığıyla otomatik soru oluşturma alanında çok sayıda teknik ve model geliştirilmiştir. Birçok dilde çalışmalar yapılmıştır.

Günümüzde öğretmenler (akademisyenler) test sorularını ve sınavları oluşturmak için çok zaman harcıyorlar. Sınav tarzı sorular, çeşitli amaçlara hizmet eden temel bir eğitim aracı olsa bile soruların manuel olarak oluşturulması, eğitim, deneyim ve kaynak gerektiren karmaşık bir süreçtir. Buda, geniş bir soru havuzu gerektiren test hazırlama sürecinin gelişimini engeller ve yavaşlatır. Soruların manuel olarak oluşturulmasıyla ilgili masrafları azaltmak ve sürekli yeni soru sağlama ihtiyacını karşılamak ve aynı zamanda her türlü bağımlılığı ortadan kaldırmaya ve zaman kaybını gidermeye yardımcı olabilecek bir yapay zeka uygulaması günümüzde ve gelecekte ki teknoloji seviyesinde olması gereken bir kolaylıktır. Bu sayede öğrenmek isteyen her türlü öğrencinin kullandığı materyallerde ki metinleri girdi olarak verebilir ve bu temelde, kendi kendini analiz (öz kalibrasyon) yapabilecekleri cevapları olan bir dizi soru almaları sağlanılabilir. Ayrıca soruların rastgele ve tekrarsız olması projenin daha esnek ve dinamik olmasını sağlamaktadır.

# 2.LİTERATÜR ARAŞTIRMASI

Web üzerinden otomatik soru hazırlama uygulamalarında Türkiye’de daha çok göze çarpan otomasyonların kullanılmasıdır. İstanbul ve Sakarya Üniversitelerin de bu konu üzerinde daha önce yazılan tezde otomasyon üzerinden web tabanlı otomatik soru hazırlama projesini görmekteyiz. Nitekim çoğu üniversitede daha çok otomasyon üzerine yoğunlaşılmış. Benim projemde daha gelişmiş bir teknoloji olan yapay zeka kullanıldı. Projem gelişen yapay zeka teknolojilerinin takibinde daha iyi sonular hedeflemeye hazırdır. Daha yaygın olarak bu proje ile ilgili Python ile quiz çalışmalarının yapılmaktadır. Bu çalışmalarda bir dosya üzerinde ki verilerden faydalanılmaktadır. Benim projemde daha geniş bir veri seti ve Google destekli kütüphaneler kullanılmaktadır. Bu sayede metinler daha iyi analiz edilip sonuçlar da beklenen kaliteye ulaşabilmektedir.

Yapay zekada bu konuda ticari çalışmalar mevcuttur. Ticari amaç güden bazı test hazırlama siteleri vardır. Bu konu üzerinde hala çalışmalar devam etmektedir. Soru kalitelerin ve metin anlamlandırmanın daha iyi yapılması için gerekli çalışmalar elbette henüz son bulmamıştır. Çünkü dünyada birçok dil ve buna nazaran gramer vardır. Her dilin kendine has gramerini programlama ile bilgisayara anlamlandırmak hayli uzun ve üzerinde çalışılması gereken bir süreçtir. Şu anda Python ile en iyi desteklenen ve sonuçları en iyi alınan dil dünya dili olan İngilizcedir.

Soru hazırlama alanında en iyi başarının elde edildiği alan soru cevap sistemleridir. Soru hazırlama sistemlerinin çoğunda SQuAd veri kümesidir. Bende bu veri kümesini kendi projemde kullandım. Soru cevap sistemlerinde otomatik yanıtlar almak çok daha kolaydır. NewsQA gibi veri kümelerinde binden fazla soru cevap çiftleri bulunmaktadır. Benim projemde sorular bir veri setinden değil metin anlamlandırması ile algoritmanın birleşimi sayesinde sorulmaktadır. Bu tarz sistemler yanlış cevaplara ihtiyaç duymazlar. Tek doğru cevap ya da benzer doğru cevaplar olabilir. Benim projemde ise doğru cevap soruya göre şekillenir ve bir tanedir. Bunun yanın da kelimelerin anlamlandırılması ile ona yakın ama doğru olmayan şıklarda oluşturulmaktadır. Çeldiriciler ile kullanıcının kendi bilgisini test etmesi amaçlanmaktadır. Bu gerçekleştirilen soru cevap sistemlerine örnek olarak Google, Yandex, Siri, Cortana gibi kişisel asistan uygulamaları verilebilirler.

Günümüz de hemen hemen her uygulama ve web sitesi online yardım robotları denilen otomatik olarak daha önceden oluşan sorunların yardımında farklı kişilere aynı sorunlarda yardımcı olma kolaylığı sağlamak için önceden bu oluşan sorunlar için hazırlanmış veri setleri ile yazılan soruya otomatik yanıt verilmektedir. Hala gelişmekte olan bu yardım botları çok daha farklı metinlerde yardım sunamamaktadır. Benim projem çok daha kapsamlı birbirinden alakasız metinlerde bilgisayar dilinin anlayacağı şekilde anlamlandırmayı rahat yapabilmektedir. Projemde birçok teknolojiyi bir araya getirdim ve bu sayede metin özetlemeyi özetlenen metinden soru hazırlamayı daha rahat sunmaktadır. Girilen aynı metinlerde aynı soru ve cevapları değil her seferinde farklılaştırarak soru hazırlamaktadır.

Günümüzde Projemde bir araya getirdiğim algoritmalar ayrı olarak bir proje konusudur. Başarılı olan bazı proje Örnekleri metin sınıflandırma ve kategorizasyon, anlamsal ayrıştırma ve soru cevaplama, bölüm etiketleme, dil üretimi ve çok belgeli özetleme örnek verilebilir.

Belli sitelerin hizmetlerini verilerini kullanan ve buna yönelik soru cevap oluşturulan projelerde yaygın. Hatta daha çok böyle tezlere rastlamakta mümkün. Örneğin IBM Wotson’ın NLP hizmetleri kullanılarak oluşturulan soru yanıtlama projesi buna örnek verilebilir.

Uzun ve karmaşık sorular oluşturabilmek için makine öğrenimin yanı sıra metin ve sorulardan oluşan bir veri kümesi kullanılmaktadır. Benim projemde hazır metin veri seti bulunmakta ve soru oluşturmak için eğitilmiş model kullanılmaktadır.

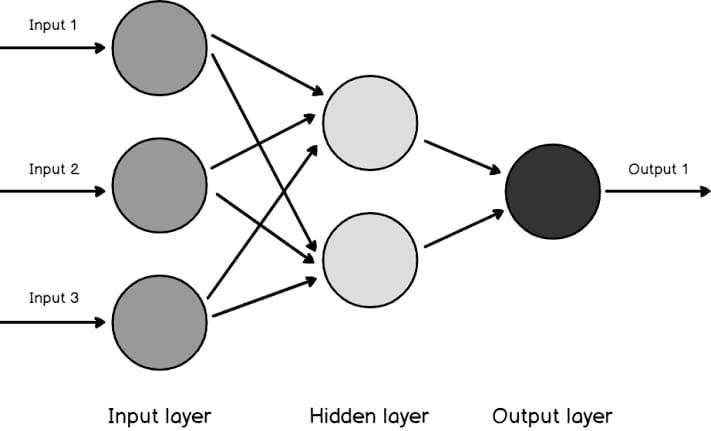
Daha önce yapılmış bir çalışma olan COVID-19 ile ilgili soruları yanıtlamak için API Webapp ile NLP ile güvenilir veri kaynaklarından soru hazırlama projesi geliştirilmiştir. Bu ve benzeri projelere sık rastlanmaktadır. Bu tarz projelerde docker ile url türetilip siteye bu şekilde ulaşım sağlanamkta. Ben projemde daha çok makine öğrenmesi ve yapay zeka projelerini yayınlamak için kullanışlı olan gradio kullandım. Gradionun oluşturduğu url ve space de Python projemi gradio kullanarak yayınlaması ve ulaşımı bu sayede daha kolay hale geldi.

