

TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

EĞİTİM TEKNOLOJİLERİ YARIŞMASI

PROJE DETAY RAPORU

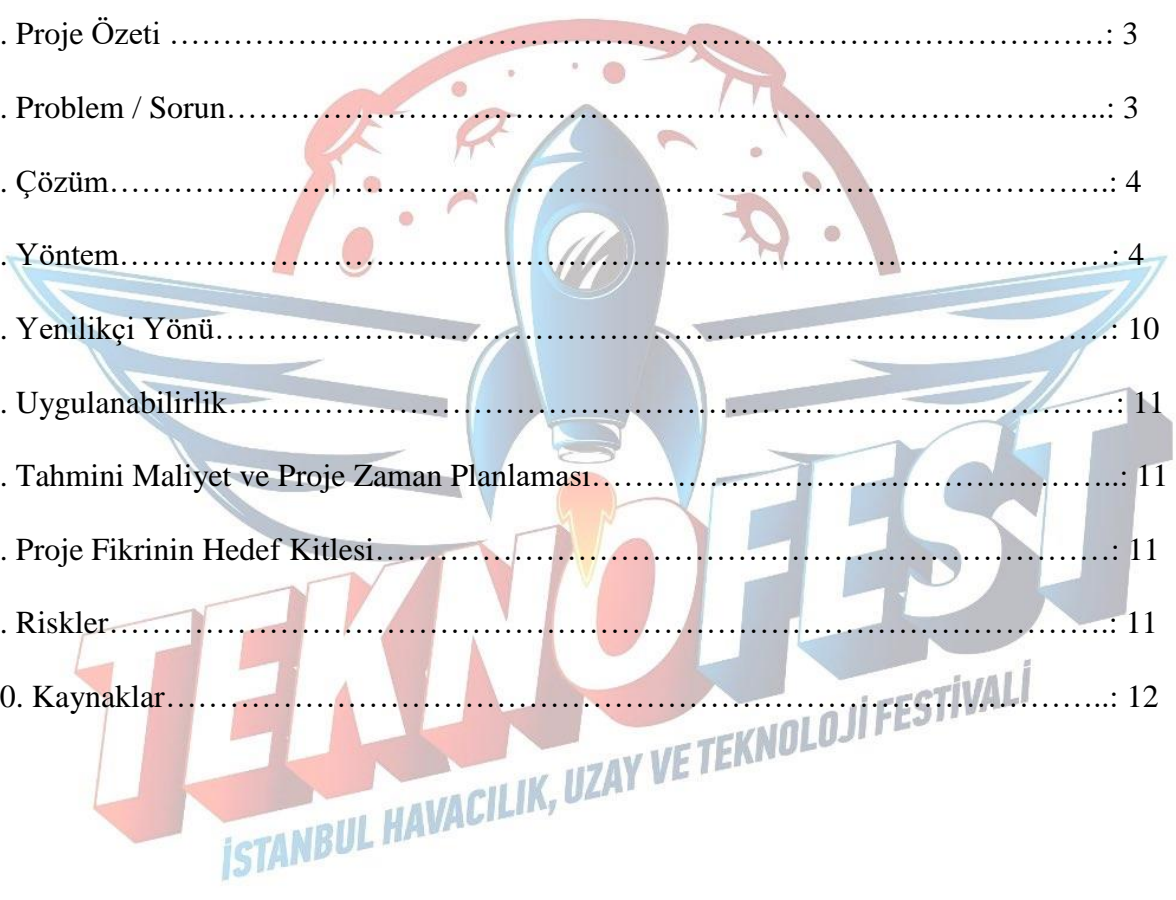
AÇIK UÇLU SORULARA AKILLI PUANLAMA

LAUTECH

TEKNOFEST
İSTANBUL HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ
50555

İçindekiler

1. Proje Özeti	3
2. Problem / Sorun.....	3
3. Çözüm.....	4
4. Yöntem.....	4
5. Yenilikçi Yönü.....	10
6. Uygulanabilirlik.....	11
7. Tahmini Maliyet ve Proje Zaman Planlaması.....	11
8. Proje Fikrinin Hedef Kitlesi.....	11
9. Riskler.....	11
10. Kaynaklar.....	12



