İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ
2010
FEN BİLİMLERİ ENSTİTÜSÜ

Açık Kaynak Soru-Cevap Sistemi

Yazılım Mühendisliği Ana Bilim Dalı
Yüksek Lisans

Erdoğan Ege ALTIN
Y210234107
ORCID 0000-0000-0000-0000

Tez Danışmanı: Doc. Dr. Aytuğ ÖNAN

Haziran 2023

Yazarlık Beyanı

Ben, **Erdoğan Ege ALTIN**, başlığı **Açık Kaynak Soru-Cevap Sistemi**
olan bu tezimin ve tezin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 10.06.2023

Açık Kaynak Soru-Cevap Sistemi

Öz

İnternetteki makaleler ve diğer çevrimiçi yazılar gibi içerik miktarı her geçen gün hızla artmaktadır. Bu nedenle, en iyi arama motorlarının bile kullanıcıların isteklerine tam olarak doğru cevapları bulmakta zorlandığı görülmektedir. Kullanıcıların istediği sonuçlar genellikle birden farklı makalelere yayılmış durumdadır ve bu da geleneksel arama motorlarının istenenleri bulmasını imkânsız hale getirmektedir.
Mevcut sistemlerdeki mevcut sorun, bilgi erişimi (IR) kısmıdır. Mevcut modeller, belirli bir sayıda belge alırken ilgili bağlamı atlayarak hareket etmektedir. IR için, K-means kümeleme yöntemi ve Random forest sınıflandırıcısı ile TF-IDF benzerlik kısmına küçültülmüş bağlam değerleri eklenmiştir. Bu model, belge kurtarma kısmında yaklaşık %84'lük bir performans elde etmektedir. İkinci kısım olan okuma-anlama (RC) kısmı için ise, deepset'ten pre-trained roberta-base-squad2 isimli hugging face kütüphanesi kullanılmıştır.

Anahtar Sözcükler: Makine öğrenmesi,derin öğrenme, araştırma, mühendislik

Teşekkür

Yüksek lisansa başladığım günden itibaren ilgi ve alakasını bir an bile olsa eksiltmeyen sayın Doç Dr. Aytuğ ÖNAN'a teşekkür ederim. Çalışma hayatımı ve okul hayatımı destekleyen aileme ve Aybüke Nur DURSUN'a teşekkür ederim.

Table of Contents

1. Giriş:

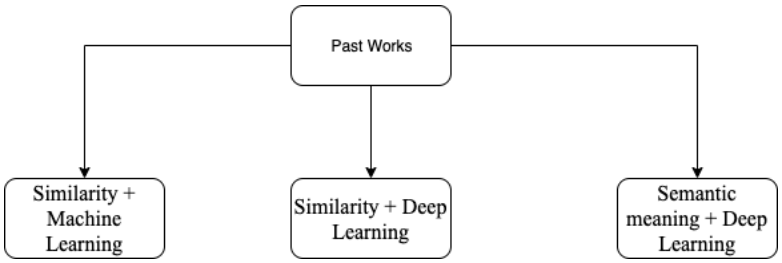
Metin, görsel ve diğer öğeler gibi birçok medyada çevrimiçi olarak büyük bir miktarda kaynak bulunmaktadır. Bu nedenle, sorulara basit çözümler bulmak giderek zorlaşmaktadır. Olası bir çözüm için ilgili web sitelerini taramak bazen zorlu ve zaman alıcı olabilir. Soru cevaplama (QA) yöntemleri bu sorunu ortadan kaldırabilir veya azaltabilir. QA sistemleri, ilgili kaynaklardan en iyi yanıtı çıkarmak için NLP algoritmalarını kullanır.

İki tür QA sistemi vardır: Kapalı domain ve açık domain QA sistemleri.
Kapalı domain QA sistemi, belirli bir domain veya aranması gereken belge ile ilgilenir. Bu nedenle, bu tür sistemlerde sadece bir kısım bulunur, o da Okuma Anlama (RC) kısmıdır. Bu kısımda, geliştiriciler sorulara en iyi yanıtı bulmak için NLP algoritmalarını uygularlar. Bu algoritmalar, makine öğrenmesinden derin öğrenmeye kadar çeşitlilik gösterebilir. Bu tür sistemler için belge kurtarma gereksizdir.

Öte yandan, açık domain QA sistemlerinde iki ana bölüm vardır. İlk bölüm bilgi erişimi (IR), ikinci bölüm ise Okuma-Anlama (RC)'dır. Bunun nedeni, açık domain sistemlerde cevap için belirli bir domain veya belge olmamasıdır. Sorunun cevabı için veri farklı alanlardan veya belgelerden gelir. İlk kısımda, ilgili belge bulunmalıdır. Bunun nedeni, büyük belgelerde cevap bulmanın zaman alıcı olabileceği ve yanlış sonuçlara yol açabileceğidir. Ancak açık domain QA sistemlerinde tam belgeyi bulmak neredeyse imkansızdır çünkü belge veri kümesi genellikle çok büyüktür. Bu nedenle, bu sistemlerin çoğu IR kısmı için en az on belge döndürür. İlgili belgeler bulunduktan sonra, bu belgeler RC kısmına iletilir. Bu kısımda, NLP tekniklerinin yardımıyla sorunun yanıtı alınır.

Benzer araştırmaların çoğunda, yazarlar SQuAD veri kümesini kullanmıştır. Bu nedenle, sonuçları diğer çalışmalarla karşılaştırmak için bu deneyde de bu veri kümesi kullanılmıştır. NLP alanındaki son gelişmelerle birlikte, RC performansı artmıştır. SQuAD veri kümesi için RC için state-of-the-art modellerin doğruluk oranı %90'ın üzerine çıkmıştır. Bu nedenle, bu araştırma ağırlıklı olarak IR sorununa odaklanmaktadır. Çünkü eğer bilgi erişimi algoritması ilgili belgeyi döndürürse, RC modelinin cevabı kaçırma olasılığı düşüktür. Bu amaçla, tüm belge veri kümesi, belge kümeleme teknikleri kullanılarak sorulara dayalı olarak küçültülmüştür. Ardından, en yüksek skora sahip olan RC algoritması için en iyi K belgesi RC algoritmasına iletilmiştir. En iyi sonuçlar, IR kısmı için K-means kümeleme ve Random forest sınıflandırıcının kombinasyonuyla elde edilmiştir. RC kısmı için ise deepset'ten pre-trained roberta-base-squad2 isimli hugging face kütüphanesi en yüksek F1 skorunu almıştır.

2. Literatür Taraması



Şekil 1: Literatürden farklı metodolojiler

Literatür taraması sırasında, QA (soru-cevap) sistemleri için 3 farklı metodoloji tespit edildi. İlk yöntem, IR (bilgi erişimi) kısmı için TF-IDF veya BM25 benzerliği, RC (okuma anlama) kısmı için makine öğrenimi algoritmalarını kullanmaktadır. İkinci yöntemde, IR kısmı için TF-IDF veya BM25 benzerliği kullanılırken, RC kısmı için derin öğrenme algoritmaları tercih edilmektedir. Son yöntemde ise, IR kısmı için anlamsal anlama araması, RC kısmı için ise derin öğrenme algoritmaları kullanılmaktadır.

2.1 Benzerlik + Makine Öğrenimi:

Bir makalede (Chen et al., 2017), yazarlar açık domain QA (soru-cevap) sistemleri geliştirdiler. Cevapları Wikipedia makalelerinde aramaktadırlar. Bu görevin zorluğu, Wikipedia'nın büyük ölçekli metinler içermesidir. Bu geliştirmede başarı elde etmek için yazarlar, bu sorunu hemen hemen her benzer geliştirmede olduğu gibi iki bölüme ayırmışlardır. İlk adım olarak belge geri almadır, burada TF-IDF eşleştirmesi ve bigram karma kullanılmıştır. Soruları ve makaleleri karşılaştırmak için TF-IDF ağırlıklı kelime torbası vektörleri kullanılmıştır. Şemaları eşlemek için karma işlemi, hız ve bellek verimliliğini iyileştirmek amacıyla kullanılmıştır. Bu bölüm, herhangi bir soru için beş ilgili makale döndürmektedir. Beş ilgili belgeyi geri alımdan sonra, çok katmanlı bir tekrarlayan sinir ağı modeli, soruları yanıtlamak için eğitilmiştir. Bu model, bir soru ve paragraf kodlamasını ve tahminleri içermektedir. Sonuç olarak, belge geri alım bölümü için iyi sonuçlar elde edilmiştir. Bununla birlikte, bu araştırmadaki eksiklik, soru-cevap bölümünde bulunmaktadır. Daha gelişmiş kodlama tekniklerinin kullanılması bu sorunu çözmeye yardımcı olabilir.