# 3.PROJE

## KURAMSAL TEMELLER

# A.1 Yapay Sinir Ağları

Yapay sinir ağları, örnekleri dikkate alarak sınıflandırma, tahmin, karar verme, görselleştirme ve başka görevleri gerçekleştirmeyi öğrenebilen, beynin nöral yapısına dayanan, hesaplamalı ve doğrusal olmayan bir modeldir. Makine çevirisinden arama motorlarına ve uygulamalardan bilgisayarlar asistanlarına kadar derin öğrenmede için sinir ağları kullanılır. Bir yapay sinir ağı, yapay nöronlardan veya işleme elemanlarından oluşur. Birbirine bağlı üç katmanda düzenlenir. Bunlar: girdi, birden fazla katmana sahip olabilen gizli ve çıktıdır.

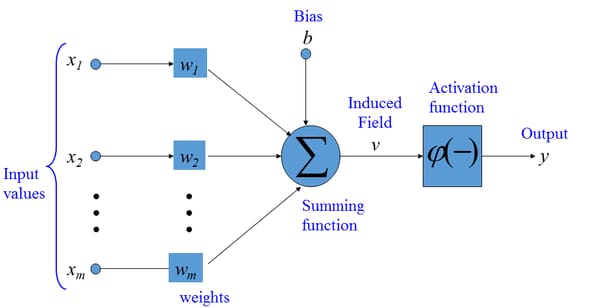


**Şekil 3.1:** Bir Yapay Sinir Ağı

Giriş katmanı, gizli katmana bilgi gönderen giriş nöronlarını içerir. Gizli katman, verileri çıkış katmanına gönderir. Her nöronun ağırlıklı girdileri (sinapslar), bir aktivasyon fonksiyonu (bir girdi verilen çıktıyı tanımlar) ve bir çıktısı vardır. Sinapslar, bir sinir ağını parametre haline getirilmiş sisteme dönüştüren ayarlanabilir parametrelerdir. Girdilerin ağırlıklı toplamı, nörondan bir çıktı elde etmek için aktivasyon fonksiyonuna iletilen aktivasyon sinyalini üretir. Yaygın olarak kullanılan aktivasyon fonksiyonları doğrusal, adım, sigmoid, tarih ve düzeltilmiş doğrusal birim fonksiyonlarıdır.

Bir diğer sinir ağı modeli ileri beslemeli sinir ağı modelidir. İleri beslemeli sinir ağları, düğümlerin döngü oluşturmadığı yapay sinir ağlarıdır. Bu tür bir sinir ağı aynı zamanda çok katmanlı bir sinir ağı olarak da bilinir, çünkü tüm bilgiler yalnızca ileri iletilir.

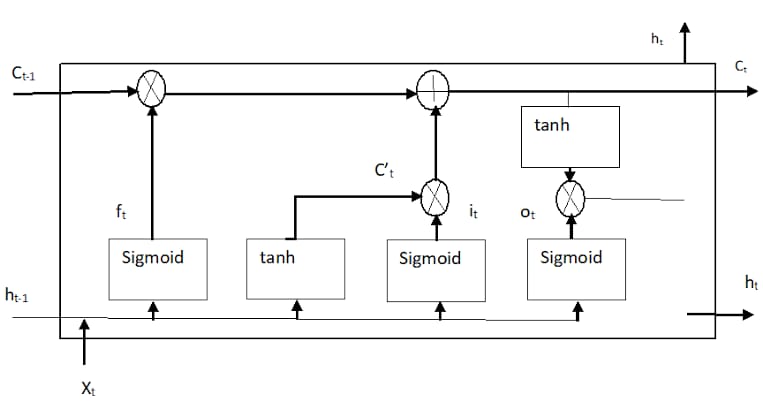
Veri akışı sırasında, giriş düğümleri, gizli katmanlardan geçen ve çıkış düğümlerinden çıkan verileri alır. Ağda, çıkış düğümünden bilgi göndererek alışılabilecek hiçbir bağlantı yoktur. Temel bir sinir ağı türüdür çünkü girdi yalnızca bir yönde işlenir. İleri beslemeli sinir ağları, Google Fotoğraflar uygulamasında gösterildiği gibi, fotoğraflarda nesne algılamanın temelini oluşturur.



**Şekil 3.2:** Üç Girişli Yapay Nöron

# A.2.Uzun- Kısa Süreli Bellek (Long Short-Term Memory)

Derin öğrenme alanında kullanılan yapay bir yinelemeli sinir ağı (RNN) mimarisidir. Standart ileri beslemeli sinir ağlarının aksine, LSTM'nin geri bildirim bağlantıları vardır. Yalnızca anlık veriyi (resim gibi) değil, veri dizilerini (konuşma veya video gibi) de işleyebilir. Örneğin, LSTM bölümlenmemiş, bağlı el yazısı tanıma, konuşma tanıma ve ağ trafiğinde anomali veya saldırı tespit sistemleri tespiti gibi görevler için geçerlidir. LSTM ağları, zaman serisi verilerine dayanarak sınıflandırmak, işlemek ve tahminler yapmak için çok uygundur, çünkü bir zaman serisindeki önemli olaylar arasında bilinmeyen süreli gecikmeler olabilir. LSTM'ler, geleneksel RNN'leri eğitirken karşılaşılabilecek patlayan ve yok olan gradyan problemleriyle başa çıkmak için geliştirilmiştir.



**Şekil 3.3:** Uzun Kısa Süreli Bellek Ünitesi

LSTM dört farklı katmana dayanır. Bunlar:

* σ: bir sigmoid fonksiyonudur.
* tanh bir tanh işlevidir.
* + ve x, sırasıyla toplamı ve çarpımı temsil eden işlenenlerdir.
* C: hücre durumundaki değerlerdir (uzun süreli bellek).
* h: gizli durumdaki değerlerdir (kısa süreli bellek).
* x: giriş değerleridir.
* W: ağırlıklardır.
* b: önyargılardır.

LSTM, dört sinir ağı ve hücre adı verilen farklı bellek blokları içeren bir zincir yapısına sahiptir. Bilgi hücreler tarafından tutulur ve hafıza manipülasyonları kapılar tarafından yapılır. Üç kapı vardır. Bunlar:

1. Unut Kapısı: Hücre durumunda artık yararlı olmayan bilgiler unut kapısı ile kaldırılır. İki girişi vardır. X-t (belirli bir zamanda giriş) ve h\_t-1 (önceki hücre çıkışı) kapıyla beslenir ve ağırlık matrisleri ile çarpılır ve ardından önyargı eklenir. Sonuç, ikili çıktı veren bir etkinleştirme işlevinden geçirilir. Belirli bir hücre durumu için çıktı 0 ise, bilgi parçası unutulur ve çıktı 1 için bilgiler ileride kullanılmak üzere saklanır. Aşağıdaki formül ile özetlenebilir:

ft = σ(Wf · [ht−1,xt] + bf)

1. Giriş kapısı: Hücre durumuna yararlı bilgilerin eklenmesi giriş kapısı tarafından yapılır. İlk olarak, bilgi sigmoid işlevi kullanılarak düzenlenir ve h\_t-1 ve X-t girişlerini kullanarak unutma kapısına benzer hatırlanacak değerleri filtreler. Daha sonra, h\_t-1 ve X-t'den olası tüm değerleri içeren -1'den +1'e bir çıktı veren tan(h) işlevi kullanılarak bir vektör oluşturulur. X-t. Sonunda, yararlı bilgileri elde etmek için vektörün değerleri ve düzenlenmiş değerler çarpılır. Formülden çıkan 2 değer bir araya getirilirse ve durumdaki güncellemeyi tanımlayacaktır:

it = σ(Wi · [ht−1,xt] + bi)

C˜t = tanh(WC · [ht−1,xt] + bC)

1. Çıkış kapısı: Çıkış olarak sunulacak mevcut hücre durumundan faydalı bilgiler çıkarma görevi çıkış kapısı tarafından yapılır. İlk olarak, hücre üzerinde tan(h) fonksiyonu uygulanarak bir vektör oluşturulur. Daha sonra bilgiler sigmoid fonksiyonu kullanılarak düzenlenir ve h\_t-1 ve X-t girişleri kullanılarak hatırlanacak değerlere göre filtrelenir. Son olarak vektörün değerleri ve regüle edilen değerler çarpılarak bir sonraki hücreye çıktı ve girdi olarak gönderilir.