1. Proje Özeti (Proje Tanımı)

Proje açık uçlu soruların cevaplarını puanlayan bir yapay zeka uygulamasıdır. Uygulamayı yazarken Python dili, yapay zeka kütüphanesi olarak Tensorflow kütüphanesi kullanılmıştır. Derin öğrenme modeli olarak da bir yinelenen sinir ağı çeşidi olan Gated Recurrent Unit (GRU) kullanılmıştır. Projede öznitelik çıkarımı yöntemi olarak kelime vektörleri kullanılmıştır. Proje türetilmeye hazır bir prototiptir. Uygulama, örneklem olarak kullanılan soruya verilen açık uçlu cevapları doğru ve yanlış olarak nitelendirmekte ve 0 ile 100 arasında puan vermektedir. Uygulama eğitim safhasında hiç kullanılmayan test setinde % 92 başarı sağlanmıştır. Bulgular gösteriyor ki daha büyük bir veri setinin kullanılması halinde bu oranın artacağı düşünülmektedir.

Bu projede genellikle sıralı verilerde kullanılan yinelenen sinir ağlarını kullanılmıştır. Yinelenen sinir ağı çeşidi olarak da Gated Recurrent Unit (GRU) kullanılmıştır. Oluşturulan sinir ağı dört katmandan oluşmaktadır. Sinir ağının ilk katmanı olan Embedding Layer, daha önceden hazırlanan veriyi kelime vektörlerinden oluşan dizilere dönüştürmektedir. Sinir ağının ikinci ve üçüncü katmanları GRU'lerden oluşmaktadır, bu katmanlar verilerden özellikleri çıkarmaktadır. Sinir ağının son katmanı, ona daha önceki katmandan gelen özellikleri, Sigmoid aktivasyon fonksiyonunu kullanarak doğru ya da yanlış olarak sınıflandırmaktadır. Sinir ağında kayıp fonksiyonu olarak iki sınıflı (binary) veri setlerinde kullanılan Binary Cross Entropy kayıp fonksiyonu ve Adam optimize edicisi kullanılmaktadır.

Açık Uçlu Soruların Cevaplarını Yapay Zeka İle Değerlendirme Araştırma Projesi Prototipidir.

ÇEVİRİMİÇİ TÜRKİYE GENELİ COĞRAFYA SINAVI

Soru 1:
Dünya 4. ocakta güneşe en yakın olduğu pozisyonundadır. Buna rağmen bu dönemin kuzey yarımkürede hava sıcaklarının en düşük olduğu dönem olmasının sebebi nedir?

Eksen eğikliği nedeniyle

Cevabı Gönder

Puanın
Bu cevap 100.0% doğru

Doğruluk yüzdesi, cevabın doğru olma olasılığını gösterir. Bu yüzden %50'den büyük sonuçlar doğru kabul edilecektir.

2. Problem/Sorun:

Lautech takımı olarak lisede öğrenim görmekteyiz ve çözmek istediğimiz problemi kendi gözlemimizle anlatmak isteriz. Ülkemizde seçme sınavlarında ölçme aracı olarak çoktan seçmeli testler kullanılmaktadır. Üniversite sınavını ele alacak olursak; sınav çoktan seçmeli olduğundan bir alt kademe olan lise öğretmenleri sınavları çoktan seçmeli test soruları kullanarak hazırlamaktadır. Çünkü lise dönemi bir yandan da üniversiteye hazırlık dönemidir. Fakat çoktan seçmeli testler sınava giren kişinin akıl yürütme, problem çözme, ilişki kurma, yorum yapma, yaratıcı ve eleştirel düşünme gibi üst düzey bilişsel becerileri kazanıp kazanmadığını ölçmemekte. Çoktan seçmeli

testlerle ölçülen bir süreç ise kişiyi öğrenmekten ziyade ezberlemeye yönlendirmektedir. Oysa açık uçlu maddelerde öğrenci cevabı kendisi yapılandırmakta, cevabının gerekçelerini açıklama fırsatı bulmakta ve düşüncelerini daha özgür bir biçimde ifade edebilmektedir (Gronlund, 1998). Açık uçlu maddeler bu yönüyle; problem çözme, problemleri organize etme, yeni ve orijinal fikirler üretme, fikirleri değerlendirme, bilgileri değişik durumlarda işe koşma, neden-sonuç ilişkileri kurma, genellemeler yapma, hipotez üretme, alternatifler arasında karşılaştırmalar yaparak bir yargıya varma gibi üst düzey becerilerin ölçülmesi için en uygun soru türüdür (Tan ve Erdoğan, 2004).