Yazarlar (Lee et al., 2018), Açık domain QA sistemlerinde belge geri alma bölümünün sistemin performansı üzerinde büyük bir etkisi olduğunu belirtmişlerdir. Belge geri alma performansını iyileştirmek için, belge geri alma ve soru yanıtlama adımları arasına bir paragraf sıralayıcı eklemişlerdir. Daha fazla belge almak, ilgisiz belge sayısını artırır ve performansı düşürür. Bir belge okuyucu kullanıldı ve paragraf sıralaması için çift yönlü uzun kısa süreli hafıza (Bi-LSTM) uygulandı. Belge okuyucularının performansını ortalama olarak %7,8 artırdılar, ancak sorun, daha gelişmiş makine öğrenimi ve derin öğrenme algoritmalarının kullanımıyla benzerlik puanlarının hesaplanarak çözülebilir.

Belge geri alma bölümü için doğru sayıda belgenin (n sayısı) geri alınması önemlidir. Büyük bir n sayısı seçilirse, soru-cevap adımına çok fazla gürültü (yavaşlatacak) eklenir ve işlem daha yavaş hale gelir. Küçük bir n sayısı seçilirse, doğru belge gözden kaçabilir. Araştırmalarında, yazarlar bu sorunu soru ve belge veri setine dayalı olarak dinamik n seçerek çözmeyi amaçlanmıştır. Önerilen çözüm, eşik tabanlı geri almayla elde edilmiştir. Kümulatif

güven puanı belirli bir eşiğe ulaşırsa, belge sayısı geri alma bölümü için seçilir. Bazı veri kümeleri için iyi sonuçlar elde edilmekle birlikte, genel sonuçlar tatmin edici değildir.

Bu araştırmada (Nishida et al., 2018), yazarlar açık domain QA sistemlerinin bilgi geri alma (IR) bileşeniyle ilgili soruna odaklandılar. Onlara göre, mevcut modellerdeki sorun, cevap aralıklarını dikkate almadan eğitilen IR bileşeninin, büyük bir paragraf koleksiyonundan birkaç uygun bölümü doğru şekilde seçmekte zorlandığıydı. Bu nedenle, yazarlar, cevap aralıklarını dikkate alan denetimli çoklu görev öğrenimi ile eğitilen, IR ve okuma anlama (RC) için ortak gizli katmanları olan bir model önerdiler. IR ve RC için ortak katmanlar, kelime gömme, bağlamsal gömme, dikkat akışı ve modelleme katmanıdır. Dikkat akışı katmanında, kesin eşleme için çift yönlü dikkat (BiDAF) kullanılır ve geri kalanı için LSTM kullanılır. DrQA modelinden (Chen et al., 2017) daha iyi sonuçlar elde etmelerine rağmen, sonuçlar en son teknolojiye sahip modellerle karşılaştırıldığında tatmin edici değildi. Bunun nedeni, BiDAF ve LSTM'nin en son teknoloji algoritmaları kadar güçlü olmaması olabilir.

2.2 Benzerlik + Derin Öğrenme:

Bu araştırma makalesinde (Yang et al., 2019), yazarlar metin uzunluğu sorununu çözmek istediler. Daha ayrıntılı bir şekilde açıklamak gerekirse, araştırma yaparken, çoğu QA sisteminin bir makale, birkaç paragraf veya “k” cümle gibi küçük metinlerle uğraştığını gördüler. Ancak deneylerinde, Wikipedia makaleleri gibi büyük belgeler kullandılar. Hedeflerine ulaşmak için belge geri alma bölümü için Anserini'yi ve soru cevaplama için BERT'i kullandılar. Öncelikle belgeleri kurtaran ve ardından bunlar içindeki bölümleri puanlayan çok aşamalı bir geri alıcı yerine, Wikipedia metin segmentlerini hemen BERT okuyucuya beslemek için tek aşamalı bir geri alıcı kullandılar ve sıralama işlevi olarak BM25'i kullandılar. Bu adımdan sonra, BERT-Base modelini (uncased, 12 katman, 768 gizli, 12 başlıklar, 110M parametre) kullanarak SQuAD veri setinin 1.1 sürümüyle yeniden eğitim yaptılar. Sonuç olarak, ihtiyaçlarına dayanarak BERT modelini değiştirdiler. Deneyin çoğu sonucu mevcut sistemlerden çok daha iyi durumdadır. Bununla birlikte, geri alım ve cevap çıkarmanın hala geliştirilmesi gerektiğini belirtmişlerdir.

Yapılan çalışmaya benzer olan önceki çalışmalarda araştırmacılar, aynı soruya ilişkin birden çok paragraf için bağımsız eğitim örneğiyle BERT algoritmasını eğitiyordu. Bu durum, cevaplara karşı karşılaştırılamaz puanlar üretme olasılığına yol açabilmekteydi. Bu sorunu çözmek için bu araştırmanın yazarları çoklu geçişli bir sistem geliştirdiler. Bu sistemde, her bir paragrafa ayrı ayrı bakmak yerine, cevapları aynı anda tüm paragraflarda bulmaya çalışmakta. Ayrıca, sistem performansını artırmak için makaleleri 100 kelime uzunluğunda böldüler. Bu durum, BERT-RC (Okuma anlama) ile nispeten benzer bir sürece sahiptir, ancak normalleştirme adımından sonra, tüm bölümlerdeki tüm kelime konumlarını eşitlemek için softmax kullanılır. Performansı %4 artırdılar ve olağanüstü bölümleri seçmek için bir bölüm sıralayıcısı uyguladıktan sonra sisteme ekstra %2 eklediler. Bu araştırmadaki eksiklik, yazarların hangi yöntemleri kullandıklarını yeterince belirtmemiş olmaları olabilir.

Başka bir araştırmada (Kratzwald et al., 2019), araştırmacılar geleneksel Açık domain Soru-Cevap (QA) sistemine bir adım daha eklemişlerdir. İlk iki adım, belge alımı ve metin anlama, değiştirilmemiştir. Üçüncü adımda, cevapları yeniden sıralama, alım ve anlama özelliklerinin karışımı kullanılarak, QA sürecinden doğrudan elde edilen birkaç özellikten yararlanılmaktadır. Belge alımı için DrQA modeli (Chen et al., 2017) ve makine anlama için BERT-QA kullanılmıştır. Ardından, en iyi “k” aday cevap seçilir. Her cevap için QA sürecinden bir dizi özellik çıkarılır ve cevap birleştirme uygulanır. Son olarak, en iyi “k”

aday, önceki iki adıma dayanarak yeniden sıralanır. Bu deneyin sonucunda, araştırmacılar dört devrin üçünde daha iyi performans elde etmişlerdir. Ancak, ilk iki adımda kullandıkları model artık devrin en gelişmiş modeli olarak kalamamışlardır.

Yazarlar (Seo et al., 2018) bu araştırmada Makine Anlama(Machine Comprehension) problemine odaklanmışlardır. Araştırmaları sırasında, dikkat mekanizmalarının makine anlama(Machine Comprehension) için iyi performans gösterdiğini tespit etmişlerdir. Ancak, mevcut modelin sorunu, bağlamı sabit bir vektörle özetlemek ve bunun küçük bir kısmına odaklanmaktır. Bu nedenle, erken özetleme sorunlarını önlemek için Bi-Directional Attention Flow (BiDAF) ağı geliştirmişlerdir. Bu deneyde SQuAD ve CNN/DailyMail veri setleri kullanılmıştır. Bu model altı katmana sahiptir: Karakter Gömme, Kelime Gömme, Bağlamsal Gömme, Dikkat Akışı, Modelleme ve Çıkış Katmanı. Yazarlar bu araştırma sırasında dönemin en gelişmiş sonuçlarına sahip olsalar da, günümüzde Bi-Directional algoritmaların (örneğin BERT) makine anlama için kodlayıcılarla birleştirilmiş daha gelişmiş teknikleri bulunmaktadır. Bu boşluğu doldurmak için bu bölümde daha gelişmiş teknikler kullandık ve bunları belge alımıyla birleştirdik.

Bu araştırmada (Weissenborn et al., 2017), yazarlar Soru-Cevap problemleri için basit ve verimli bir uçtan uca sinirsel model olan FastQA modelini geliştirmişlerdir. Ayrıca, FastQAExt adında genişletilmiş bir versiyonu da göstermişlerdir. Bu model, popüler veri setleri olan SQuAD, NewsQA ve MsMARCO'da devrin en iyi performansını elde etmiştir. İki adım vardır; cevap aralığının türü, sorunun istediği cevap türüyle eşleşmelidir ve cevap, soruya uygun bir bağlama sahip olmalıdır. Bu iki adımı gerçekleştirmek için Bi-Directional Rekürrent Sinir Ağları (BiRNN) kullanılmıştır. Üç ana bölümü vardır: gömme, özellikler ve kodlama. Bazı durumlar için iyi sonuçlar alınmasına rağmen, araştırmalarında büyük bir boşluk bulunmaktadır. Modeli, birçok NLP problemi için istenmeyen bir şekilde tür eşleme heuristiğine dayanmaktadır. Çünkü model, yanlış yazılmış ya da benzer kelime kullanımı için iyi performans gösteremez.