Ct = ft ∗ Ct−1 + it ∗ C˜t

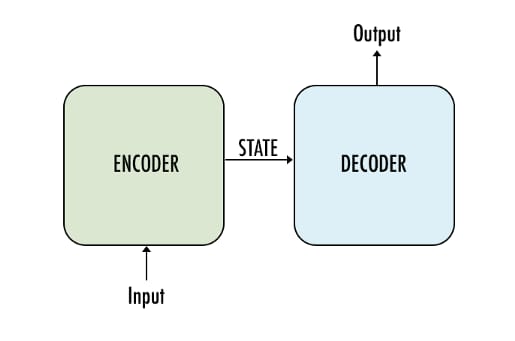
ot = σ(Wo · [ht−1,xt] + bo)

ht = ot ∗ tanh(Ct)

# A.3 Transformatörler

LSTM birimleri, Makine Çevirisi veya Soru Oluşturma gibi Diziden Diziye tahmin problemlerinde yaygın olarak kullanılır. Google tarafından 2014 yılında ilk kez tanıtılan Diziden Diziye modeller farklı mimariler üzerine kurulabilir, ancak çok etkili olduğu kanıtlanan bir yaklaşım Kodlayıcı-Kod Çözücüdür.

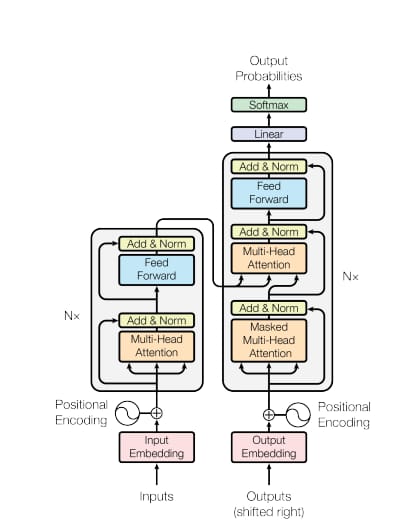
Kodlayıcı-Kod Çözücü RNN bileşeninden oluşan bir öğrenme modelidir. Kodlayıcının görevi, bir giriş dizisinin her belirtecini işlemek ve ara durum olacak değerler vektörü oluşturmak için kodlamaktır. Bağlam vektörü olarak da bilinen bu vektör, tüm girdi öğelerinin tüm anlamının kapsüllenmesine yardımcı olur ve kod çözücüyle doğru tahminlerde bulunabilmesi için mümkün olduğunca fazla bilgi verecek şekilde oluşturulmuştur; Kod çözücü, aslında kodlayıcının son durumlarıyla beslendikten sonra kod çözücüyü, çıktı dizisini oluşturmak için işler.



**Şekil 3.3:** Kodlayıcı-Kod Çözücü

# A.5 BERT Algoritması

Metni özetleme sürecinde "bert-extractive-summarizer" dan faydalanılır. Bu model BERT algoritmasına dayanıyor. BERT Algoritması Google’ın diğer birçok algoritma güncellemeleri gibi sorguları daha iyi anlamak ve kullanıcılarına daha doğru sonuçlar sunmak adına geliştirilmiştir. “Bidirectional Encoder Representations from Transformers” ifadelerinin baş harflerinden oluşan BERT algoritması, yapay zeka ve makine öğrenimi teknolojilerini bir arada kullanan bir doğal dil işleme tekniği olarak açıklanabilir. Basitçe söylemek gerekirse BERT robotik bir sinir ağıdır. BERT, doğal dil işlemenin bir ön eğitim modelidir. Bu, model veri setinin çeşitli sistemler geliştirmesi için kullanılır. NLP görevleri için etiketli metinler kullanılarak ince ayar yapmak için tasarlanmıştır. Google tarafından 2018 yılında sunulmuş bir dönüştürücü modelidir. BERT ile birlikte Google çeviri başarısı da önemli ölçüde artmıştır.



**Şekil A.5:** Dönüştürücü Model Mimarisi

BERT modeli, bir dizi anahtar-değer çiftini bir çıktıyla eşlemektedir. Burada sorgu, anahtarlar, değerler ve çıktının kendi aralarındaki korelasyonu ifade edecek vektörler oluşmaktadır. BERT modelinde metin sağdan sola hem de soldan sağa doğru işlemektedir. Bu sayede metin içerisindeki öğelerin ilişkileri öğrenibilmektedir. Eğitim aşamasında, MLM (Masked Language Modeling) ve NSP (Next Sentence Prediction) teknikleri kullanılmaktadır. MLM tekniğinde, maskelenen kelimeler, açık (maskelenmeyen) kelimeler kullanılarak tahmin edilmeye çalışılmaktadır. Bu teknik ile cümle içindeki kelimeler üzerinden inceleme ve tahminleme yapılmaktadır. NSP tekniğinde, bir cümlenin kendisinden sonra gelen cümle ile ilişkisine bakarak cümlelerin birbirleri ile ilişkisi incelenmektedir. Bu sayede metinlere üzerinde ince ayar ve özet oluşturmayı sağlamaktadır.

# A.6. T5: Metinden Metne Aktarım Transformatörü

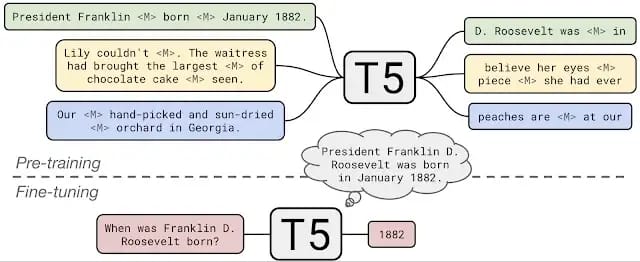
T5 metinden metne aynı formatta dönüştürmeye çalışır. Bir Maskeli LM ve Sonraki Cümleleri önceden eğitilmesini ve ardından aşağı akış görevlerinde ince ayar yapılmasını sağlar. Metinden metne çerçeve, aksine, tüm NLP görevlerinde aynı modelin, kayıp fonksiyonunun ve hiper parametrelerin kullanılmasını önerir. Bu yaklaşımda girdiler, modelin bir görevi tanıyacağı şekilde modellenir ve çıktı basitçe beklenen sonucun "metin" versiyonudur.

# A.7. C4-Corpus

Büyük etiketlenmemiş veri kümelerinde dil modellerini önceden eğitmen gerekir. Common Crawl, bu tür veri kümelerinden biridir. Web sayfalarını kazıyarak ve HTML'den işaretlemeyi göz ardı ederek elde edilir. Her ay yaklaşık 20 TB veri üretir. Ancak Common Crawl, hata mesajları gibi büyük miktarda anlamsız metin veya yinelenen metinleri içermektedir. Ayrıca, rahatsız edici kelimeler(spam), yer tutucu metin veya kaynak kodları gibi kayda değer miktarda gereksiz metin vardır.C4 için yazarlar, Nisan 2019'dan itibaren Common Crawl scrape'ı aldı ve üzerine bazı temizleme filtreleri uyguladılar. Bunlar:

* Yalnızca nokta, ünlem işareti, soru işareti veya son tırnak işareti ile biten tutma cümlelerin ayrıştırılması.
* Kirli, Müstehcen, Müstehcen Sözler Listesinde” görünen rahatsız edici kelimeler içeren herhangi bir sayfanın kaldırılması.
* “JavaScript etkinleştirilmelidir” tipi uyarılar, JavaScript kelimesini içeren herhangi bir satır filtrelenerek kaldırılır.
* "lorem, ipsum" gibi yer tutucu metin içeren sayfalar kaldırılır.
* Kaynak kodları, kaşlı ayraç "{" içeren sayfalar kaldırılır
* 3 cümlenin yinelenen oluşumları filtrelenir.