Genel sınavlarda açık uçlu soruların kullanılmaması eğitimin temel amaçlarından olan, bireyin problem çözme becerisi kazanmasına engel olmaktadır.

3. Çözüm

Problemin çözümü ise genel sınavlarda açık uçlu sorular kullanmaktır. Fakat sübjektif puanlama yapabilmek için, değerlendiriciler arasında yorum farkını engellemek yani bütün cevapların tek bir değerlendirici tarafından değerlendirilmesi gerekmektedir. Tabiki değerlendirme süresinin de kısa olması gerekmektedir. Biz de bunu bir insan yapamayacağına göre neden bir makine yapmasın dedik ve açık uçlu soruların cevaplarını değerlendirebilen bir uygulama yazdık. Böylece açık uçlu değerlendirme ölçeği puanlamasının zaman alıcı ve objektif olması dezavantajları ortadan kaldırılmış olacak.

Ülkemizde ehliyet sınavları e-sınav merkezlerinde online olarak yapılmaktadır. Uygulamamız e-sınav sistemine entegre edilebilir ve kullanılabilirliği test edilebilir. Eğer testten olumlu sonuç çıkarsa diğer sınavlara da yaygınlaştırılabilir.

Uygulamamız sayesinde, her alanda problemlerimizi çözerek hayatımıza giren yapay zeka eğitim alanında da büyük bir soruna çözüm getirmiş olacaktır. Temel beceri ve kavramların yanında, muhakeme yapma ve matematiksel düşünme becerileri de ölçülen öğrenciler, sınava hazırlık sürecini sadece ezber yaparak geçirmek yerine üst düzey bilişsel becerilerini geliştirecek hazırlıklar da yapacaklardır. Bu da eğitim sisteminin ezberden gerçek öğrenmeye evrilmesine sebep olacaktır.

Sorun	Çözüm	Eğitimdeki Katkısı
Genel sınavlarda çoktan seçmeli testlerin kullanılması	Genel sınavlarda açık uçlu soruların kullanılması	Öğrencilerin üst düzey bilişsel becerilerinin ölçülebilmesi

4. Yöntem

Prototipimizi “Dünya 4 ocakta güneşe en yakın olduğu pozisyonudadır. Buna rağmen bu dönemin kuzey yarım kürede hava sıcaklarının en düşük olduğu dönem

olmasının sebebi nedir?" sorusuna üzerinden oluşturduk. Sorumuzu Google Formlar anketi ile farklı Whatsapp gruplarında paylaştık, ulaşabildiğimiz kişilere yüz yüze uyguladık. Toplamda elde ettiğimiz 433 cevabı doğru ve yanlış olarak etiketledik. Ardından, modeli eğitmeden sonra test edebilmek için 50 cevaptan ve cevapların etiketlerinden oluşan bir test seti oluşturduk.

Topladığımız verideki alfabetik ve sayısal karakterler hariç her şeyi çıkardık ve tüm harfleri küçük harf haline getirdik. Topladığımız veriyi kullanarak modeli eğittik. Test setinde yapılan tahminlerin doğruluk yüzdesini ve karışıklık matrisini inceledik, karışıklı matrisi dengeliydi, doğruluk yüzdesi ise yetersizdi.

Biz de elimizdeki veri setini büyütmek için kelime vektörü yaklaşımıyla veri setindeki kelimelerin alternatiflerini tespit ettik ve elimizdeki cümlelerde bulunan kelimelerle değiştirdik.