Bu araştırmada (Devlin et al., 2019), yazarlar önceden eğitilmiş BERT modeliyle farklı deneyler yapmışlardır. BERT, cümle özetlemeden soru cevaplama gibi çeşitli NLP görevleri için güçlü bir araçtır. Bu işlem, önceden eğitilmiş BERT modeline yalnızca bir ek çıkış katmanı ekleyerek yapılabilmektedir. Bu, model için mimari değişiklik yapma gereksinimi olmadığı anlamına gelmektedir, bu da işlemi daha kolay ve hızlı yapılabilir duruma getirmektedir. Bu deneyler için SQuAD v1.1 ve 2.0 gibi farklı türde veri setleri kullanılmıştır. Sonuç olarak, deneylerinde dönemin en iyi sonuçlarını elde edilmiştir. Bu, transferin güçlü ve hesaplama açısından ucuz olduğunu göstermektedir. Ancak, bu araştırmadaki açık belge alımıdır.

Başka bir araştırmada (Wang et al., 2016), yazarlar mevcut soru-cevaplama prosesi için var olan veri setlerinin eğitilmiş bir modele göre çok küçük olduğunu ya da mevcut Makine anlama yöntemlerinin etkinliğini test etmek için yeterince zorlayıcı olmadığını düşünmektedir. Bu nedenle, SQuAD veri seti yayınlandıktan sonra Multi-Perspective Context Matching (MPCM) adlı bir model geliştirmişlerdir. Bu model, bir metindeki cevabın başlangıç ve bitiş aralığını tahmin eden uçtan uca(baştan sona giden) bir sistemdir. Bu modelin yaklaşımı, her kelime gömme vektörünü sorguya karşı hesaplanan bir ilgi ağırlığıyla çarpmaktır. Ağırlıklı geçiş ve soru, çift yönlü LSTM'ler kullanılarak kodlanır. Yaklaşımları, her noktanın metindeki bağlamını, kodlanmış sorguyla birkaç açıdan karşılaştırarak, metindeki her nokta için eşleşme vektörü oluşturur. Verileri birleştirmek ve eşleşen vektörleri

kullanarak başlangıç ve bitiş noktalarını tahmin etmek için başka bir çift yönlü LSTM kullanılmıştır. Ancak, sonuçlar tatmin edici değildir ve belge alımı eksiktir.

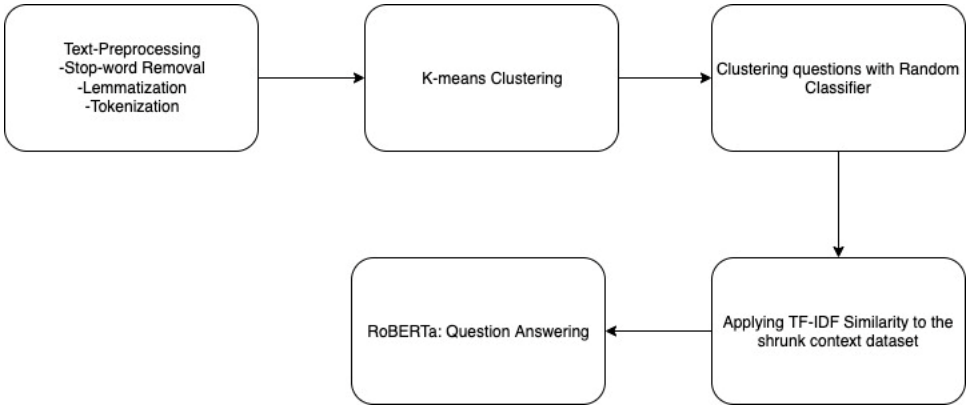
Bu araştırmada (Wang et al., 2017), yazarlar soru-cevaplama görevi için Gated Attention Reader (GAR) adlı bir model kullanmışlardır. Model, okuyucunun bir belgeyi okumasını ve belgedeki önemli kısımlara odaklanmasını simüle eden dikkat mekanizmasını kullanmaktadır. Model, bir soruyu ve belgeyi çift yönlü rekürrent sinir ağlarıyla kodlar ve ardından soru ve belge arasında bir dikkat ağı oluşturur. Bu dikkat ağı, okuyucunun belgedeki önemli kısımlara odaklanmasına yardımcı olur. Son olarak, dikkat ağı ve kodlama katmanları birleştirilerek cevap üretilir. Yazarlar, bu modelin SQuAD veri setinde diğer modellere kıyasla rekabetçi sonuçlar elde ettiğini belirtmiştir. Ancak, bu araştırmadaki boşluk, belge alımının sınırlı olmasıdır ve daha büyük veri setlerinde test edilmediği için genelleme yeteneği belirsizdir.

2.3 Kavramsal anlam + Derin Öğrenme:

Lee et al. (2019) adlı makalede, bilgi geri alma (IR) sürecinin bir kara kutu olduğunu varsaydılar. Bu nedenle, IR olmadan geri alıcı ve okuyucunun ortak öğrenilmesini ilk kez önerdiler ve bunu Açık-Geri Alma Soru Cevaplama (ORQA) olarak adlandırdılar. IR sistemlerinin QA görevleri için arama alanını önemli ölçüde azalttığını öne sürdüler. Sonuç olarak, geri alıcı ve okuyucu bileşenlerinin ortak olarak öğrenildiği bir uçtan uca model olan ORQA geliştirildi. Her iki bileşen için de BERT kullanıldı. Öğrenilen geri alma yönteminin, bir kullanıcının cevap aradığı veri kümelerinde BM25'e göre 19 puan kadar daha iyi performans gösterdiği gösterildi. Ancak, soruyu soranın cevabı zaten bildiği veri kümelerinde BM25 gibi geleneksel bir IR çerçevesi yeterlidir. Bu makalede SQuAD veri kümesi kullanıldığından, bu model yetersizdir çünkü yazarlar SQuAD veri kümesi için BM25+BERT modelinden daha kötü sonuçlar elde etmişlerdir.

Karpukhin et al.(2020) adlı makalede yazarlar, QA sistemlerinde belge geri alma sorununu ele aldılar çünkü açık domain QA sisteminin performansının belge geri alma kısmına bağlı olduğunu belirttiler. Geleneksel TF-IDF veya BM25 teknikleri yerine yoğun temsilleri kullandılar. Belge geri alma işlemlerinin çalışma akışı, önce metin geçişi için yoğun kodlayıcının uygulanması ve soru için farklı bir kodlayıcının uygulanması şeklindeydi. Bundan sonra, sistem, sorgu vektörüne en benzer vektörlere sahip olan “k” geçiş içeren vektörleri döndürür. Metin geçişleri ve sorular arasındaki benzerlik, kodlanmış vektörlerinin nokta çarpımı kullanılarak hesaplandı. Okuma anlama bölümü için araştırmacılar iki bağımsız BERT ağı kullandı. SQuAD veri kümeleri için en iyi sonuçları BM25 + yoğun geçiş geri alma (DPR) ile BM25 kullanımı sadece ve çoklu eğitim BM25 + DPR karşılaştırılarak elde ettiler. Bununla birlikte, 20 ve 100 belge üzerinde deney yaptılar ve 100 belgenin okuma anlama kısmı için çok büyük olabileceğini düşündüler. Bu dikkate alınması gereken bir konudur.

3. Araştırma Metodolojisi



Şekil 2; İş akış şeması

3.1: Genel Bakış:

Bazı yayınlara bakılmış ve Doküman Geri Alım bölümü için alternatif çözümler incelenmiştir. Birçok makale, belge geri alma aşamasında benzerliği hesaplamak için TF-IDF veya BM25 kullanmayı önermektedir. Sonuçlar, bilgi geri alma aşamasında kullanılan yaygın makine öğrenimi ve makine öğrenimi olmayan tekniklerin TF-IDF, BM25, yoğun geçiş geri alma ve RNN ile LSTM olduğunu göstermektedir.

Bu makalede, belge geri alımı bölümü hedeflenmiştir. Bunun sebebi, en son okuma anlama modellerinin, örneğin BERT ve T5'in, başarılı sonuçlara (yaklaşık %90 doğruluk) sahip olmasıdır. Bu da, eğer belge geri alma modeli doğru bağlamı makine anlama modeline iletebilirse, tüm modelin tatmin edici sonuçlar elde edebileceği anlamına gelmektedir. Mevcut belge geri alma modellerinin sorunları çeşitli nedenlere bağlı olabilir. Bunlardan biri benzerlik hesaplaması olabilir. Geleneksel bilgi geri alma modellerinin çoğu, TF-IDF veya BM25 benzerliği kullanır. Ancak, sadece bu hesaplamaların kullanılması, ilgili bağlamın gözden kaçmasına neden olur. Bunun sebebi, aynı kelimenin farklı anlamlara sahip olabilmesidir; örneğin "elma" hem bir meyve hem de bir şirket olabilir. Bu yüzden tüm içeriğe bakmak daha iyi bir yaklaşımdır. En son araştırmalardan bazıları, bu zorluğu farklı tekniklerle çözmeye çalıştı. Bunlardan biri LSTM'dir. Ancak, bu modellerin sorunu, soruların bu ileri seviye makine öğrenimi tekniğini uygulamak için çok kısa olabileceği ve hesaplama süresinin çok uzun olmasıdır.
Bu makalede, bu sorunu çözmek için farklı bir yaklaşım önerilmiştir. Basit bir ifadeyle, konu kümeleme yöntemi uygulandı ve bu sayede bağlam veri setinin boyutu küçültüldü. Bu şekilde, ilgisiz bağlamların sayısı azaldı ve en büyük sorun kısmen çözüldü. Çünkü bilgi geri alma modellerinin hiçbiri tam olarak bir bağlam döndürmez. Çoğu kez on ve yüze kadar belge döndürürler. Bu kadar çok belge elde etmek, bağlam sayısı daha az ve maliyet açısından daha verimli olduğunda daha kolaydır.