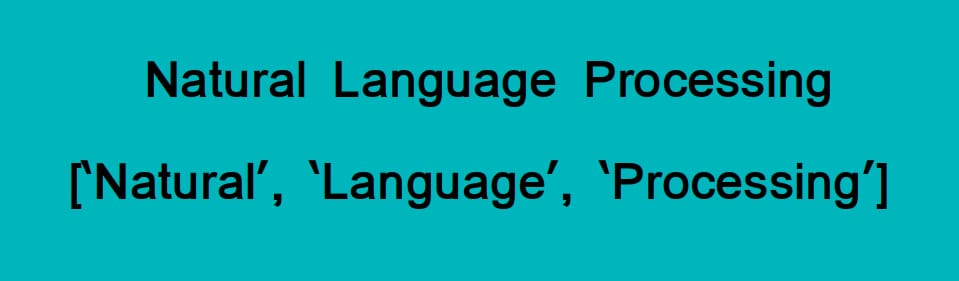
T5, Kodlayıcı-Kod Çözücü mimarisiyle gürültü giderme, bozma yayılma hedefi için ilk olarak C4 veri kümesinde önceden eğitilmiştir. Ardından, metinden metne ayarı için uygun girdi modellemesi ile denetimli bir hedefle aşağı akış görevlerinde ince ayar yapılır.



**Şekil A.6:** **T5:** Metinden Metne Aktarım Transformatörü

# A.8. Tokenization

Tokenization, bir metin parçasını token adı verilen daha küçük birimlere ayırmanın bir yoludur. Burada tokenler, kelimeler, karakterler veya alt kelimeler olabilir.,



**Şekil A.8:** **T5:** Tokenizasyon Örneği

# A.9. Projede Kullanılan Kütüphaneler

## A.9.1.Numpy

[Python](https://tr.wikipedia.org/wiki/Python) programlama dili için büyük, çok boyutlu [dizileri](https://tr.wikipedia.org/wiki/Dizi_(veri_yap%C4%B1s%C4%B1)) ve [matrisleri](https://tr.wikipedia.org/wiki/Matris_(matematik)) destekleyen, bu diziler üzerinde çalışacak üst düzey matematiksel işlevler ekleyen bir kitaplıktır. NumPy'nin atası Numeric, ilk olarak Jim Hugunin tarafından diğer birkaç geliştiricinin katkılarıyla oluşturuldu. 2005 yılında Travis Oliphant, Numarray'in özelliklerini kapsamlı değişikliklerle Numeric'e dahil ederek NumPy'yi yaratmıştır.

NumPy, optimize etmeyen bir bayt kodu yorumlayıcısı olan Python'un [CPython](https://tr.wikipedia.org/wiki/CPython) referans uygulamasını hedeflemektedir. Python'un bu sürümü için yazılan matematiksel [algoritmalar](https://tr.wikipedia.org/wiki/Algoritma), genellikle derlenmiş eşdeğerlerinden çok daha yavaş çalışmaktadır. NumPy, çok boyutlu diziler ve diziler üzerinde verimli çalışan işlevler ve operatörler sağlayarak yavaşlık sorununu kısmen ele alır. Bunları yapmak için NumPy kullanarak, çoğunlukla iç döngüler olmak üzere bazı kodların yeniden yazılmasını gerekmektedir.

## A.9.2. Random

Python, [nesne yönelimli](https://tr.wikipedia.org/wiki/Nesne_y%C3%B6nelimli), [yorumlamalı](https://tr.wikipedia.org/wiki/Yorumlanan_programlama_dili), birimsel (modüler) ve etkileşimli [yüksek seviyeli](https://tr.wikipedia.org/w/index.php?title=Y%C3%BCksek_seviyeli_programlama_dili&action=edit&redlink=1) bir [programlama dilidir](https://tr.wikipedia.org/wiki/Programlama_dilleri).

Girintilere dayalı basit söz dizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır. Bu da ona söz diziminin ayrıntıları ile vakit yitirmeden programlama yapılmaya başlanabilen bir dil olma özelliği kazandırır.

[Django](https://tr.wikipedia.org/wiki/Django), [Zope](https://tr.wikipedia.org/wiki/Zope) uygulama sunucuları, [YouTube](https://tr.wikipedia.org/wiki/YouTube) ve orijinal [BitTorrent](https://tr.wikipedia.org/wiki/BitTorrent) istemcisi Python kullanan önemli projelerden bazılarıdır. Ayrıca [Google](https://tr.wikipedia.org/wiki/Google), [NASA](https://tr.wikipedia.org/wiki/NASA) ve [CERN](https://tr.wikipedia.org/wiki/CERN) gibi büyük kurumlar da Python kullanmaktadır. [Pygame](https://tr.wikipedia.org/wiki/Pygame) ile 2D oyun yapılabilir, [Blockchain](https://tr.wikipedia.org/wiki/Blok_zinciri) uygulamaları kodlanabilir, uzaktan kontrol veya görüntü işleme yapılabilir, veri analizi veya veri kontrolü yapılabilir, [TensorFlow](https://tr.wikipedia.org/wiki/TensorFlow), [PyTorch](https://tr.wikipedia.org/wiki/PyTorch), [Keras](https://tr.wikipedia.org/wiki/Keras) gibi kütüphanlerle derin makine öğrenmesi uygulamaları yapılabilir.

## A.9.3 Os

Bildiğiniz gibi, işletim sistemlerinin çalışma mantığı birbirinden farklıdır. Örneğin Windows ve GNU/Linux işletim sistemleri aynı işi birbirlerinden farklı şekillerde yaparlar. Mesela Windows’ta bir dizin içinde hangi klasör ve dosyaların olduğunu öğrenmek için dir komutunu kullanırız. GNU/Linux’ta ise aynı işlev için ls adlı bir komut vardır.

Aynı şekilde, iki işletim sistemi arasında dizin ayraçları konusunda da farklılık bulunur. Windows’ta dizinleri birbirinden ayırmak için ters taksim (**\**) işareti kullanılırken, GNU/Linux’ta aynı iş için düz taksim (**/**) işareti kullanılır.

İşte biz hem Windows’ta, hem de GNU/Linux’ta çalışacak bir program yazmak istediğimizde bu farklılıkları göz önünde bulundurmamız ve farklı durumların her biri için ayrı kodlar yazmamız gerekirken, os modülü bizi bu zahmetten kurtarır ve bize ortak bir arayüz üzerinden farklı işletim sistemleri ile tutarlı bir şekilde iletişim kurabilmemizi sağlayacak pek çok fonksiyon ve nitelik sunar.

## A.9.4 PyTorch

PyTorch, Torch kütüphanesine dayanan açık kaynaklı bir makine öğrenme kütüphanesidir,[2][3] bilgisayarla görme ve doğal dil işleme gibi uygulamalar için kullanılır.[4] Öncelikle Facebook'un AI Araştırma laboratuvarı (FAIR) tarafından geliştirilmiştir.[5] Değiştirilmiş BSD lisansı ile piyasaya sürülen ücretsiz ve açık kaynaklı bir yazılımdır. Python arabirimi daha öne çıkan ve geliştirmenin birincil odağı olmasına rağmen, PyTorch'un bir C ++ arabirimi de vardır

PyTorch, derin öğrenme modelleri oluştururken geliştiricilerin işlerini oldukça kolaylaştıran bir Python kütüphanesidir. Grafik işlem birimlerini kullanan PyTorch sağladığı esneklik ve hız ile de günümüzde oldukça popüler konumdadır.

PyTorch’un başarısının arkasındaki en önemli nedenlerden biri, tamamen Pythonic olması ve sinir ağı modellerini zahmetsizce oluşturabilmesidir. PyTorch, diğer rakiplerine göre piyasada daha genç bir oyuncu olsa da mükemmel bir ivmeyle gelişmekte.,

PyTorch Özellikleri:

* Basit Arayüz: Kullanımı kolay API sunar, bu nedenle Python gibi çalıştırılması çok basittir.
* Pythonic in nature: Pythonic olan bu kütüphane, Python veri bilimi ile sorunsuz bir şekilde bütünleşir. Böylece Python ortamı tarafından sunulan tüm hizmetlerden ve işlevlerden yararlanabilir.
* Hesaplamalı grafikler: PyTorch dinamik hesaplama grafikleri sunan mükemmel bir platform sağlar, böylece çalışma zamanı sırasında bunları değiştirebilirsiniz. Bu, nöral bir ağ modeli oluşturmak için ne kadar bellek gerekeceğini bilmediğinizde oldukça kullanışlıdır.