Veri setini bu şekilde büyütmek için Türkçe Wikipedia üzerinde bir kelime vektörü eğittik. Eğittiğimiz kelime vektörü modelini kullanarak veri setinde bulunan her kelimeye en yakın üç kelimeyi tespit ettik, tespit ettiğimiz kelimeleri JSON formatında kaydettik. Oluşturduğumuz JSON dosyasını kullanarak yakın kelimeleri değiştirme yöntemiyle veri arttıracak bir sınıf yazdık, bu sınıfı kullanarak veri sayısını 1433'e çıkardık. Daha sonra veri setinde bulunan her kelimeyi bir tam sayı değerle eşleştirdik ve veriyi modelde kullanmaya hazır hale getirdik.

```

# Metin dosyasını okudum ve bir değişkene kaydettim.
with open("../input/turkish-wikipedia-dump/wiki_00",mode="r",encoding="UTF-8") as F:

    data = F.read()

print(data[:300])

# Metnin içinde bulunan HTML etiketlerini temizleyen bir fonksiyon tanımladım
# ve metni HTML etiketlerinden temizledim.
defhtml_cln(raw_html):
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, '', raw_html)
    return cleantext

data =html_cln(data)

pattern = "[^a-zA-Z0-9öçşüğöçşüğü]"

# Metni cümlelere ayırdım
sentence_data = nltk.sent_tokenize(data)

# Cümlelerden harfler ve sayılar hariç tüm karakterleri temizledim
sentence_data = [re.sub(pattern, " ",sentence) for sentence in sentence_data]

# Cümleleri kelimelere ayırdım
tokenized_data = [nltk.word_tokenize(sentence) for sentence in sentence_data]

print(tokenized_data[0])

from gensim.models import Word2Vec

# Kelime vektörü modelini eğittim
model = Word2Vec(sentences=tokenized_data>window=3,min_count=1,workers=4,sg=1)

# Eğittiğim modeli kaydettim
model.wv.save_word2vec_format('model.bin')
```

```

class SynonymGenerator():
    def __init__(self,json_path):

        with open(json_path,mode="r",encoding="ISO-8859-9") as F:
            # JSON dosyasını okudum.
            self.syn = json.load(F)

    def generate_n_texts(self,mount_texts,n_generate):

        text_generated = 0
        generated_texts = []

        while text_generated != n_generate:

            while True:

                # Elimdeki veriden rastgele bir cevap seçtim
                selected_sentence = random.choice(mount_texts)

                # Eğer cümle'nin uzunluğu 4 kelimedenden kısaysa
                # seçilen cümle ile devam etmek için döngüyü kırdım.
                if len(nltk.word_tokenize(selected_sentence)) > 4:
                    break

                # Rastgele seçilen cümleyi kelimelerin ayırdım
                tokens = nltk.word_tokenize(selected_sentence)

                # Rastgele seçilen cümleden rastgele iki adet kelime seçtim
                token1 = random.choice(tokens)
                token2 = random.choice(tokens)

```

```

            while True:

                try:
                    # Seçtiğim kelimelere en yakın kelimelerden rastgele
                    # bir tane seçtim
                    token1_syn = random.choice(self.syn[token1])
                    token2_syn = random.choice(self.syn[token2])
                    break
                except:
                    token1 = random.choice(tokens)
                    token2 = random.choice(tokens)

                # Seçtiğim kelimeler ile onlara yakın kelimeleri değiştirdim
                # Oluşturulan yeni cümleyi listeye ekledim
                generated_sentence = selected_sentence.replace(token1,token1_syn)
                generated_sentence = generated_sentence.replace(token2,token2_syn)

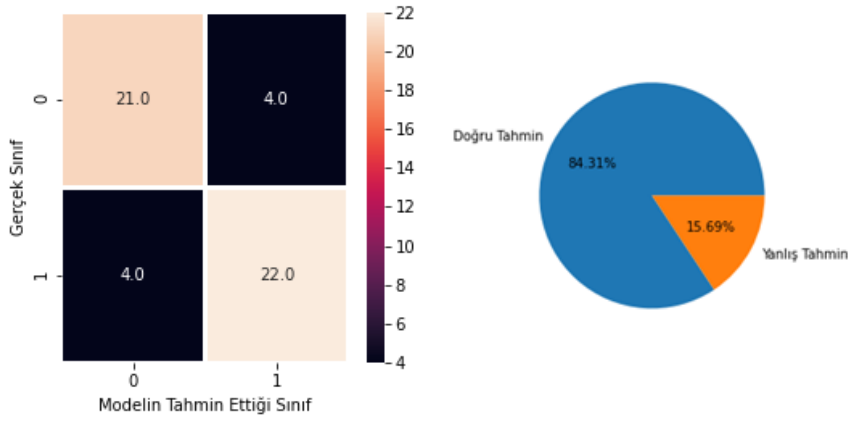
                generated_texts.append(generated_sentence)

                text_generated += 1

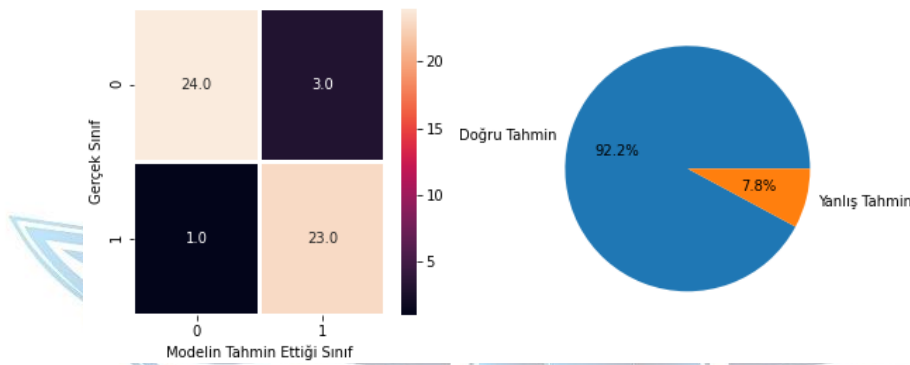
            return generated_texts

datagen = SynonymGenerator("/content/syn_end.json")
x_generated_true = datagen.generate_n_texts(x_true.values,500)
x_generated_false = datagen.generate_n_texts(x_false.values,500)