3.2: Veri Seti:

Bu makalede SQuAD v1 (Rajpurkar et al., 2016) veri seti kullanıldı. Bunun sebebi, benzer araştırmacıların benzer deneylerinde bu veri setini kullandığı için daha doğru karşılaştırma sonuçları elde etmektir. Bu veri seti belge geri alma bölümünün eğitim ve testi için kullanılmıştır. Ayrıca okuma anlama bölümünü test etmek için de kullanıldı. SQuAD veri seti, 536 Wikipedia makalesinden toplanan 100.000'den fazla crowdsourced soru/cevap çifti içermektedir. Bu veri setinin dört ana özelliği vardır. İlk olarak, modelin cevabını tahmin etmesi gereken soru bulunmaktadır. SQuAD veri seti Wh(where,what,why vb.) soruları, evet veya hayır soruları gibi çeşitli tiplerde sorular içermektedir. Bu nedenle yapılan işlem daha zor hale gelmektedir ve gerçek bir soruna daha yakın hale gelmektedir. İkinci olarak, sorunun cevabını aradığı bağlamdır. Bağlamın uzunluğu sabit değildir. Bu veri setinde kısa ve uzun metinler bulunmaktadır. Üçüncü olarak, bağlamın başlığıdır. Son olarak ise cevap'tır. Cevap özelliği içinde, cevabın metni ve cevabın bağlamdaki konumu olan bir sözlük bulunmaktadır. Okuma anlama modelini eğitmek için SQuAD v2 (Rajpurkar et al., 2018) kullanılmıştır, ancak bu veri setinin ayrıntılı açıklaması bu araştırmanın kapsamı dışındadır.

3.3: Metin Önişleme:

3.3.1 Metin temizleme:

Makine öğrenimi ve derin öğrenme eğitimi için veri setleri temizlenmelidir. Bu işlemin yapılmasının birkaç nedeni bulunmaktadır. Bunlardan biri, yinelenen değerlerin veri setinden kaldırılmasıdır. Bu temizleme tekniği veri setinin yapısına bağlı olsa da, yapılandırılmamış veri setlerinde uygulanması gerekmektedir çünkü aynı değerlerle eğitim veya tahmin yapmak yanlış doğruluk elde etmenize yol açabilir. Bu temizleme teknikleri uygulanmasına bağlı olarak değişmektedir. Bir örnekle açıklamak gerekirse, veri seti sayısal değerler içeriyorsa, sayısal temizleme teknikleri uygulanır, çünkü bu tip bir veri seti metin temelli değerlere ait olduğu için dil çevirisi teknikleri uygulanamaz.
Bu deneyde kullanılan SQuAD veri seti temizlenmiş bir veri setidir, yani içinde yinelenen veya boş değerler bulunmamaktadır. Bu nedenle bu işlem bu deney için atlanmıştır.

3.3.2 Sembolizasyon (Tokenization):

Sembolizasyon (tokenization) NLP'de yaygın bir aktivitedir. Metni cümleler veya kelimeler gibi daha küçük parçalara bölmek işlemidir. Metnin cümlelere bölünmesine cümle sembolizasyonu, kelime kelime bölünmesine ise kelime sembolizasyonu denir. Makinelerin insan dilini anlaması zordur, yani makineler, ham metnin anlamını anlamak için yeterince akıllı değildir. İşte bu nedenle sembolizasyona ihtiyacımız vardır. Sembolizasyon başlangıçta anlamsız görünebilir. Ancak durum böyle değildir, çünkü metni anlamlı formlara bölmek gerekmektedir. Bir örnek vermek gerekirse, "New Jersey". Bu kelime "New" ve "Jersey" olarak bölünemez. Bu, gerçek anlamından tamamen farklı bir anlama sahip olmasına yol açmaktadır. Sonuç olarak, sembolizasyon NLP'deki en önemli adımlardan biridir.

3.3.3 Stop Kelimelerinin Kaldırılması (Stop-Word Removal):

Kelimeler sembolize edildikten sonra stop kelime kaldırma işlemi uygulanır. İşlem adının açıkladığı gibi, stop kelimelerin kaldırılması için kullanılır. Yaygın stop kelimeler; 'am', 'is', 'I' vb. gibi kelimelerdir. Basit bir ifadeyle, metin anlamına büyük bir etkisi olmayan kelimelerdir. Bu yöntemin bu deney ve benzer deneylerde uygulanma nedeni, benzerlik teknikleri uygulanırken bu kelimelerin modele karışması ve model performansını etkilemesidir.
Bu hedefe ulaşmak için Spacy'nin stop kelime kaldırma yöntemi kullanılmıştır. Aynı amaçla çevrimiçi olarak kullanılabilecek çok fazla kütüphane bulunmaktadır, örnek olarak NLTK bu kütüphanelerden birisidir. Stop kelime listesini görevine göre genişletebilir veya daraltabilir. Ancak bu deneyde bir temel stop kelime listesi kullanılmıştır.

3.3.4 Kök Bulma (Lemmatization):

İngilizce dilinde farklı zamanlarda farklı yazılışlara sahip fiiller bulunmaktadır(Tense'ler). Ancak bu farklı yazılışlar aynı fiili ifade etmektedir. Ayrıca, isimlerin tekil ve çoğul formları da bulunmaktadır, örneğin "kalem - kalemler". Aynı kelime farklı eklere sahip olabilmektedir. Bu nedenle, benzerlik hesaplamalarının kullanıldığı bu ve benzer deneylerde bu dil bilgisi kuralı sorun oluşturabilmektedir. Bu sorunu aşmak için bu deneyde kök bulma (lemmatization) yöntemi kullanılmıştır. Kök bulma işlemi, kelimelerin morfolojik incelemesini dikkate alır. Bunun için bir kapsamlı sözlük kullanarak algoritmanın formu köküne bağlaması gerekmektedir. Yine bu amaçla Spacy kütüphanesi kullanılmıştır.

Ayrıca, kök bulma (lemmatization) benzer amaçlar için kullanılabilecek tek teknik değildir. NLTK'dan gelen "stemming" de aynı amaçla kullanılabilir. Ancak stemming, fiil bitişleriyle ilgili bir soruna sahiptir. Örneğin; "studies" kelimesine stemming uygulandığında sonuç "studi" olurken, "study" olması gerekmektedir. Kök bulma (lemmatization) bu sorunu çözmektedir.

İşte stop word ve kök bulma (lemmatization) uygulanmadan önceki örnek metin:

"*In 2004, worldwide sales of audio CDs, CD-ROMs and CD-Rs reached about 30 billion discs. By 2007, 200 billion CDs had been sold worldwide. CDs are increasingly being replaced by other forms of digital storage and distribution, with the result that audio CD sales rates in the U.S. have dropped about 50% from their peak; however, they remain one of the primary distribution methods for the music industry. In 2014, revenues from digital music services matched those from physical format sales for the first time*"

Stop word ve kök bulma (lemmatization) uygulandıktan sonraki hali:
"*2004 , worldwide sale audio cd , cd - rom CD - Rs reach 30 billion disc . 2007 , 200 billion cd sell worldwide . cd increasingly replace form digital storage distribution , result audio cd sale rate U.S. drop 50 % peak ; , remain primary distribution method music industry . 2014 , revenue digital music service match physical format sale time .*"

3.4 Belge Alımı (Document Retrieval):

3.4.1 Genel Bakış:

Bu deney için çeşitli yöntemler doküman geri getirme için uygulanmıştır. Temel iş akışı, konu kümelemeyle örneklerin daraltılması ve daraltılan bağlam üzerinde TF-IDF'nin uygulanmasıdır. TF-IDF ve BM25 benzerlikleriyle en iyi performans elde edilmiştir. Bu deneyde, TF-IDF benzerliğine ek olarak işlemlerle doğruluk oranının artırılması amaçlanmıştır. İlk olarak, soruya dayalı yalnızca ilgili konuları elde etmek için konu kümelemesi uygulanmıştır. Bu bölümden sonra, en yüksek puanlara sahip olan 10 ve 100 belgeyi elde etmek için TF-IDF benzerliği uygulanmıştır. Bunun nedeni, konu kümelemesi ile programın yalnızca ilgili bağlamları almasıdır. Bu bağlamlara TF-IDF benzerliği uygulanması doğruluğu artırabilir.