## A.9.5 nltk

Doğal Dil Araç Seti, geliştirme ve eğitimde kullanılmak üzere ilk olarak Steven Bird, Edward Loper ve Ewan Klein tarafından yazılan Python programlama dili için açık kaynaklı bir kütüphanedir.

Hesaplamalı dilbilim konularının yanı sıra, programlamada derin bilgiye sahip olmayan dilbilimciler, mühendisler ve araştırmacı dilbilim, öğrenciler ve eğitimciler için araştırmacılar için uygun hale getiren programlama temelleri ile birlikte pratik bir kılavuzla birlikte gelir.

NLTK, 50'den fazla şirket ve Penn Treebank Corpus, Açık Çok Dilli Wordnet, Problem Report Corpus ve Lin'in Bağımlılık Sözlüğü gibi sözcüksel kaynakları içerir.

###### A.9.6 stopwords(Gereksiz Kelimeler)

Stopwords, herhangi bir dildeki cümleye fazla anlam katmayan kelimelerdir. Cümlenin anlamından ödün vermeden güvenle göz ardı edilebilirler. İngilizce’de bu kelimelere “and”,”the”,”to” örnek verilebilir.



**Şekil A.9.6:**  Gereksiz Kelime Örneği

## A.9.7. String

Python’da herhangi bir karakter dizisine string denir. Tek bir harften oluşabileceği gibi, içerisinde boşluklar olabilir, özel karakterler veya rakamlar barındırabilir. Python’da string veri tipi “str” ile ifade edilir.

Bilgisayarlar, ikili (binary) sayılardan yani 1 ve 0’lardan anlarlar. Her ne kadar ekranda karakterleri görsekde, içerde ASCII ve Unicode gibi yöntemlerle ikili sisteme dönüştülürler (encoding).

## A.9.8 pke

pke, açık kaynaklı bir python tabanlı anahtar sözcük çıkarma araç takımıdır.

pke modülü, her bir bileşenin yeni yaklaşımlar geliştirmek için kolayca değiştirilebildiği veya genişletilebildiği uçtan uca bir anahtar kelime ayıklama hattı sağlar. pke aynı zamanda son teknoloji anahtar sözcük çıkarma yaklaşımlarının kolayca karşılaştırılmasına olanak tanır ve SemEval-2010 veri kümesinde eğitilmiş denetimli modellerle birlikte gönderilir .

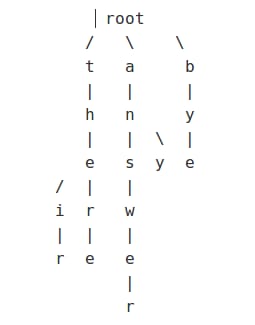
A.9.9 Traceback

Bu modül, Python programlarının yığın izlerini ayıklamak, biçimlendirmek ve yazdırmak için standart bir arabirim sağlar. Bir yığın izi yazdırdığında Python yorumlayıcısının davranışını tam olarak taklit eder. Bu, tercümanın etrafındaki bir "sarmalayıcı" gibi, program kontrolü altında yığın izlerini yazdırmak istediğinizde kullanışlıdır

## A.9.10. flastext

FlashText, özellikle bir belgedeki sözcükleri aramak ve değiştirmek amacıyla oluşturulmuş bir Python kitaplığıdır. Şimdi, FlashText'in çalışma şekli, bir kelime veya bir kelime listesi ve bir dize gerektirmesidir. FlashText'in anahtar kelimeleri çağırdığı kelimeler daha sonra dizide aranır veya değiştirilir.

FlashText'in çalışması hakkında ayrıntılı olarak kontrol edelim. Anahtar kelimeler, arama veya değiştirme için FlashText'e iletildiğinde, bunlar, Alma görevlerinde çok verimli olan bir Trie Veri Yapısı olarak saklanır. Aşağıda, bir Trie Veri Yapısının nasıl göründüğüne dair bir örnek verilmiştir.



**Şekil A.9.10:**  Kelime Ayrıştırma

Anahtar sözcüklerin aranması durumunda, FlashText dizede bulunan anahtar sözcükleri döndürür. Değiştirme durumunda, FlashText değiştirilen anahtar sözcüklerle yeni bir dize oluşturacaktır. Bu işlemlerin her ikisi de tek bir geçişte gerçekleşir.

Aho-Corasick algoritması ve Trie Sözlüğü kullanır. Bu algoritma, birden fazla kullanım durumuna ve onu kullanmanın farklı yollarına sahip. Genel olarak, anahtar kelime eşleme, verilen korpusta her bir anahtar kelimeyi aramak gibi çalışır. Örneğin: 100 anahtar kelimeniz var ve 2k dokümanda arama yapmanız gerekiyor. Her seferinde bir anahtar kelime alınacak ve 2k korpusta aranacaktır. Yine bir sonraki anahtar kelime. Bu, 100\*2000 = 200000 yineleme anlamına gelir.

## A.9.11 gradio

Makine Öğrenimi modelleri yapmak, birçok açık kaynaklı ve özel tabanlı yönetim (örneğin Python, R, SAS) nedeniyle giderek daha basit hale geliyor. Her ne kadar profesyoneller sürekli olarak bitmiş modellerini test etmek ve ortaklar veya ortaklarla paylaşmak için etkili bir şekilde arayüzler oluşturmanın zor olduğunu düşünseler de.

Bu sorun için olası bir yanıt, iş arkadaşlarınıza ve yol arkadaşlarınıza bağlanmak için kolayca aktarabileceğiniz model kullanıcı arabirimleri oluşturmanıza yardımcı olan ücretsiz bir açık kaynaklı Python paketi olan Gradio'dur.

Gradio, ML modeliniz, herhangi bir API veya herhangi bir öznel kapasite için yalnızca birkaç kod satırında hızlı bir şekilde kullanımı basit, ayarlanabilir UI parçaları yapmanıza izin veren açık kaynaklı bir python kitaplığıdır. Resimlerinizi, metninizi veya kendi ses kaydınızı vb. sürükleyip bırakarak ve çıktıyı etkileşimli bir şekilde canlı görerek web tarayıcınızda modellerinizle oynamayı kolaylaştırır. GUI'yi doğrudan Python not defterinize koordine edebilir veya bağlantıyı herkesle paylaşabilirsiniz.

Müşterileriniz/kullanıcılarınız/ekip üyeleriniz için yararlı olabilecek makine öğrenimi kodlarınızın demolarını oluşturabilirsiniz. Geliştirme sırasında etkileşimli olarak modellerinizin hatalarını ayıklayabilirsiniz.Kullanıcılardan model performansı hakkında geri bildirim alabilirsiniz. Sonuç olarak, modelinizi kolayca ve daha hızlı geliştirebilirsiniz. TensorFlow ve PyTorch modelleriyle entegre edilebilir.