```



Veri artırma işlemi yapılmadan eğitilen modelin test sonucu



Veri artırma işlemi yapıldıktan sonra eğitilen modelin test sonucu

Modeli tüm veri setini kullanacak şekilde toplam beş tur eğittik. Eğitim yaparken tüm veri 64'lü parçalara böldük ve parçaları ayrı ayrı olacak şekilde eğitimde kullandık. Model her parçanın sonunda kaybı hesapladığı ve parametreleri güncellediği için daha başarılı bir şekilde eğitilmiş oldu.

```
# tokenizing texts
from tensorflow.keras.preprocessing.text import Tokenizer

# Tokenizer nesnesini oluşturdum
tokenizer = Tokenizer()

tokenizer.fit_on_texts(x_end)

# Toplam farklı kelime sayısını bir değişkene kaydettim.
NUM_TOKEN_WORDS = len(tokenizer.word_counts)

# Kelimeleri sayısal karşılıkları ile eşleştirerek sayı dizilerine
# dönüştürdüm.
x_tokens = tokenizer.texts_to_sequences(x_end)

# Her sayı dizisini aynı uzunluğa getirmek için en uzun cevabı
# belirledim.
max_len = max([len(text) for text in x_tokens])

from tensorflow.keras.preprocessing.sequence import pad_sequences

# Kısa cevapların başına sıfır ekleyerek tüm cevapları aynı
# uzunluğa getirdim
x_tokens_pad = pad_sequences(x_tokens, maxlen=max_len)
```

```
# Vector size deęişkeni Embedding katmanından çıkacak vektörlerin boyutunu belirler
VECTOR_SIZE = 100
# Token size deęişkeni Embedding layer'a gelen sayı dizilerinin uzunluęunu
# belirler
TOKEN_SIZE = max_len

model = Sequential()

# Bu katman kelime vektörlerinden oluşan diziler oluşturur
model.add(layers.Embedding(input_dim = VOCAB_SIZE,
                           output_dim = VECTOR_SIZE,
                           input_length = TOKEN_SIZE))

# Yinelenen sinir aęı katmanı 1
model.add(layers.GRU(128,activation="relu",return_sequences=True))

# Yinelenen sinir aęı katmanı 2
model.add(layers.GRU(128,activation="relu",return_sequences=False))

# Çıktı katmanı, sigmoid aktivasyon fonksiyonu sayesinde 0 ve 1 arasında bir deęer
# döndürür.
model.add(layers.Dense(1,activation="sigmoid"))

# Modelin inşasını tamamladım.
model.compile(optimizer="Adam",loss="binary_crossentropy",metrics=["accuracy"])

model.summary()
```

Modelin başarısını ölçmek için elli cümleden oluşan bir test seti oluşturduk. Model bu cümleleri tek tek sınıflandırdı. Modelin tahminleri kullanılarak başarı oranını hesapladık. Sonuç %90 çıktı.