3.4.2 Konu Kümelemesi (Topic Clustering):

Konu kümelemesi uygulamak için çeşitli yöntemler bulunmaktadır. Bu deneyde, deneydeki veri kümesinin 'konu' adında bir bilgiye sahip olmadığı için denetimsiz konu kümeleme teknikleri kullanılmıştır. Yalnızca bir 'başlık' bilgisi vardır, ancak konu amacıyla kullanılamamaktadır. Ayrıca, başlık bilgisi olmayan belgelerin olduğu arama motorlarında bu tür sistemler kullanılmaktadır. Bu deneyde K-means kümeleme ve LDA, ek yöntemlerle birlikte kullanılmıştır.

3.4.2.1 LDA:

Konu kümelemesi için uygulanan ilk yöntem Latent Dirichlet Allocation (LDA) adı verilen bir yöntemdir. Benzer konulara sahip belgelerde benzer kelime grupları kullanılır. Bu, LDA'nın nasıl çalıştığına dair basit bir açıklamadır. Konu modellemesi için LDA, konuların kelimeler üzerinde olasılık dağılımları olduğunu ve belgelerin gizli temalar üzerinde olasılık dağılımları olduğunu varsaymaktadır. LDA'nın çalışma akışını daha iyi açıklamak için bir belge kümesi olduğunu varsayalım. Geliştirici tarafından keşfedilecek K sayısındaki sabit konular seçilmelidir. K için doğru bir sayı bulunmamaktadır, kullanıcı farklı K değerlerini denemeli ve en iyi sonucu bulmaya çalışmalıdır, bu da K-Means kümeleme ile aynıdır. Her bir metnin konu temsilcisi ve her konuyla ilişkili terimler LDA'nın öğrenmesi gereken şeylerdir. LDA, her belge üzerinden geçer ve belgedeki her kelimeyi rastgele olarak K konulardan birine atar. Bu rastgele atama ile kullanıcı, her belge için zaten konu temsilleri ve her konu için kelime dağılımları elde etmiştir. Bununla birlikte, bu başlangıçta rastgele atanan konular, ilk iterasyonda anlamlı olmayacaktır. Bir sonraki adımda, iterasyon, her belgedeki her kelime üzerinde birçok kez gerçekleştirilerek konuların geliştirilmesi için devam eder. Her iterasyonda, kelimeler ve belgeler için olasılıklar hesaplanır. Bu sürecin sonunda, her belge bir konuya atanır. K-ortalama kümelemeyle aynı şekilde, bu konular sayılardır. İnsan tarafından okunabilir formatta, ek bir yöntem kullanılmalı ve en üstte yer alan kelime sayısı belirtilmelidir.

LDA modelinin bir tahmin yöntemi olmaması sorun teşkil etmektedir. Bu deneyde, hem LDA hem de K-Means modelleri, bağlam değerleriyle eğitilmiş ve soru değerlerini tahmin etmeleri gerekmektedir. Yukarıda belirtildiği gibi, bunun nedeni, bağlam değerlerinin TF-IDF

benzerlik kısmına geçmeden önce daraltılması gerektiği ve bunun yapılabilmesi için soru konusunun tahmin edilmesi ve yalnızca aynı konuda olan bağlamların seçilmesidir.

Bu sorun için birkaç çözüm yöntemi önerilmiştir. İlk yöntem, metin benzerliğinin kullanılmasıdır. Metin benzerlik modellerinde, giriş ile sınıflandırma etiketleri arasındaki benzerlik, dönüşümcülerin yardımıyla hesaplanmaktadır. HuggingFace platformunda kullanılmak üzere birçok önceden eğitilmiş metin benzerlik modeli bulunmaktadır ve ücretsiz olarak kullanılabilir. Bu deneyde sentence-transformers'dan all-MiniLM-L6-v2 ve paraphrase-MiniLM-L6-v2 kullanılmıştır. LDA ile metin benzerliğini birleştirme işlemi için ilk adım, LDA ile K konu sayısını bulmaktır. Ardından, her konu için en üstte yer alan 10 kelimeyi almak ve bu kelime gruplarını etiket olarak kullanarak soru ile bu etiketler arasındaki metin benzerliğini hesaplamaktır. Son olarak, soru en yüksek benzerlik skoruna sahip kelime grubunun etiketiyle etiketlenir, ki bu aynı zamanda konuyu temsil etmektedir. Bu deneyde K sayısı için 8 seçilmiştir. Bu, her bir K değeri için 10 kelime çarpılarak 8 ile çarpılarak bu deney için 80 etiket olduğu anlamına gelir. Metin benzerliği, soru etiketlerini tahmin etmek için hesaplama açısından maliyetli bir yöntemdir.

İkinci yöntem, zero-shot sınıflandırma kullanmaktır. Zero-shot sınıflandırma modelleri, konu kelimelerinin cümle olmadığı durumlarda daha uygun olabilir. Bu nedenle metin benzerliği düşük bir doğruluk oranına sahip olabilir. Ancak bu, zero-shot için geçerli değildir. Zero-shot, cümle ve tek kelime karşılaştırmalarıyla iyi çalışmaktadır. Metin benzerlik modelinde olduğu gibi HuggingFace platformunda birçok zero-shot sınıflandırma modeli bulunmaktadır. Bu deneyde valhalla'dan distilbart-mnli-12-1 ve facebook'tan bart-large-mnli kullanılmıştır. Her iki model için de çok sınıflı zero-shot sınıflandırma kullanılmıştır, çünkü kelime gruplarında benzer kelimeler vardır ve tek sınıflı sınıflandırma yanlış sonuçlara yol açabilir. Metin benzerlik modelinde olduğu gibi, her soru için 80 etiket vardır. Her bir soru için çok sınıflı zero-shot uygulanır ve soru en yüksek puan alan etiketle etiketlenir. Zero-shot modeli, metin benzerlik modelinden daha yavaştır. Bu, hesaplama açısından daha maliyetlidir.

Üçüncü yöntem, denetimli sınıflandırma algoritmalarının kullanılmasıdır. Yukarıda belirtildiği gibi, bu bir denetimsiz görevdir. Ancak, bu konu kümelemesi için uygulanabilir. Konular bu noktaya kadar zaten kümeleme yapılmış olduğundan, veri setinde bağlam ve soru konularıyla ilgili gerekli bilgiler bulunmaktadır. Bu nedenle görev, denetimsizden denetimliye dönüştürülebilir. Bu amaçla çeşitli sınıflandırma algoritmaları kullanılmıştır: Random Forest Classifier, Support Vector Machine Classifier ve XGBoost Classifier. Bununla birlikte, bilgisayar gücü sınırlamaları nedeniyle bu sınıflandırma modelleri optimize edilememiştir. Her sınıflandırma modeli birkaç parametre alır ve bu parametreler için doğru bir sayı yoktur. Bu parametreler için farklı sayılar belirlenir ve her iterasyonda bu sayıları kullanarak modelin en iyi parametreleri bulmak için farklı kombinasyonlarla model eğitilir. Bunlar denetimli algoritmalar olduğu için beklenen çıktı bilinir ve modelin performansı tahmin edilen ve beklenen çıktıyı karşılaştırarak hesaplanabilir. Bu sürece optimizasyon denir. Veri kümesi eğitim ve test olmak üzere iki farklı kümeye (%80 ve %20) bölünmüştür. Bu, denetimli algoritmalarda model performansını hesaplamak için yaygın olarak kullanılan bir tekniktir. Daha sonra, modeller eğitim veri kümesi ile eğitilir ve bazı parametreler rastgele geçilir. Ancak, veri kümesi metin değerleri içerdiği için, her soru değeri model eğitimine iletilmeden önce vektörleştirilmelidir. Son olarak, model performansı 'predict' metoduyla hesaplanır. Üç sınıflandırma yönteminin kullanılmasının nedeni, bazı algoritmaların farklı türdeki veri kümelerinde iyi çalışabilmesidir. Bu nedenle, performansı ölçmek için birden fazla model kullanmak her zaman daha iyidir.

3.4.2.2 K-means kümeleme:

Daha önce belirtildiği gibi, deneyde birden fazla konu kümeleme yöntemi uygulanmıştır. Konu kümeleme için uygulanan ikinci yöntem K-Means kümelemesidir. K-means kümeleme, en popüler denetimsiz kümeleme tekniklerinden biridir. Kümeleme, gizli desenleri keşfetmek amacıyla benzerliklerine dayanarak öğeleri gruplandırmaktır. Sayısal veya metin tabanlı gibi farklı türdeki görevler için kullanılabilir. K-means tekniği, her bir küme için başlangıç noktaları olarak rastgele seçilen merkezleri kullanan ve merkezlerin konumlarını optimize etmek için tekrarlayan hesaplamaları kullanan bir öğrenme verisini işlemek için kullanır. Merkezlerin stabil olduğu veya iterasyonların önceden belirlenen sayıya ulaştığı durumlarda kümelemeyi geliştirme ve iyileştirme işlemini durdurur.

