

# **TÜRKÇE DATASET VE LSTM KULLANARAK TÜRKÇE ŞARKI ÜRETME PROJESİ**

YZM308 – Üretken Yapay Zeka Dersi  
Final Projesi  
Dr. Öğr. Üyesi MURAT ŞİMŞEK

Atakan

**Hasan Fatih Öztürk**  
**220212002**

Ostim Teknik Üniversitesi  
Yapay Zeka Mühendisliği

19 Mayıs 2025

# İçindekiler

0.1	Proje Tanımı ve Amacı . . . . .	2
0.2	Mevcut Durum Analizi . . . . .	3
0.3	İş Gereksinimleri . . . . .	3
0.4	Teknik Gereksinimler . . . . .	4
0.5	Veri Analizi ve Hazırlığı . . . . .	4
0.6	Model Seçimi ve Geliştirme . . . . .	5
0.7	Risk Analizi ve Kısıtlar . . . . .	5
0.8	Karşılaşılan Zorluklar ve Çözümler . . . . .	6
0.9	Modelin Mimari Tasarımı . . . . .	6
0.10	Model Eğitimi . . . . .	6
0.11	Modelin Performansı . . . . .	7
0.12	Görselleştirme ve Analiz . . . . .	7
0.13	Sonuç ve Gelecek Çalışmalar . . . . .	8
0.14	Kaynakça . . . . .	8

## 0.1 Proje Tanımı ve Amacı

**Proje Adı:** Türkçe Şarkı Sözü Üretimi (Generative AI Song Generator – TRLyrics)

### Kapsam & Temel Hedefler

- **Otomatik Söz Üretimi:** Transformer tabanlı model, hece ölçüsünü ve kafiye dikkate alarak özgün sözler yazar.
- **Seed/Temaya Uyum:** Kullanıcı “bir tohum dize” veya anahtar kelime girdiğinde, tutarlı devam kıtaları ve nakaratlar sunar.
- **Gerçek Zamanlı Erişim:** Web arayüzü ya da REST API ile  $\leq 2$  sn’de sonuç döndürür.
- **Parametre Kontrolü:** temperature, top-p, maksimum uzunluk gibi ayarlar, duygusal tonu kullanıcıya bırakır.

### Çözölmek İstenen Problemler

- Yaratıcı Tıkanma: Profesyonel söz yazarına ulaşamayan müzisyenler için hızlı ilham.
- Dil Desteęi Eksikliği: İngilizce odaklı araçların Türkçe kafiyede başarısız kalması.
- Zaman & Maliyet: Demo hazırlık süresini günlerden dakikalara indirme.

### Neden Generative AI?

- Özgünlük: Büyük veriyle eğitilen model, kopya riskini minimize ederek taze dizeler üretir.
- Hızlı Varyasyon: Tek tıklamayla onlarca alternatif; sanatçı beğendiğini seçip şekillendirir.
- Esneklik: Parametrelerle slow-rock’tan rap’e, hüzünden neşeye farklı üslûplar yakalanır.

### Uygulama Senaryoları

- Bağımsız Sanatçılar: Kısa sürede demo hazırlama; tek dizeyi girip tüm şarkıya ulaşma.
- Reklam Ajansları: Marka jingle’ı için 10-15 saniyelik özgün söz sloganları.
- Oyun & Film Sektörü: Tema müzikleri için konseptte uyumlu dinamik söz üretimi.
- Eğitim: Yaratıcı yazma atölyelerinde öğrencilere etkileşimli örnekler.

## 0.2 Mevcut Durum Analizi

### Rakip Analizi

- **Suno AI:** Söz+beste birlikte üretir; Türkçe desteği beta aşamasında.
- **OpenAI MuseNet:** Çokdilli enstrümantasyon; söz üretmez.
- **Yerel Uygulamalar:** Az sayıda akademik prototip; piyasada ticari araç yok.

### Hedef Kullanıcı Kitlesi ve İhtiyaçları

- Bağımsız müzisyenler & söz yazarları → ilham, taslak, hız.
- Reklam ajansları → jingle sözleri.
- Oyun & medya stüdyoları → orijinal Türkçe içerik.

## 0.3 İş Gereksinimleri

### Fonksiyonel Gereksinimler

- Kullanıcı prompt'u (tema / ilk dize / duygu) girer, sistem  $\leq 10$  sn içinde  $\geq 4$  kıta üretir.
- Parametre ayarı: temperature, top-p, uzunluk (token) seçilebilir.
- “Satır satır”, “virgül ile ayır” veya “otomatik kafiye” biçim seçenekleri.
- Sonuçları .txt/.docx indirme ve paylaşılabılır görsel (söz + arka plan) üretme.

