

Yapay Zeka Destekli Araç Sorun Tahmin Sistemi

Doğal Dil İşleme Dersi

YUSUF YİĞİT BOZGEYİK - 214603013

Projenin Amacı Ve Temel Fikir

- Kullanıcılar araçlarıyla ilgili yaşadıkları problemleri metin olarak ifade eder.
- Eğitilmiş BERT modeli , bu metni analiz ederek sorunun hangi parçaya ait olduğunu tahmin eder.
- Amaç : Servis süreçlerini hızlandırmak , sorun tespitini otomatikleştirmek.

Hangi Probleme Çözüm Getirildi ?

- Araç kullanıcılarının ilettiği sorunlar genellikle metin formatındadır ve manuel olarak sınıflandırılması zaman alıcıdır.
- Bu proje, Türkçe metinlerden araç arıza türlerini (motor, fren, elektrik vb.) otomatik tahmin ederek bu problemi çözer

Veri Ve Önışleme

- Veri Kaynağı: arac_sorunlari_veriseti.csv adlı CSV dosyası (yapay veri seti)
- Veri seti metin ve etiket (motor , şanzıman vb.) sütunlarından oluşur.
- veri dengesizliğı kontrolu ve örnek sayısının eşitlenmesi
- temizlik : eksik/hatalı verilerin ayıklanması
- tokenization : max 128 token ile BERT ile uyumlu hale getirilfi
- Dataset.from_pandas ile Hugging Face uyumlu veri formatı hazırlanmıştır.
- Etiketler label sütununda sayısal olarak modellenmiştir.
- Tokenizer çıktıları PyTorch tensör formatına dönüştürülmüştür.

Model Ve Yöntemler

- Kullanılan model: BERTurk
- Görev türü: Tek etiketli çok sınıflı metin sınıflandırma
- Tokenizer: BertTokenizerFast
- Model Eğitimi :
- Eğitim için Trainer sınıfı kullanıldı.
- Her 100 adımda model değerlendirilerek en iyi versiyonu kaydedildi.

Araç Sorunları Sınıflandırma

Araç sorununuzu detaylı bir şekilde açıklayın.

aracımın vites geçiş



MOVAVI
SCREEN
RECORDER

Tahmin Et

Sonuçlar Ve Değerlendirme

- Geliştirilen model, araç kullanıcı şikayetlerini anlamlandırarak doğru arıza türünü tahmin etme konusunda genel olarak başarılı performans sergilemiştir.
-
- Özellikle sık karşılaşılan kategorilerde (motor, fren gibi) yüksek doğruluk elde edilmiştir.
-
- Veri dengesizliği nedeniyle az sayıda örnek içeren sınıflarda modelin tahmin gücü daha düşüktür.
-
- Bazı arıza türleri semantik olarak birbirine benzediği için sınıflar arası karışıklık yaşanabilmektedir
- .
- Kullanıcı metinlerinin çok çeşitli ve düzensiz olması modelin anlam çıkarma sürecini zorlaştırmıştır.
-
- Performans değerlendirmelerinde doğruluk, F1 skoru gibi metriklerde tatmin edici sonuçlara ulaşılmıştır.
-
- Prototip düzeyde çalışan sistem
-
- Veri miktarı ve çeşitliliği artırıldıkça modelin başarısı daha da yükselebilir.

Sonuçlar Ve Değerlendirme

- Başlangıçta kredi skoru tahmini yapan bir model eğitecektim fakat aşırı veriden dolayı çok aksilik çıktı .
- Daha sonra konu araştırırken araç sorunları için tahmin yapan bir model eğitmek istedim. Veri seti bulamadım daha sonrasında yapay veri seti hazırlamak durumunda kaldım ve bu da zamanımı oldukça aldı.
- Model eğitim kısmında elimde bulunan bilgisayar güçsüz olduğundan sistemsel hatalar aldım bu beni çok zorladı. Çözümü ise kategorileri azaltarak buldum çünkü sonuca ilerlemek gerekiyor.
-

Gelecek Çalışmalar

- Veri seti büyütülerek daha nadir arıza türleri eklenebilir.
- Kullanıcı arayüzü (Gradio, mobil vb.) entegre edilebilir.
- Etiket sayısı artırılarak daha detaylı sınıflandırma yapılabilir.
- Alternatif modeller (e.g. Electra, RoBERTa, distilBERT) ile karşılaştırma yapılabilir.

arac_sorunlari_veriseti.csv

arac_sorunlari_model.py X

app.py

arac_sorunlari_model.py > ...

```
41 def model_egitimi():
57     df['label'] = df['etiket'].map(etiket_map)
58
59     # 2. Eğitim ve doğrulama setine ayır
60     print("\nVeri seti bölünüyor...")
61     train_df, val_df = train_test_split(df, test_size=0.2, random_state=42, stratify=df['label'])
62
63     # 3. HuggingFace Dataset formatına çevir
64     train_dataset = Dataset.from_pandas(train_df[['metin', 'label']])
65     val_dataset = Dataset.from_pandas(val_df[['metin', 'label']])
66
67     # 4. Tokenizer ve model
68     print("\nModel ve tokenizer yükleniyor...")
69     model_name = "dbmdz/bert-base-turkish-cased"
70     tokenizer = AutoTokenizer.from_pretrained(model_name)
71
72     def tokenize_function(examples):
73         return tokenizer(
74             examples["metin"],
75             padding="max_length",
76             truncation=True,
77             max_length=128
78         )
79
80     print("Metinler tokenize ediliyor...")
81     train_dataset = train_dataset.map(tokenize_function, batched=True)
82     val_dataset = val_dataset.map(tokenize_function, batched=True)
83
84     # Gereksiz sütunları kaldır
85     train_dataset = train_dataset.remove_columns(["metin"])
86     val_dataset = val_dataset.remove_columns(["metin"])
87     train_dataset.set_format("torch")
88     val_dataset.set_format("torch")
89
90     # 5. Modeli yükle
91     model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=6)
92
```



MOVAVI
SCREEN
RECORDER

VIDEOLARA ERİŞMEK İÇİN HERKESE AÇIK CANVA
BAĞLANTI LINKİ .TIKLAYINIZ