İlk olarak, K-means algoritması, tüm makine öğrenme algoritmalarında olduğu gibi ham metni giriş olarak kabul etmez. K-means algoritmasına uygun hale getirmek için TF-IDF vektörleştirici uygulanmıştır. TF-IDF vektörleştirici, sublinear_tf, min_df, max_df olmak üzere üç farklı parametreyi kabul eder. Ardından, vektörleştirme işlemi için fit_transform yöntemi kullanılmıştır.
Sonraki adım eğitimdir. K-means algoritması için küme sayısı belirlenmelidir. Ancak, bu bir denetimsiz teknik olduğu için k optimum değeri bulmak zordur. Çünkü denetimli bir teknikte beklenen çıktı bilinir ve modelin gerçek değer ve tahminlere dayalı performansı hesaplanabilir. Bundan sonra, farklı hiperparametreler fonksiyona geçirilir ve en etkili parametreler kaydedilir. Maalesef, bu, denetimsiz öğrenme için geçerli değildir, çünkü etiket kaydedilmez ve beklenen çıktı bilinemez. Bu sorunu aşmak için MiniBatchKMeans kullanılmıştır. Bu yöntem, 1 ile 20 arasındaki k değerleriyle birlikte kullanılmıştır. Optimum değer, keskin düşüş değeri tarafından belirlenir.

Sonrasında, kümeleme sonuçları 'clusters' etiketiyle veri çerçevesine eklenmiştir. Bu, bağlamın o belirli kümeye ait olduğu anlamına gelir. Ancak, bu küme değerleri yalnızca 0'dan k-1'e kadar olan sayılardır. Bu işlemin amacı konu modellemesi olduğundan, konu adlarının belirlenmesi gerekmektedir. Tümü küme/grup bazında en yaygın kelimeleri döndürür. Bu konu kelimelerini almak için 'get_feature_names()' yöntemi kullanılmıştır. Kelime sayısı görevin isteğine bağlıdır ve belirli bir sayısı yoktur, ancak insan tarafından okunabilir amaçlar için on veya on beş olabilecek olası sayılar vardır. Son olarak, bu kelimeler K-means tahmini üzerinde bir etkiye sahip değildir veya tahminler için kullanılamaz, çünkü daha önce belirtildiği gibi, makine öğrenimi algoritmalarının hiçbiri ham metni giriş olarak kabul etmez. Bu sadece insan tarafından okunabilir amaçlar içindir.

Eğitim tamamlandığında, bir sonraki adım tahmindir. Bu durumda, soru değerlerinin tahmin edilmesi gerekmektedir, çünkü tüm bu sürecin amacı, kümelere dayalı olarak ilgili metinleri küçültmektir. Soru metinleri eğitilmiş algoritma tarafından görülmediği için, daha fazla kodlamaya gerek yoktur. Bu sefer, giriş 'transform()' yöntemiyle vektörize edilir ve algoritmanın tahmin yöntemine beslenir. Bu sürecin doğruluğu, soru için küme tahmin değeri ile bağlam için gerçek küme arasını karşılaştırarak hesaplanabilir, çünkü veri çerçevesi bu bilgileri depolar.

LDA modeli için tahmin yapılan soru kümesi için farklı algoritmaların uygulandığı gibi, K-means kümelemesi için de aynı yöntemler uygulanmıştır. LDA modelinin aksine, K-means kümelemesi kendi 'predict' yöntemine sahiptir. Ancak, hangi tahmin yönteminin daha iyi performans göstereceği kimse tarafından bilinmez, denemeden bilemez.

LDA ve K-means kümelemesi ile birlikte, metin benzerliği, sıfır atımlı sınıflandırma ve denetimli sınıflandırma algoritmalarının bir kombinasyonu da uygulandı. Bu prosesi daha iyi açıklamak için, bu yöntemin iş akışı açıklandı. Hem metin benzerliği hem de sıfır atımlı sınıflandırma algoritmaları etiketleme aşamasında bir skorlama yapmıştır. Bu, her etiket için belirli skorların olduğu anlamına gelir. Bu skorlar, eşik değerleri olarak kullanılabilir, yani skor eşik değerinin üzerindeyse, soruya o etiket atanır, değilse diğer filtreleme yöntemleri uygulanır. Bu aşamada farklı yöntemler uygulanmıştır. İlk olarak, sonuçlar metin benzerliği modeliyle filtrelenmiş, ardından sıfır atımlı sınıflandırma ve son olarak K-means tahmini uygulanmıştır. İkinci yöntemde ise sıfır atımlı sınıflandırma ve metin benzerliği filtrelenmiş, ardından K-means tahmini uygulanmıştır.

LDA ve K-means kümelemesiyle birlikte bu yöntemlerin tümü uygulandığında, en iyi sonuç, Random Forest sınıflandırıcısı ile K-means kümelemesinin birleştirilmesiyle elde edildi. Bu sonuçlar sonuçlar bölümünde bulunabilir. Bu, belge alımı bölümünün ilk kısmının tamamlandığı anlamına gelir. Tahmin edilen soru değerleri, ikinci kısım olan TF-IDF benzerlik hesaplama kısmına aktarılma zamanıdır.

3.4.3 TF-IDF Benzerliği:

Bu, sistemin soruyla ilgili belgeleri alıp Okuma Anlama kısmına aktardığı ana kısımdır. Şimdiye kadar, program bağlamı ve soruyu kümelemiştir. Şimdi, sorunun kümesine sahip olan belgelerden ilgili cevapları almanın zamanı gelmiştir. Bu amaçla TF-IDF benzerliği uygulanır. Literatür inceleme bölümünde belirtildiği gibi, TF-IDF benzerliği sistemdeki zayıf noktadır. Ancak, bu deneyde TF-IDF uygulamasından önce yeni bir yaklaşım önerildi. Bu nedenle, bu deneyin amacı, kümeleme algoritmalarının TF-IDF benzerliği ile birlikte belge alımı doğruluğunu artırıp artırmadığını test etmektir.

TF-IDF, Terim Sıklığı - Ters Belge Frekansı'nın kısaltmasıdır. İsmin açıkladığı gibi, iki farklı bölümden oluşur; terim sıklığının hesaplanması ve ters belge frekansının hesaplanması. Bu, başka belgelere kıyasla bir belgedeki kelimelerin ağırlığını veya ilgisini ölçen bilgi erişimi (IR) alanında kullanılmaktadır. Terim Sıklığı, bir terimin belgede ne sıklıkla geçtiğini ölçmektedir. Her belgenin uzunluğu birbirinden farklıdır, bu nedenle bir ifadenin uzun belgelerde daha kısa olanlardan önemli ölçüde daha sık görünebileceği mümkündür. Bu nedenle, belgedeki kelime sayısı belgenin uzunluğuna bölünmektedir.

TF = belgedeki terim sayısı / belgedeki toplam terim sayısı

Her dilde normal kelimelerden daha sık görünen bazı kelimeler vardır. İngilizcede, bunlar 'are', 'was', vb. kelimelerdir. Bu kelimeler, TF kısmı için yanlış hesaplamaya yol açabilir, çünkü sistem, bunlar her belgede en sık geçen kelimeler olduğu için metni bu kelimeler temel alarak sınıflandıracaktır. Bu sorunu önlemek için, TF kısmı IDF kısmıyla çarpılır. Ancak, bu makale zaten başka bir çözüm önermiştir, bu da Stop-kelime kaldırma yöntemidir. Bu yöntem, K-means kümelemesi için de faydalı olmuştur.
IDF, bir kelimenin korpusta ne kadar yaygın veya yaygın olmadığına bakar. Bu bölüm, sık terimleri ağırlıklandırırken nadir olanları ölçeklendirme konusunda sisteme yardımcı olur. IDF aşağıdaki denklemle hesaplanır:

$$\log \frac{Toplam\ Doküman\ sayısı}{İçinde\ Term\ bulunan\ doküman\ sayısı}$$

Bunun sonucu olarak, TF-IDF ,TF * IDF olarak hesaplanır

3.4.4 Soru Cevaplama Modeli:

Bilgi alımı bölümünden K belge alındıktan sonra, K-means kümeleme ile Random forest sınıflandırıcısı ve TF-IDF benzerliği birleştirilerek elde edilen bu belgeler soru cevaplama modeline aktarılır. Bu modelde, birkaç derin öğrenme algoritması bu deneyde uygulanmıştır. Sıfırdan yeni bir derin öğrenme modeli geliştirmek hem hesaplama açısından maliyetli olacak hem de zaman alıcı olacaktır. Ayrıca, bu tür bir modelin en iyi performansına ulaşmak da zordur. Bunun nedeni, bu modellerin çok sayıda farklı parametre ve büyük miktarda veri kümesiyle optimize edilmiş olmasıdır. Her iki süreç de daha fazla zaman gerektirir. Ancak, bu nedenle transfer öğrenme ve önceden eğitilmiş algoritmalar günümüzde çok popülerdir. Çoğu geliştirici, bu modelleri ihtiyaçlarına göre ayarlayarak deneylerinde kullanır. Transfer öğrenimin önemi, yukarıda belirtildiği gibi, büyük miktarda veriyle eğitilmiş olmaları ve bu modellerde kullanılan gizli katmanların optimize edilmiş olmasıdır. Bu nedenle, yüksek hesaplama gücüyle bu tür modelleri eğitmek neredeyse bir ay sürebilmektedir.