### Fonksiyonel Olmayan Gereksinimler

- API geçidi: REST + async destek.
- Tepki süresi  $< 1000$  ms (cache'lenmiş model bellekte).
- Kullanıcı verisi 1 yıl saklanır; GDPR & KVKK uyumlu.
- Erişilebilirlik: mobil ve masaüstü uyumlu.

### Kullanıcı Senaryosu (Örnek)

#### User Story – Müzisyen:

“Başlamak istediği sözü girer ve şarkılar akmaya başlar.” Sentiment analysis bir sonraki sürümde gelecektir.

## 0.4 Teknik Gereksinimler

Bileşen	Seçim	Gerekeçe
Veri Tipi	Saf metin (Türkçe şarkı sözleri)	Dil modeli için yeterli
Model	LLM tabanlı	Çeşitlilik + kalite
Veri Kaynağı	Public domain / Scraping / Stüdyo arşivleri	Çeşitlilik + kalite
Veri Miktarı	$\geq 200$ bin satır, $\approx 20$ MB corpus	Dil modeli için yeterli
Tokenizasyon	SentencePiece BPE, vocab 16 000	OOV azaltır
Model Tipi	LSTM (1 katman), Embedding 128, LSTM 1024	Düşük GPU gereksinimi
Altyapı	NVIDIA A100 40 GB GPU; TensorFlow 2.x	Ölçeklenebilirlik

## 0.5 Veri Analizi ve Hazırlığı

- **Toplama:** Web scraper  $\rightarrow$  .csv; lisans kontrolü
- **Temizleme:** Bozuk karakterler temizlendi, csv geçerken utf-8 ile çözümlü
- **Kalite Kontrol:** Dilbilgisi filtreleri, manuel rastgele örnek doğrulaması
- **Gizlilik & Etik:** Telif sahipliği; kullanıcı prompt'u loglarının anonimleştirilmesi.

Ahmet Kay (Potburi) Fabrika KÄ±zÄ±	Äafak TÄ½rkÄ½sÄ½	Bir mavi otobÄ½s
Ahmet Kay AcÄ± Ninni	BaÄ½ KaldÄ±yorum	Uyusun ha iyi
Ahmet Kay AcÄ±lara Tutunmak	AcÄ±lara Tutunmak	KavuÄ½mak
Ahmet Kay Ada Sahilleri	HoÄ½ÅşakalÄ±n GÄ½zÄ½m	Ada sahillerinde
Ahmet Kay AdÄ± Bahtiyar	Dinle Sevgili Äcelkem	GeÄ½iyor
Ahmet Kay AdÄ± YÄ±lmaz	Dosta DÄ½Ä½mana KarÄ½Ä±	Dalyan gibi bir
Ahmet Kay Ah	Resitaller	YÄ½zÄ½nÄ½n
Ahmet Kay Ah Ulan RÄ±za	Resitaller	Neden hala gelmedi?
Ahmet Kay Aklanacak DÄ½nya	YazmalÄ± Gelin	AlnÄ±nÄ±n orta
Ahmet Kay Al Ä½fkemi	HoÄ½ÅşakalÄ±n GÄ½zÄ½m	Her sÄ½rgÄ½nÄ½n
Ahmet Kay AlnÄ±nda DaÄ½ AteÄ½i	Yorgun Demokrat	AlnÄ±nÄ± daÄ½
Ahmet Kay AmanÄ± MinnoÄ½	Ä½yimser Bir GÄ½l	DaÄ½da keklik
Ahmet Kay Amenna (YaÄ½ayanlar B	AcÄ±lara Tutunmak	YaÄ½ayanlar bir gÄ½r
Ahmet Kay An Gelir	An Gelir	PalDÄ±r kÄ½ldÄ½r

Şekil 1: Tablo: Temizlemeden önce veriler.

## 0.6 Model Seçimi ve Geliştirme

- **LSTM (Baseline):** Hızlı eğitim, düşük parametre; Uzun-bağlam sınırlı
- **GRU:** Daha az parametre; aynı sınırlama
- **Transformer (Decoder-only):** Daha uzun bağlam, kaliteli üretim; GPU maliyeti >2x
- **Fine-tuned GPT-2-Türkçe:** SOTA metin kalitesi; daha büyük veri ve lisans maliyeti (Kullanılmadı)

Seçim: Önce tek LSTM, sonra iki LSTM arasında dropout ve sequence düzeni.