TEKNOFEST
İSTANBUL HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ


```

from sklearn.metrics import confusion_matrix
import seaborn as sns
import warnings as wrn
wrn.filterwarnings('ignore')

# Bu fonksiyon kendisine verilen veri setine göre modeli değerlendirir.
def evaluate_model(testDataset):

    # Cevapları ve etiketleri ayrı değişkenlere aktardım
    x_test = testDataset["soru"]
    y_test = testDataset["etiket"]

    # Cevapları temizledim ve eğitim aşamasında yaptığım gibi
    # sayı dizileri haline getirdim.
    x_test = [re.sub(pattern," ",text) for text in x_test]
    x_test = [text.lower() for text in x_test]
    x_test = tokenizer.texts_to_sequences(x_test)
    x_test = pad_sequences(x_test,maxlen=max_len)

    # Modelin evaluate fonksiyonunu kullanarak modeli test ettim
    eval = model.evaluate(x_test,y_test)

    print("Test accuracy of model is {}".format(eval[1]))
    print("Test loss of model is {}".format(eval[0]))

    # Karışıklık matrisini oluşturmak için test setini modele tek tek
    # tahmin ettirdim
    y_pred = model.predict_classes(x_test)

    # Karışıklık matrisini oluşturdum ve gösterdim
    conf_matrix = confusion_matrix(y_true=y_test,y_pred=y_pred)
    plt.figure(figsize=(4,4))
    sns.heatmap(conf_matrix,annot=True,fmt=".1f",linewidths=1.5)
    plt.ylabel("Actual Class")
    plt.xlabel("Predicted Class")
    plt.show()

    # Tahmin sonuçlarını gerçek sonuçlarla karşılaştırmak için
    # Virgülle ayrılmış değerler (.csv) dosyası olarak kaydettim.
    testDataset["predicted_class"] = pd.Series(y_pred.reshape(-1,))
    testDataset.to_csv('test_results.csv',index=False,encoding="utf-8")

evaluate_model(testDataset=test_data)

//
Test accuracy of model is 0.9019607901573181
Test loss of model is 0.31719857454299927

```

Cümleleri sınıflandırmak için eğitilmiş modeli ve kelimeleri sayılara dönüştüren Tokenizer nesnesi kaydettik.

Daha sonra eğitilmiş modeli gerçek cümleleri tahmin etmekte kullanmak için bir fonksiyon yazdık.

Bu fonksiyon parametre olarak aldığı cümleyi veri seti hazırlanırken olduğu gibi gereksiz karakterlerden temizledi ve harfleri küçük harf haline getirdi.

Temizlediği cümleyi, eğitimde kullanılan veri setindeki kelimelerin sayısal karşılıklarını kullanarak sayısal hale dönüştürdü. Sayısal hale dönüştürülen cümle modelde sınıflandırıldı ve sonuçlar ekrana yazdırıldı.

Uygulamanın kodlarını tamamladıktan sonra daha rahat kullanmak için bir arayüz tasarladık. Arayüzde sorular tek tek gelecek, sol tarafta soru görülüp cevap verildikten sonra öğrenci Cevabı Gönder butonuna tıklayacak ve arayüzün sağında cevabının doğruluğunu görebilecektir.

Açık Uçlu Soruların Cevaplarını Yapay Zeka İle Değerlendirme
Araştırma Projesi Prototipidir.

ÇEVİRİMİÇİ TÜRKİYE GENELİ COĞRAFYA SINAVI

Soru 1:
Dünya 4 ocakta güneşe en yakın olduğu pozisyonudur.
Buna rağmen bu dönemin kuzey yarım kürede
hava sıcaklarının en düşük olduğu
dönem olmasının sebebi nedir?

Puanın

Bu cevap 100.0% doğru

Doğruluk yüzdesi, cevabın doğru olma olasılığını gösterir
Bu yüzden %50'den büyük sonuçlar doğru kabul edilecektir.

Cevabı Gönder

Aşağıda modelde puanlanan bazı cevapları ve doğruluk yüzdelerini göreceksiniz:

▶ `test_model("Dünyanın ekseninin düz olmaması")`

Bu cevap %96.64054107666016 oranında doğru

▶ `test_model("Kuzey yarım küreye ışınların eğri gelmesi")`

Bu cevap %82.4139175415039 oranında doğru

▶ `test_model("0 tarihlerde dünya güneşe uzak olduğu için")`

Bu cevap %0.3335237503051758 oranında doğru

▶ `test_model("İklim sebebiyle havalar soğuktur")`

Bu cevap %5.9281439781188965 oranında doğru

Not: Sayfa sınırlamasından dolayı kodlarımız ve test verilerimizin bir kısmı paylaşamamıştır.