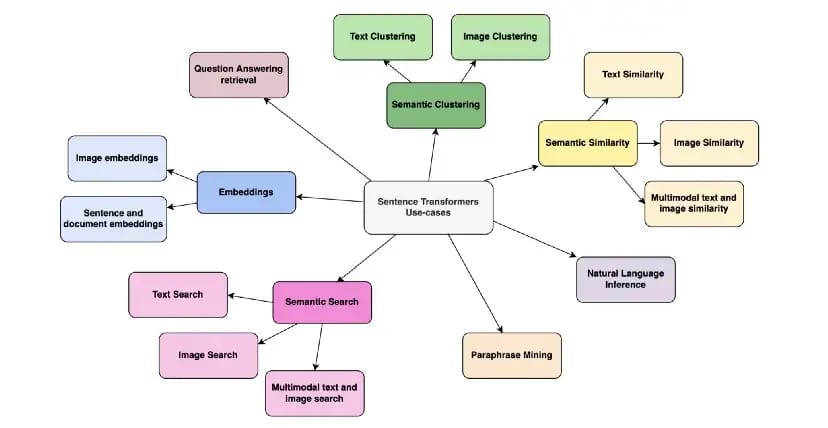
## A.4.3 Sense2Vec

Sense2Vec (Bağlamsal anahtarlı sözcük vektörleri) daha ilginç ve ayrıntılı kelime vektörleri öğrenmenizi sağlayan word2vec'in güzel bir versiyonudur. Bu kitaplık, sense2vec modellerini yüklemek, sorgulamak ve eğitmek için basit bir Python uygulamasıdır. Özellikleri:

* Konuşma bölümü etiketlerine ve varlık etiketlerine dayalı olarak çok sözcüklü tümcecikler için sorgu vektörleri sağlar.
* spaCy ardışık düzen bileşeni ve uzantı öznitelikleri vardır.
* Sense2vec vektörlerinizi spaCy model paketlerinizle kolayca gönderebilmeniz için tamamen seri hale getirilebilir.

## A.4.3 SenseceTransformer

Bu çerçeve, cümleler , paragraflar ve resimler için yoğun vektör temsillerini hesaplamak için kolay bir yöntem sağlar . Modeller, BERT / RoBERTa / XLM-RoBERTa vb. gibi trafo ağlarına dayalıdır ve çeşitli görevlerde son teknoloji performans elde eder. Metin, benzer metin yakın olacak ve kosinüs benzerliği kullanılarak verimli bir şekilde bulunabilecek şekilde vektör uzayına gömülmektedir.



**Şekil A.4.3. :** SentenceTransformers kitaplığının kullanım örnekleri

## A.4.3 NormalizedLevenshtein

Python'da Levenshtein Mesafesi ile Metin Benzerliği için kullanılır.

Metinler arasındaki benzerliği belirlemek, NLP'de yaygın bir sorundur ve dünya çapında birçok şirket tarafından kullanılmaktadır. Metin benzerliğinin en yaygın uygulaması, intihal edilmiş metni belirleme biçiminden gelir. Dünyanın her yerindeki ilkokul, lise, kolej ve üniversitelerden eğitim tesisleri Turnitin gibi hizmetleri kullanmaktadırlar. Öğrenciler tarafından gönderilen çalışmaların orijinal ve kendilerine ait olmasını sağlamak, metin benzerliği uygulamaları, Stack Overflow veya Stack Exchange'e benzer bir yapıya sahip şirketler tarafından yaygın olarak kullanılır. Yinelenen soruları tanımlayabilmek ve işaretleyebilmek istiyorlar, böylece soruyu gönderen kullanıcı, çözümle birlikte orijinal gönderiye atıfta bulunabiliyor. Bu, sorulan yinelenen soruların sayısını azaltır ve platformlarındaki benzersiz soruların sayısını artırır.

Metin benzerliği anlamsal benzerlik ve sözlüksel benzerlik olmak üzere iki bileşene ayrılabilir. Bir çift metin verildiğinde, çiftin anlamsal benzerliği, belgelerin anlam olarak ne kadar yakın olduğunu ifade eder. Oysa sözcüksel benzerlik, sözcük dağarcığındaki örtüşmenin bir ölçüsüdür. Çiftlerdeki her iki belge de aynı kelime dağarcığına sahipse, sözlüksel benzerlikleri 1'dir ve kelime hazinelerinde örtüşme olmasaydı tersi 0 olur .

Gerçek anlamsal benzerliğe ulaşmak, hem NLP'de hem de Matematikte çok zor ve çözülmemiş bir iştir. Bu, yoğun bir şekilde araştırılan bir alandır ve önerilen çözümlerin çoğu, içlerinde belirli bir dereceye kadar sözcük benzerliği içerir. Bu makalenin odak noktası olarak, anlamsal benzerliğe çok daha derinlemesine dalmayacağım, ancak sözcüksel benzerliğe çok daha fazla öncelik vereceğim.

Levenshtein Mesafesi çok etkilidir çünkü karşılaştırılabilmeleri için iki dizinin eşit uzunlukta olmasını gerektirmez. Sezgisel olarak konuşursak, Levenshtein mesafesini anlamak oldukça kolaydır. Gayri resmi olarak, iki kelime arasındaki Levenshtein mesafesi, bir kelimeyi diğerine değiştirmek için gereken minimum tek karakterli düzenleme sayısıdır (eklemeler, silmeler veya ikameler).

## Textwrap3

Metin kaydırma, doğal dil işleme, veri analizi ve hatta sanat veya tasarım çalışmalarında metin dillerini işlerken faydalı bir araçtır. Python textwrap modülü, bir metin kaydırma efekti oluşturmaya yardımcı olacak kullanışlı işlevler sağlar.

Bu modül ayrıca boşlukları normalleştirmek, girintileri kaldırmak veya eklemek, önek ve sonek metni ve sayısız başka kullanım için kullanılabilecek birkaç ilgili işlev sağlar. Ek olarak:

* Metin Sarma
* Modül Özellikleri
* Bellek açısından daha verimli değerlendirme ile benzer işlevsellik sağlayan bir sınıftır.

## YÖNTEM

# B.1.Temel Cümleleri/Kavramları Belirlenmesi

İlk olarak cümlelerin değerlendirilmesi ve verilen metindeki anahtar kavramların bulunması ilk adım olacaktır. Daha sonrasında az önemli olan cümleleri filtrelemiş oluruz.

# B.2.Anahtar Kelimelerin Belirlenmesi

Cümleleri filtreledikten sonra anahtar kelime öbeklerini veya anahtar kelimeleri çıkarmamız gerekiyor. Anahtar kelime öbekleri veya anahtar kelimeler çoktan seçmeli soruların üretileceği temelde doğru cevap görevi görecek cümlelerdeki önemli kavramlardır.

# B.3.Çoktan Seçmeli Sorular Oluşturulması

Kişiler verilerin metin bilgisi ile çoktan seçmeli soruyu oluşturabilirler. Bu sorular, tek anahtar kelime veya anahtar kelime öbekleri ile onlara cevap olarak oluşturulmuştur. Kişiler soru ve doğru yanıtın yanı sıra çeldiriciler de (yanlış yanıt seçenekleri) oluşturacaktır. Temel kural, çeldiricilerin doğru cevaba benzer olması, ancak doğru cevabı verecek kadar açık halde olmamasıdır. Sınava giren kişinin cevap şıkları arasında düşünmesini sağlamak ve kafa karışıklığı oluşturmak içindir.

# B.4.Bir NLP Probleminin Belirlenmesi

Kişiler MCQ’ları oluşturmak için sezgisel olarak kullanacağı adımlardan geçtikten sonra, bunu bir doğal dil işleme problemi olarak belirlemek ve bu problemi çözmeye çalışmaktır. Süreç toplamda 5 adımdan oluşmaktadır. Bunlar;

1. Soyutlayıcı/Açıklayıcı Özetleme
2. Cümlelerin Açıklanması
3. Anahtar Kelimeler
4. Soru Üretimi
5. Yanlış Seçimler/Dikkat Dağıtıcılar Üretimi

# B.4.1.Adım: Soyutlayıcı/Açıklayıcı Özetleme

Özetleme, tüm metnin kısa özetinin oluşturulmasıdır. Metin özetlemenin temelinde 2 yöntem vardır:

* Çıkarımsal Özet: Bu yöntem, orijinal metini tarayarak en önemli cümle alt kümesini seçer ve metni özetler. Adından da anlaşılacağı gibi, metinden en önemli bilgileri çıkarır. Bu yöntemin kendi başına metin oluşturma özelliği yoktur ve bu nedenle çıktı her zaman orijinal metnin bir kısmına sahip olacaktır.
* Soyutlayıcı Özet: Bu yöntemde ise orijinal metnin temel bağlamını anlamak ve bu anlayışa dayalı olarak yeni metin üretmektir. İnsanların metni kendi yollarıyla okuma ve özetleme biçimleriyle karşılaştırılabilir. Soyutlayıcı özetin çıktısı, orijinal metinde bulunmayan öğelere sahip olabilir.