### Eğitim & Test Stratejisi

- 80-10-10 (train/val/test) bölme
- Epoch: 20; Batch: 128
- Erken durdurma (patience 5)

### Başarı Kriterleri

- Perpleksite (PPL) < 35
- İnsan Turing testi:  $\geq 60\%$  “insan yazdı”
- BLEU-4 > 0.25 (prompt  $\rightarrow$  söz benzerliği)
- Kullanıcı memnuniyeti anketi  $\geq 4/5$

## 0.7 Risk Analizi ve Kısıtlar

Kategori	Risk	Azaltım
Teknik	Overfitting küçük korpus	Veri augmentasyonu & dropout
Teknik	Uzun çıkışlarda tekrara girme	Top-p + n-gram blokaj
Yasal	Telif haklı sözlerin ezberlenmesi	Kayıtları filtrele
Etik	Nefret söylemi üretimi	Content filter
Operasyonel	GPU tedarigi	Bulut multi-region alternatifleri

## 0.8 Karşılaşılan Zorluklar ve Çözümler

- Veri Seti Temizliği: Türkçe karakterlerde bozulmalar, eksik veya anlamsız satırlar
- Donanım Kısıtlamaları: Eğitim süresinin uzunluğu, yüksek bellek ihtiyacı
- Overfitting: Dropout ve erken durdurma (EarlyStopping) ile aşıldı

## 0.9 Modelin Mimari Tasarımı

Model, Sequential bir Keras modeli olarak tasarlanmıştır:

- **Embedding Layer:** Tokenleri vektöre dönüştürür.
- **LSTM Layer:** Sıralı verileri işler, dil modeli kurar.
- **Dropout:** Aşırı öğrenmeyi engeller.
- **Dense Layer:** Çıktı vektörünü oluşturur.

```
####Tek lstm kullanmak için bu kodu kullanın
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, EMBED_DIM, input_length=SEQ_LEN),
    tf.keras.layers.LSTM(LSTM_UNITS),
    tf.keras.layers.Dense(vocab_size, activation="softmax")
])

optimizer = Adam(learning_rate=0.001)
model.compile(
    optimizer=optimizer,
    loss="sparse_categorical_crossentropy"
)
```

Şekil 2: Tablo: model kodları.

Modelin eğitiminde EarlyStopping ve ModelCheckpoint gibi geri çağırma fonksiyonları kullanılmıştır.