5. Yenilikçi (İnovatif) Yönü

Ülkemizde 2018 yılından bu yana ehliyet sınavları, 2020 yılından bu yana açık öğretim lise sınavları ve pandemi sebebiyle üniversitelerin sınavları çevrimiçi olarak yapılmaktadır. Fakat sınavlarda çoktan seçmeli sorular kullanılmaktadır. Bizim uygulamamız çevrimiçi sınavlarda açık uçlu soruların kullanımına imkan sağlamaktadır.

Yapay zeka ve kelime vektörleri bir çok yeni uygulamada kullanıldığını görmekteyiz. Fakat eğitimde ölçme değerlendirme alanında kelime vektörlerinin kullanımına dair bir örnekle karşılaşmadık.

6. Uygulanabilirlik

Projemiz ölçmede yapay zeka kullanımını test edecek bir prototiptir. Prototip soruya verilen cevapları eş zamanlı olarak prototip uygulamamız ve gerçek öğretmenler puanladılar. Aynı cevaba verilen puanları kıyasladığımızda puanların uyumlu olduğunu gördük. Yani yapay zeka uygulamamızın gerçek hayatta kullanılabilmesi kanıtlanmış oldu. Uygulama, açık lise ve motorlu taşıtlar sınavları gibi var olan e-sınav sistemlerine entegre edilip test sorularının yanında açık uçlu sorular da ölçme değerlendirmede kullanılabilir.

7. Tahmini Maliyet ve Proje Zaman Planlaması

Aynı zamanda Tübitak tarafından düzenlenen Liselerarası Araştırma Projeleri Yarışmasına da başvuru yaptığımızdan, projemiz tamamlanmıştır. Projemizin tamamlanması 7 ay sürmüştür. Projemiz yapay zeka, yazılım projesi olduğundan ve uygulamada bilgisayar haricinde bir donanıma ihtiyaç duymadığından maliyeti yoktur. Projemiz gerçek yaşamda uygulamaya geçerse çevrimiçi sınav sistemlerine ayrılan bütçe haricinde bütçe gerektirmemektedir.

8. Proje Fikrinin Hedef Kitlesi (Kullanıcılar):

Projemizin hedef kitlesi örgün ve yaygın eğitime dahil olan bütün öğrenci, öğretmen ve idarecilerdir. Projemizde öğrencileri eğitim sistemimizdeki ezber döngüsünden çıkarmak istedik. Sınavların çoktan seçmeli testlerden oluşması öğretmenleri ezberletmeye, öğrencileri ezberlemeye, idareci ve diğer personeli ise yapılan ezberin başarılı olmasına olanak sağlayan düzeni dizayn etmelerine sebep olmaktadır. Sınavlarda kullanılan ölçme araçları çeşitlendirilirse problem çözme, analiz-sentez yapma gibi üst düzey bilişsel becerilen kazandırılması eğitim sürecinin öncelikleri arasında yer alacaktır.

9. Riskler

Projemizin uygulanmasında ve var olan çevrimiçi sistemlere entegre edilmesinde sorun yaşanacağını düşünmemekteyiz. Fakat soruların cevaplarını puanlayabilmek için öncelikle her bir soru için doğru ve yanlış cevaplardan oluşacak bir veri seti oluşturmak gerekmektedir. Biz projemizi yaparken bu verileri çevremizdeki insanlardan anket yoluyla topladık. Türkiye geneli bir sınavda kullanılacak açık uçlu soruların cevaplarını veri olarak toplamak sınav gizliliği açısından riskli olacaktır. Bu durum projemizin uygulamaya geçmesi açısından risk teşkil etmektedir. Fakat görüyoruz ki insanlar

ödev dahil her türlü sorularını arama motorlarına sormaktadır. İnternette çok fazla sayfada her türlü sorunun çeşitli cevaplarına ulaşılmaktadır. Hatta forum sitelerinde bir çok insan kişisel görüşlerini yazmaktadır. Biz de B planı olarak bu cevapların veri seti oluşturulurken kullanılabilceğini düşünüyöruz.

Karşılaşılabilecek bir diđer problem modelin cevabı yanlış puanlamasıdır. Bu sorun da veri seti büyütölerek çözülebilir

10. Kaynaklar

- Gronlund, N.E. (1998). Assessment of student achievement. Boston: Allyn and Bacon.
Tan, Ş., ve Erdoğan, A. (2004). Öğretimi planlama ve değerlendirme. Ankara: PegemA Yayıncılık