# B.4.2.Adım: Cümlelerin Açıklanması

Cümlelerin başka kelimelerle ifade edilmesinde ilk adım olarak çıkarımsal özetleme kullanıyorsak, sonucu yorumlamamız gerekir. Buna rağmen soyutlayıcı özetlemeyi daha önce kullanmış olsaydık, özetleme sırasında zaten başka kelimelerle ifade edildiği için bu adım atlanılmış olur.

# B.4.3.Adım: Anahtar Kelimeler

Daha Sonrasında başka kelimelerle ifade edilen cümleler verildiğinde bu anahtar kelimeler, bir sonraki adımda oluşturulacak sorular için temel teşkil etmektedir.

Çok parçalı sıralama, anahtar kelimelerin çıkarılmasında kullanışlı bir algoritmadır. Bu görev için açık kaynaklı pk3 kitaplığından yararlanabiliriz.

# B.4.4.Adım: Soru Üretimi

Bu adımda soru oluşturmak için gerekli olan anahtar sözcüklere ve metine sahibiz. Bazı bağlamları ve alakalı anahtar kelimeyi kullanmak ve uygun soruları oluşturmak için eğitilmiş T5 Transformer modelini kullanabiliriz.

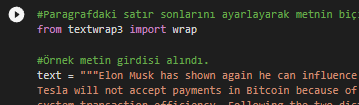
# B.4.5.Adım: Yanlış Seçimler/Dikkat Dağıtıcılar Üretimi

Sorumuz ve doğru cevabımız bu adımda mevcut. MCQ’nun oluşturulmasını tamamlamak için, Şimdi MCQ oluşturmak için yanlış secimler olan çeldiriciler oluşturmak gerekiyor. Örneğin, Türkiye’nin başkenti neresidir olan sorunun cevabı İstanbul ise bu cevaplar içinde başka metropol şehirleri üreterek çeldiriciler oluşturmalıyız. Çeldiricileri bulmak için kelime vektör algoritmalarını kullanabiliriz. Ayrıca OpenAl’nin GPT-3 dil modeline yönelik gelişmiş sistemleri kullanabilir ve uygun çeldiricileri oluşturabiliriz.

## PROJENİN GERÇEKLENMESİ

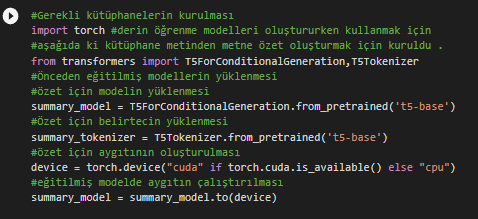
# C.1.Programda modelin eğitimi için kullanılması istenen örnek metin girdileri

**Tablo C.1**.Örnek Metin Girdileri



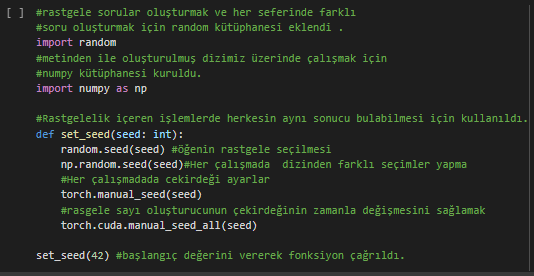
# C.2. Metin Özetleme için Gerekli veri setlerinin İndirilmesi

**Tablo C.2**.Metin Özetleme



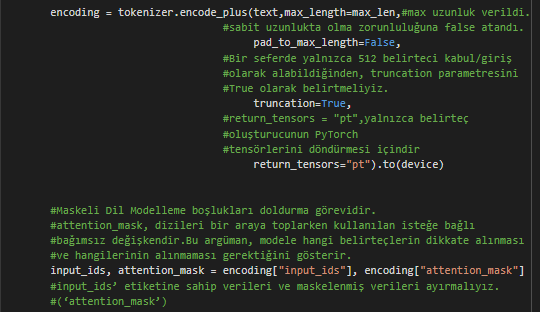
# C.3. Metinden Random Seçimler Yapma

**Tablo C.3**.Random Secim



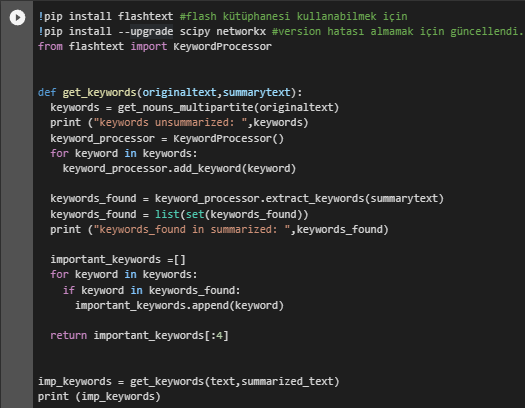
# C.4. Kodlama - Kod Çözücülerin Kullanımı ile Veri Seti Eğitimi

**Tablo C.4**.Metin Özetleme



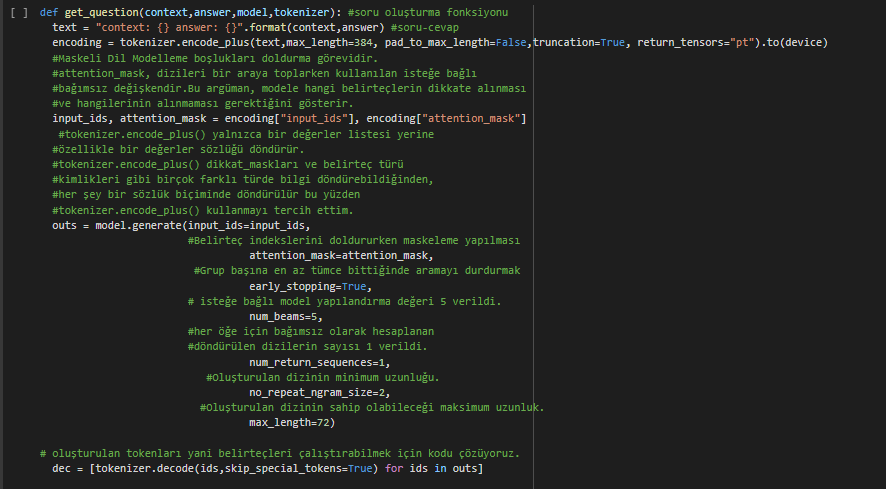
# C.5. Yanlış Şık oluşturabilmek için Sözcük türetimi

**Tablo C.5**.Sözcük Türetimi



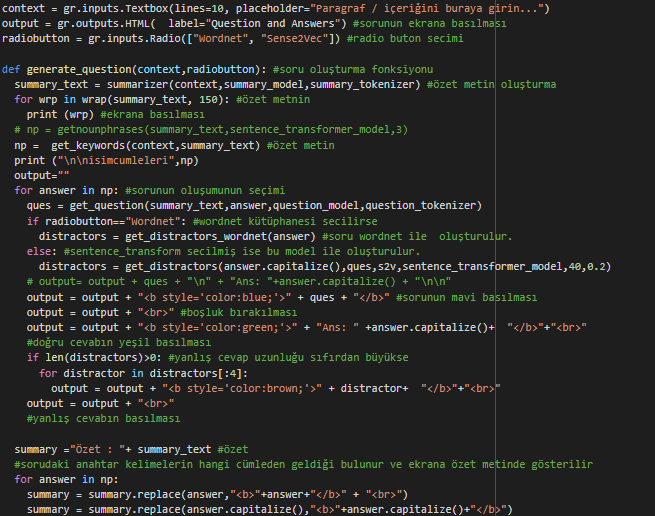
# C.6. T5 ile Soru Üretimi

**Tablo C.6**.Soru Üretimi



# C.7. Gradio ile Tasarım ve Projenin Web Sunumu İçin Kodlar

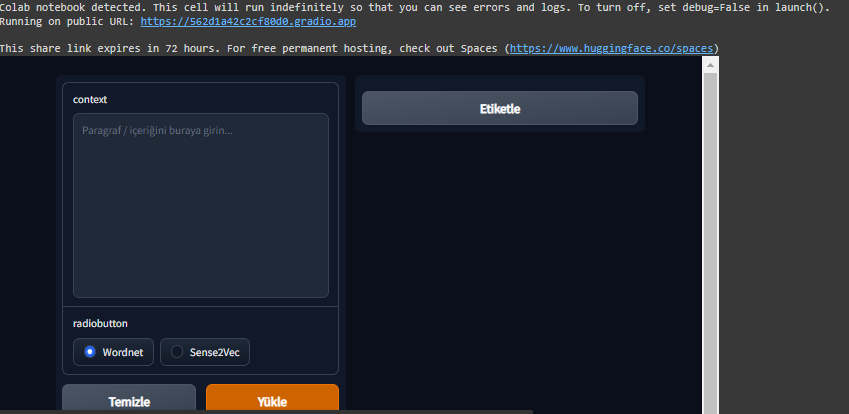
**Tablo C.7**.Gradio İle Proje Web Sunumu