Bu deneyde kullanılan transfer öğrenme yöntemlerinden biri T5 algoritmasıdır. T5, NLP görevlerini metne sınırlayarak, öğrenme görevinin puanını belirlemek için incelenen metni sınırlar (Bird, Ekárt ve Faria, 2021). T5, metin özetleme, soru cevaplama ve metin oluşturma gibi çeşitli görevlerle önceden eğitilmiş bir kodlayıcı-dekoder modelidir, her bir görev bir metin-metin formatına dönüştürülür. T5 modelinin mimarisi, 12 katman, 768 gizli katman ve 200 milyondan fazla parametreden oluşur. Bir geliştirici, böyle bir transfer öğrenme algoritmasını kullanmalı ve görevi için uygun hale getirmek için onu ayarlamalıdır çünkü yukarıda belirtildiği gibi, bu modeller farklı türdeki görevler için kullanılır ve bu algoritmayı kullanmanın amacı belirtilmelidir. Bu sürecin iş akışı, veri kümenizi modele iletmek ve görevinize bağlı olarak bazı gizli katmanları dondurmak şeklindedir. Çünkü her görev için tüm gizli katmanlara ihtiyaç duyulmaz.

Bu deneyde T5 modeli fine-tuning edilmiştir. Bununla birlikte, hesaplama sınırlamaları nedeniyle model tatmin edici sonuçlar vermemiştir. Bunun nedenlerinden biri, SQuAD veri kümesinin büyük olması ve bu veri kümesiyle modelin fine-tuning yapılması zaman alıcı olmasıdır. SQuAD veri kümesinin küçük bir bölümünün kullanılması bir çözüm olabilir, ancak bu alt uyum sorununa neden olabilir.

Bu deneyde kullanılan bir sonraki yöntem BERT - Dil Anlama için Derin Çift Yönlü Dönüşümlü Önceden Eğitme (Devlin et al. 2019)'dir. Yazarlara göre, BERT modeli SQuAD v1.0 ve SQuAD v2.0 ile birlikte soru cevaplama da dahil olmak üzere on bir doğal dil işleme görevinde yeni en iyi sonuçları elde etmektedir. BERT, her bir giriş ve çıkış öğesinin bağlantılı olduğu ve aralarındaki ağırlıkların bu ilişkiye bağlı olarak dinamik olarak belirlendiği Transformers adlı bir derin öğrenme modeline dayanmaktadır. 24 katmanı, 1024 gizli katmanı ve 340 milyon parametresi bulunmaktadır. Bu deneyde, BERT modeli fine-tuning edilmiştir. Bu işlem, T5 modeline kıyasla nispeten daha az zaman alır. Bununla birlikte, sonuçlar hala tatmin edici değildir.

Bu deneyde denenen son yöntem önceden eğitilmiş hugging face kütüphanelerini kullanmaktır. İki kütüphane kullanıldı: deepset'ten roberta-base-squad2 ve deepset'ten xlm-roberta-base-squad2. Her ikisi de SQuAD veri kümesi sürüm 2 ile eğitilmiştir, bu da her iki modelin de bu deney için uygun olduğu anlamına gelir. Roberta-base-squad2'nin F1 puanı, yalnızca İngilizce değerlerinden oluştuğu için xlm-roberta-base-squad2'den daha yüksektir, bu da bu araştırmada kullanılan dilin İngilizce olmasıdır. RoBERTa, Robustly Optimized BERT Pretraining Approach (Liu ve diğerleri, 2019) kısaltmasıdır. Yazarlar, BERT'in önemli ölçüde

yetersiz eğitildiğini ve yayınlanmasından sonra her modele eşit veya daha iyi performans gösterebileceğini bulmuşlardır. Bu nedenle RoBERTa'yı önermişlerdir.

3.5.Değerlendirme Ölçütleri:

Bu deneyde sonuçları değerlendirmek için F1 puanı ve doğruluk (accuracy) kullanılmıştır, çünkü görev türü sınıflandırmadır ve tahmin edilen cevabın gerçek cevapla eşleşip eşleşmediği belirlenmek istenmektedir.

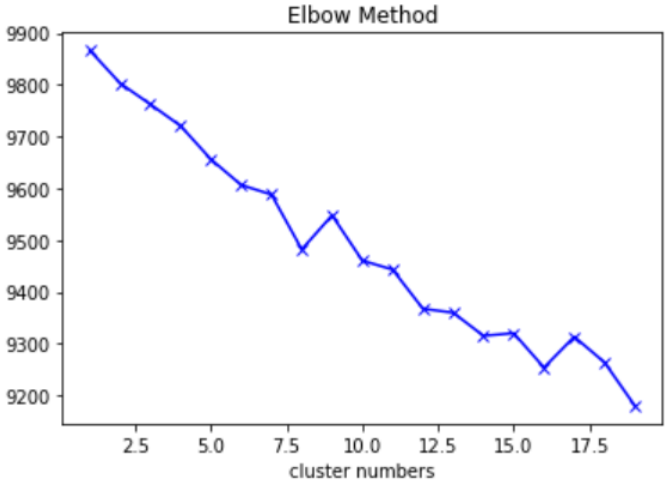
Doğruluğu hesaplamak için toplam doğru tahmin sayısını toplam tahmin sayısına bölebiliriz.

F1 puanını hesaplamak için ise:
2 * (Hassasiyet * Geri Çağırma) / (Hassasiyet + Geri Çağırma)

Hassasiyeti hesaplamak için, doğru pozitiflerin sayısını toplam pozitif sayısına böleriz.
Geri çağırma (recall) hesaplamak için ise, doğru pozitiflerin sayısını doğru pozitiflerin sayısı ve yanlış negatiflerin sayısına böleriz.

4.Sonuçlar:

Daha önce belirtildiği gibi, QA sistemleri iki kısımdan oluşur. Ancak, bu deneyde görev üç bölüme ayrılmıştır; konu kümeleme, TF-IDF benzerliği ve okuma anlama. Konu kümelemesi için K-means ve Random forest sınıflandırıcısı kullanılmıştır. K-means algoritmasını eğitmek için 'cleaned_context' sütunu TF-IDF vektörleştirici'nin fit_transform işlevi ile kullanılmıştır, çünkü K-means algoritması ham metni giriş olarak kabul etmemektedir. Bundan sonra, optimum K değerini bulmak için MiniBatchKMeans uygulanmıştır.



Şekil 3 Optimum K değerleri

```
9     2195
12     839
14     674
11     668
4      630
16     571
1      566
0      562
13     535
7      472
8      398
3      381
2      368
6      338
5      305
10     254
15     244
Name: clusters, dtype: int64
```

Şekil 4 Cluster toplamları

	level_0	index	id	title	context	question	answers	cleaned_context	cleaned_question	clusters
0	2216	2216	56cc52186d243a140015ef03	Sino-Tibetan_relations_during_the_Ming_dynasty	Kublai Khan did not conquer the Song dynasty i...	What did Khubilai claim for a while?	{'text': ['universal rule'], 'answer_start': [...	Kublai Khan conquer Song dynasty South China 1...	Khubilai claim ?	5
1	195	195	573383e94776f41900660c5a	University_of_Notre_Dame	Besides its prominence in sports, Notre Dame i...	Where among US universities does Notre Dame rank?	{'text': ['among the top twenty'], 'answer_sta...	prominence sport , Notre Dame large , - year ,...	university Notre Dame rank ?	1
2	898	898	56d4eb762ccc5a1400d83352	Beyoncé	Beyoncé has received numerous awards. As a sol...	When did Beyoncé receive the Legend Award?	{'text': ['the 2008 World Music Awards'], 'ans...	Beyoncé receive numerous award , solo artist s...	Beyoncé receive Legend Award ?	4
3	4700	4700	56ce6aabaab44d1400b88777	To_Kill_a_Mockingbird	The origin of Tom Robinson is less clear, alth...	What was the name of the black teenager that T...	{'text': ['Emmett Till'], 'answer_start': [870]}	origin Tom Robinson clear , speculate characte...	black teenager Tom Robinson supposedly base ?	9
4	8999	8999	5733703c4776f41900660ad7	Financial_crisis_of_2007%E2%80%9308	In September 2008, the crisis hit its most cri...	When did the financial crisis hit its most cri...	{'text': ['September 2008'], 'answer_start': [3]}	September 2008 , crisis hit critical stage . e...	financial crisis hit critical stage ?	13

Şekil 5 Data setin K-Means kullanıldıktan sonraki durumu

K-means algoritmasının yalnız başına doğruluk oranı yaklaşık olarak %60 olarak hesaplandı. Hesaplama gücü eksikliği nedeniyle deney için 10000 satır seçildi. Deney ortamı olarak Jupyter Lab kullanıldı ve bu ortamda metnin kök çıkarması ve lemmatizasyonu yaklaşık 20 dakika sürdü. Satır sayısı arttırıldığında, RC bölümünden önce mevcut RAM'in tükenmesi problemi ortaya çıktı. %60 doğruluk oranı bu adım için tatmin edici değildi. Bu nedenle farklı metodolojiler uygulandı. Metin benzerliği ve K-means'in birleşimi yaklaşık %65 doğruluk sağlarken, zero-shot sınıflandırma ve K-means'in birleşimi yaklaşık %50 doğruluk sağladı. En iyi sonuç, K-means ve Random Forest sınıflandırıcının birleşimi ile elde edildi. Bu metodolojinin doğruluk oranı %90'a yakındı.