## 0.10 Model Eğitimi

Model, 20 epoch boyunca, batch\_size=128 olacak şekilde eğitilmiştir. Eğitim sırasında loss (kayıp) değerleri takip edilmiştir. Modelin eğitim süresi, kullanılan donanıma (A100 GPU) göre yaklaşık 20 dakika olarak belirtilmiştir.

```

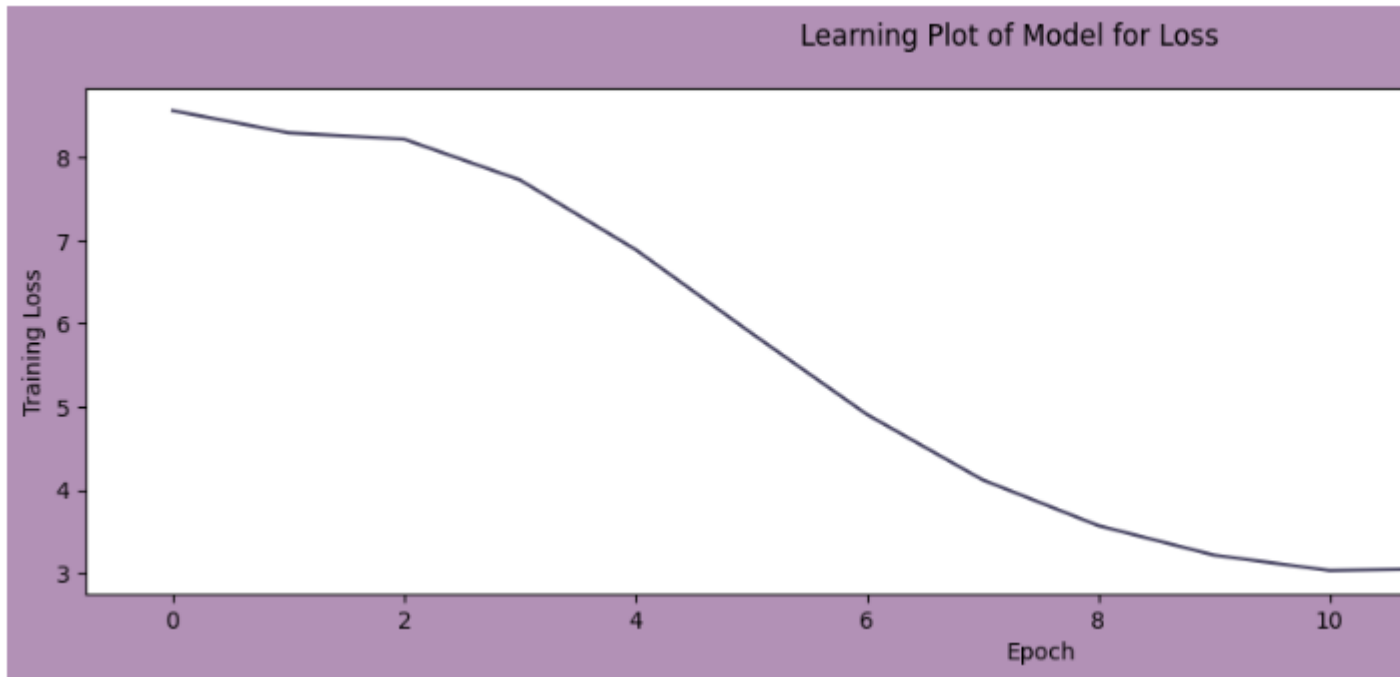
checkpoint = ModelCheckpoint(
    '/content/drive/MyDrive/MRT_SMSK/check4_model.keras', ##HER KULLANIMI
    monitor='loss',
    save_best_only=True
)#Loss yükselmeye başlarsa son epochu kaydedecek
#Loss grafiği çıkarmak için history.df bağlayacağız
history = model.fit(
    dataset,
    epochs=EPOCHS,
    callbacks=[es, checkpoint]
)

```

Şekil 3: Tablo: Model.

## 0.11 Modelin Performansı

Model eğitimi tamamlandığında loss değeri gözlemlenmiş, loss değeri 4 altına düşmemiştir. Modelin eğitim sonrası test edilmesi için çeşitli başlangıç dizeleriyle şarkı sözü üretimi denenmiştir. Üretilen şarkı sözleri, anlam bütünlüğü, tutarlılık ve yaratıcılık açısından manuel olarak incelenmiştir.



Şekil 4: Tablo: Kayıp fonksiyon verileri LATEX sebebiyle kaymıştır.

## 0.12 Görselleştirme ve Analiz

- Eğitim ve doğrulama loss grafiklerinin çizimi



- Modelin zaman içerisindeki başarımı, olası overfitting/underfitting durumları
- Üretilen şarkı sözlerinin örnekleri:

## 0.13 Sonuç ve Gelecek Çalışmalar

Bu proje, Türkçe şarkı sözü üretiminde LSTM tabanlı yaklaşımların başarılı olabileceğini göstermiştir. Özellikle yaratıcı metin üretimi için, daha büyük ve dengeli veri setleriyle, farklı mimariler (Transformer tabanlı modeller, GPT, vb.) denenerek başarı daha da artırılabilir. Ayrıca, kullanıcıdan belirli bir tema, ruh hali veya anahtar kelimeyle şarkı üretimi için ilave arayüzler geliştirilebilir.

## 0.14 Kaynakça

- Chollet, F. diğerleri. (2015). Keras [Yazılım]. GitHub. Retrieved May 20, 2025, from <https://keras.io/api/>
- Keras 3 API Documentation. (2025). Models API, Layers API, Callbacks API ve diğer bileşenler [Çevrimiçi belge]. Retrieved May 20, 2025, from <https://keras.io/api/>
- Kudo, T. Richardson, J. (2018). google/SentencePiece [Yazılım]. GitHub. Retrieved May 20, 2025, from <https://github.com/google/sentencepiece>
- The pandas development team. (2020). pandas-dev/pandas: Pandas [Yazılım]. Zenodo. Version: latest. DOI: 10.5281/zenodo.3509134. Erişim: <https://doi.org/10.5281/zenodo.3509134>
- Hochreiter, S. Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- Gers, F. A., Schmidhuber, J. Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10), 2451–2471.
- Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N. Wu, Y. (2016). Exploring the Limits of Language Modeling. arXiv:1602.02410. Retrieved May 20, 2025, from