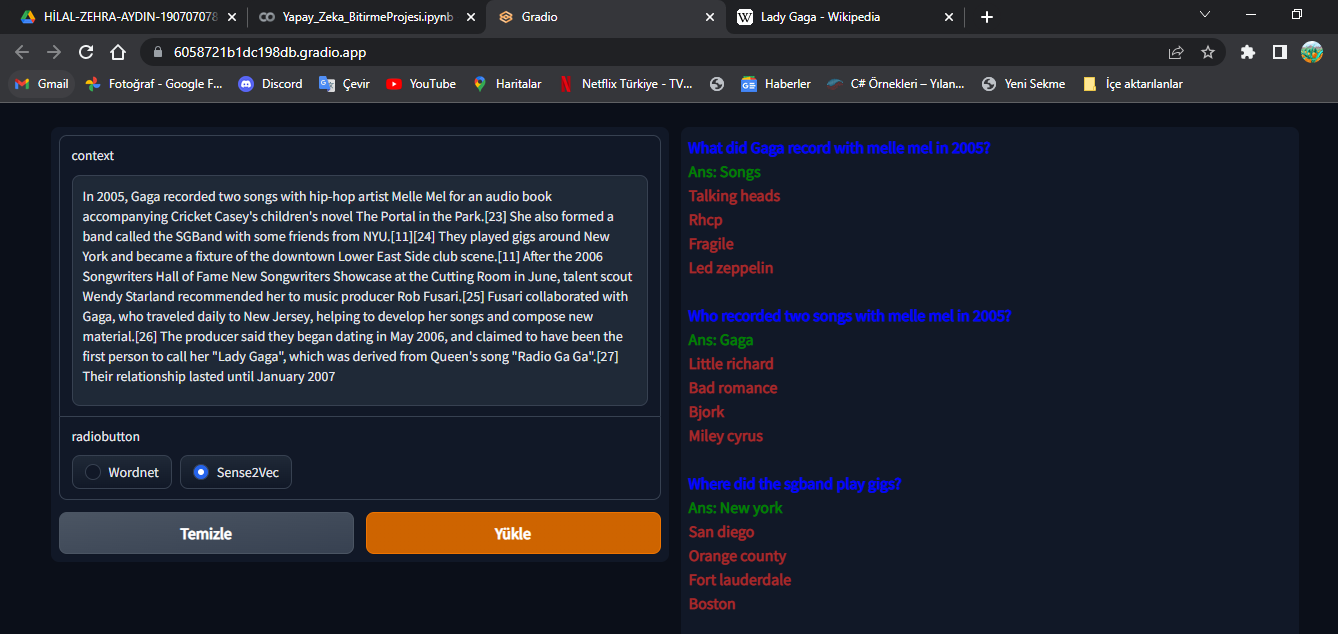
## PROJEDEN ELDE EDİLEN SONUÇLAR

# D.EKRAN ÇIKTILARI

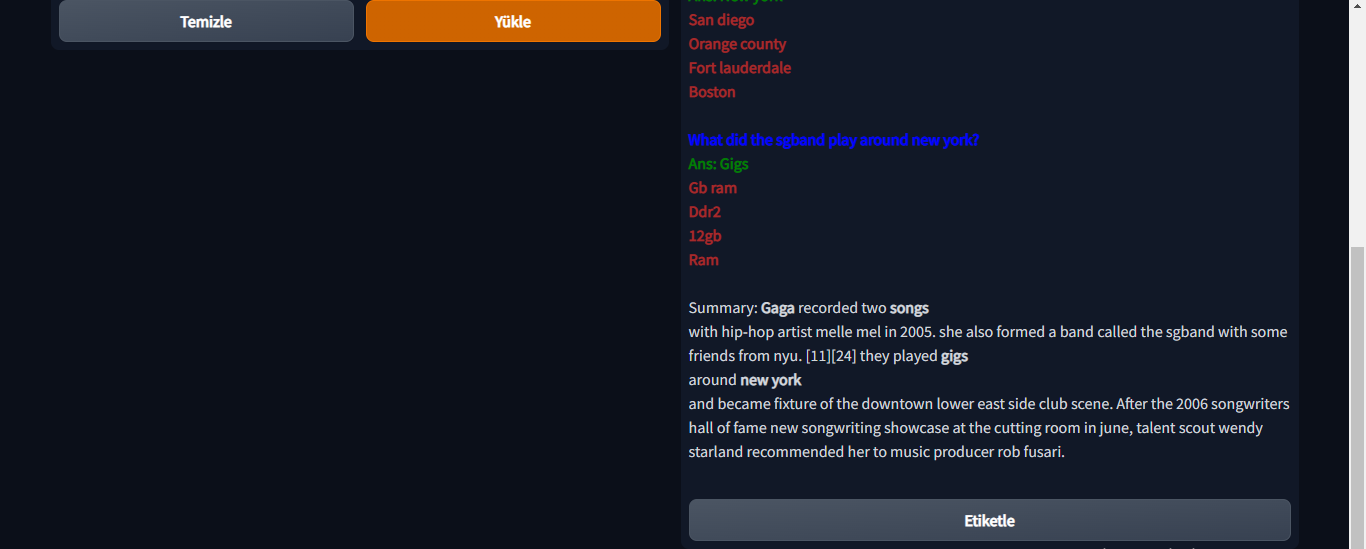
**Tablo D.1**. Gradio ile Web için Hazırlık



**Tablo D.2.** Web Sayfasında Metin Girdisi ve Çıktı Sonucu-1



**Tablo D.3.**Web Sayfasında Metin Girdisi ve Çıktı Sonucu-2(Devamı)



# KAYNAKÇA

[1] **Harris, Charles R.; Millman, K. Jarrod; van der Walt, Stéfan J.; Gommers, Ralf; Virtanen, Pauli; Cournapeau, David; Wieser, Eric; Taylor, Julian; Berg, Sebastian; Smith, Nathaniel J.; Kern, Robert** (17 Eylül 2020). ["Array programming with NumPy"](http://www.nature.com/articles/s41586-020-2649-2). *Nature* (İngilizce). **585** (7825): 357-362. [doi](https://tr.wikipedia.org/wiki/Say%C4%B1sal_nesne_tan%C4%B1mlay%C4%B1c%C4%B1s%C4%B1):[10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). [ISSN](https://tr.wikipedia.org/wiki/Uluslararas%C4%B1_Standart_S%C3%BCreli_Yay%C4%B1n_Numaras%C4%B1) [0028-0836](https://www.worldcat.org/issn/0028-0836). 26 Mayıs 2021 tarihinde kaynağından [arşivlendi](https://web.archive.org/web/20210526005448/https:/www.nature.com/articles/s41586-020-2649-2). Erişim tarihi: 11 Haziran 2021.

[2] "PEP 373 -- Python 2.7 Release Schedule". Python.org (İngilizce). 13 Ocak 2020 tarihinde kaynağından arşivlendi. Erişim tarihi: 22 Eylül 2019

[3] Whetting Your Appetite". The Python Tutorial. Python Software Foundation. Archived from the original on 26 October 2012. Retrieved 20 February 2012.