Bu aşamadan sonra, TF-IDF benzerliği yardımıyla K adet belge alındı. TF-IDF benzerliği uygulanmadan önce, aday sorunun kümesi Random Forest sınıflandırmasının tahmin işleviyle belirlendi. Kümenin tahmin edilmesiyle birlikte, bu küme için benzersiz bağlam değerleri alındı. Daha sonra, soru ile her bir bağlam arasındaki benzerlik hesaplandı ve en yüksek puan alan 10 belge döndürüldü. Doğruluğu, bu en iyi 10 belgenin gerçek bağlamı içerip içermediği şeklinde hesaplandı. Bu deneyde rastgele 500 değer kullanıldı. Doğruluk ölçümü için sadece 100 soru değeri kullanıldı, çünkü bilgisayar gücü eksikliği vardı. İlk olarak, 1000 rastgele değer denendi, ancak RAM yetersiz olduğu için program bağlantısı kesildi. Bu işlem birkaç kez uygulandı ve sonuçlar şu şekildeydi: %84, %86, %84.5. Bu nedenle, belge alımı bölümü

için beklenen doğruluk oranı tüm veri seti için %80-90 arasındadır. Bu sonuçlar, çoğu state-of-the-art modelle rekabet edebilir.

Son olarak, RC bölümünün doğruluğu hesaplandı. İlk olarak, belge alımı süreci tamamlandı ve doğruluğu hesaplandı. 10 belge alındıktan sonra RC modeli aday soru ve bu belgeler üzerinde uygulandı. Her iterasyonda, model olası cevapları ve puanları alır.

Bu özellikler bir sözlükte saklandı, böylece daha sonra en yüksek puanlı cevap bulunabilirdi. En yüksek puanlı cevap bulunduktan sonra, tahmin edilen cevap ile gerçek cevap arasındaki benzerlik, tüm-distilroberta-v1 adlı modelin yardımıyla hesaplandı. Bunun nedeni, bazı tahmin edilen cevapların beklenen cevapla aynı olmamasıdır. Bununla birlikte, bu, programın yanlış cevap tahmin ettiği anlamına gelmez. Eğer 'Gri Kurt' ve 'Kurtlar' gibi çok benzerlerse, program bunları doğru olarak sayar. Eşik değeri 0.5 olarak belirlendi, ancak bu daha fazla araştırma gerektirir.
Sınırlı hesaplama gücü nedeniyle bu deneyde yalnızca 8 soru değeri test edildi. Program 8 sorunun 7'sini doğru tahmin etti, bu da doğruluk oranını 0.875 yapar.

```
  self._setitem_single_block(indexer, value, name)
The correct answer: ['universal rule']
Maximum value: universal rule
<ipython-input-43-90ab594da6b2>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['context'][index] = doc
/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
The correct answer: ['among the top twenty']
Maximum value: top twenty
<ipython-input-43-90ab594da6b2>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['context'][index] = doc
/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
The correct answer: ['the 2008 World Music Awards']
Maximum value: 2008 World Music Awards
```

Şekil 6 RC performansının gösterilmesi

5. Tartışma:

Bu araştırmada, QA sistemlerinin belge alımı kısmının iyileştirilmesi hedeflenmiştir. Benzer araştırmalar daha önce uygulama yapmadan TF-IDF benzerliği kullanmıştır. Bu nedenle, bu deneyde bu soruna yeni bir yaklaşım önerilmiştir. Bu araştırmada kullanılan metodoloji, IR kısmı için state-of-the-art modelin doğruluk oranına benzer bir doğruluk oranına sahiptir. Ancak, daha önce belirtildiği gibi, TF-IDF benzerliği cümle içindeki anlamını önemsemeden kelimenin frekansını hesaplar. Bu nedenle, ileri düzey NLP tekniklerinin konu kümeleme ile birleştirilmesi gelecekte değerlendirilmelidir.

6. Son:

Açık alan QA sistemleri, çevrimiçi olarak büyük miktarda veri artışı nedeniyle zorlu işlemlerdir ve dikkat gerektirmektedir. Bu deneyde, belge alımı bölümü için yeni bir yaklaşım önerilmiştir. Literatür taramasına dayanarak, state-of-the-art modelinin IR kısmı için yaklaşık %80 doğruluk oranı elde ettiği belirtilmiştir, oysa bu model rastgele seçilen 500 değer için %84 doğruluk oranındadır. Bu modelin tüm veri setine uygulandığında belirgin bir düşüş beklenmemektedir, çünkü rastgele seçilmiştir, ancak %2-3'lük bir düşüş beklenmektedir. Bununla birlikte, bu hala state-of-the-art performansıdır ve hesaplama gücü eksikliği nedeniyle K-means ve Random forest sınıflandırıcının parametreleri optimize edilememiştir. Bu optimizasyonlar yapıldığında ve model tüm veri setine uygulandığında, IR kısmı için beklenen doğruluk oranı %83-85 arasındadır. Bu modelin en büyük avantajı, RC kısmının, doğru bağlama en üstte 10 belge içerisinde yer aldığında, nadiren sorulara cevap verememesidir.

Bu deney, K-means kümeleme ve Random forest sınıflandırıcının birleşiminin TF-IDF benzerlik performansını, daralan bağlam veri setiyle iyileştirdiğini kanıtlamaktadır. Daha önce belirtildiği gibi, gelecekteki çalışmalarda, K-means ve Random forest algoritmaları için optimizasyon tekniklerinin uygulanması gerekmektedir. Bunun yanı sıra, TF-IDF benzerliğinin yerine geçecek daha ileri düzey benzerlik teknikleri kullanılmalıdır. Bunun nedeni, TF-IDF benzerliğinin kelimelerin anlamını aramak yerine kelime frekansını kullanmasıdır. Bu bölüm için planlanan, NLP görevlerinin çoğu için state-of-the-art bir model olan BERT algoritmalarını kullanmaktır. RC kısmı, doğru bağlam alındığında başarılı bir şekilde gerçekleştirildiği için daha az dikkat gerektirir.

7.Kaynakça:

Alberti, C., Lee, K. and Collins, M. 2019. *A BERT Baseline for the Natural Questions*. [online] Available at: https://arxiv.org/pdf/1901.08634.pdf [Accessed 13 Sep. 2022].

Chen, D., Fisch, A., Weston, J. and Bordes, A. 2017. *Reading Wikipedia to Answer Open-Domain Questions*. [online] Available at: https://arxiv.org/pdf/1704.00051.pdf [Accessed 13 Sep. 2022].

Clark, C. and Gardner, M. 2017. *Simple and Effective Multi-Paragraph Reading Comprehension*. [online] Available at: https://arxiv.org/pdf/1710.10723.pdf [Accessed 13 Sep. 2022].

Das, R., Dhuliawala, S., Zaheer, M. and Mccallum, A. 2019. *MULTI-STEP RETRIEVER-READER INTERACTION FOR SCALABLE OPEN-DOMAIN QUESTION ANSWERING*. [online] Available at: https://arxiv.org/pdf/1905.05733.pdf [Accessed 13 Sep. 2022].

Devlin, J., Chang, M.-W., Lee, K., Google, K. and Language, A. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. [online] Available at: https://arxiv.org/pdf/1810.04805.pdf [Accessed 13 Sep. 2022].

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.-T. and Ai, F. 2020. *Dense Passage Retrieval for Open-Domain Question Answering*. [online] Available at: https://arxiv.org/pdf/2004.04906.pdf [Accessed 13 Sep. 2022].

Kratzwald, B. and Feuerriegel, S. 2018. *Adaptive Document Retrieval for Deep Question Answering*. [online] Available at: https://arxiv.org/pdf/1808.06528.pdf [Accessed 13 Sep. 2022].

Kratzwald, B., Eigenmann, A. and Feuerriegel, S. 2019. *RankQA: Neural Question Answering with Answer Re-Ranking*. [online] Available at: https://arxiv.org/pdf/1906.03008.pdf [Accessed 13 Sep. 2022].

Lee, J., Yun, S., Kim, H., Ko, M. and Kang, J. 2018. *Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering*. [online] Available at: https://arxiv