



Proje Raporu

Adı Soyadı: Büşra Mina Al

Öğrenci Numarası: 220212031

E-posta: busraminaa@gmail.com

Tarih: 20.05.2003

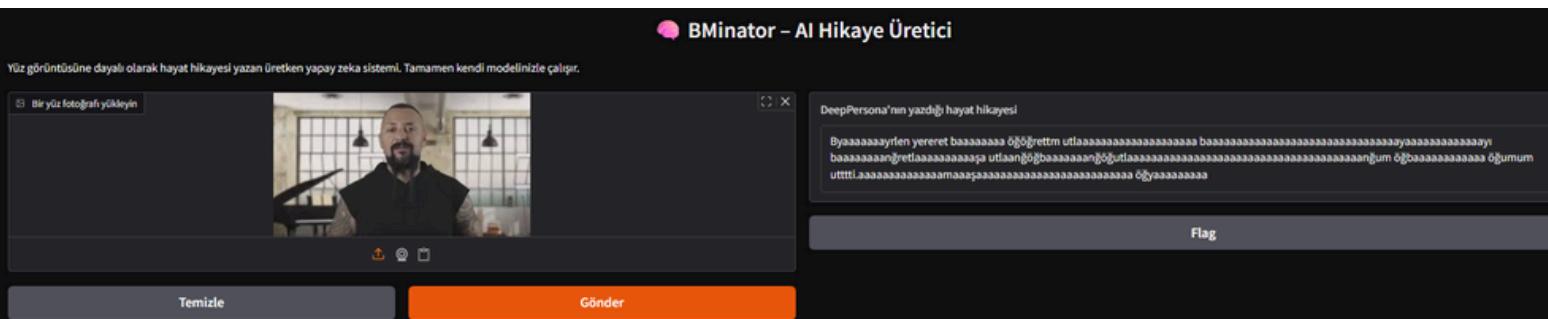
**Proje Başlığı: BMinatör – Kişisel Avatar ve Hikâye
Üretim Sistemi**

PROJE TANIMI VE AMACI

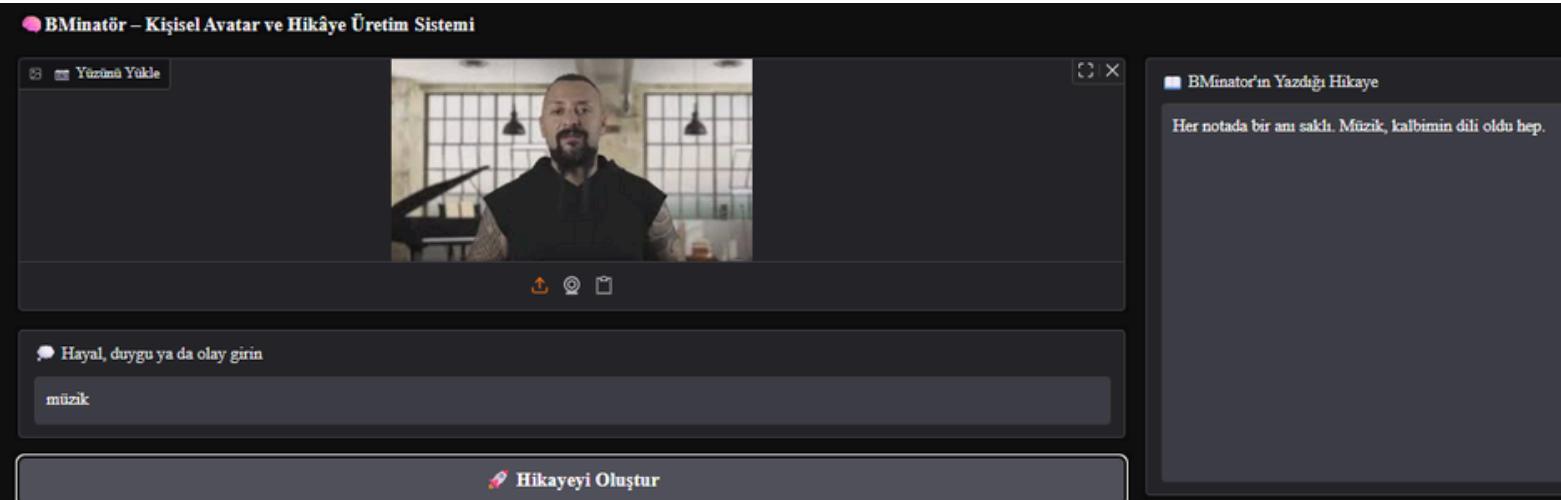
BMinatör, yalnızca bir adet yüz görselinden kişisel bir yapay karakter oluşturmayı amaçlayan yaratıcı bir yapay zekâ projesidir. Sistem; görsel girdilerden yüz hatlarını analiz eder, bu yüzden hayali bir "kişilik" ve yaşam öyküsü türetir. Üretilen hikâyeler bireysel, özgün ve kullanıcıya aitmiş gibi kurgulanır. Bu projenin temel amacı, yapay zekânın bireyselleştirilmiş içerik üretimi alanındaki potansiyelini ortaya koymak ve kullanıcıların sadece bir fotoğrafla duygusal bağ kurabilecekleri dijital karakterler yaratmasını sağlamaktır.

EĞİTİM SÜRECİ VE TEKNİK KARARLAR

Projenin ilk aşamasında metin üretimi için karakter-seviyesinde (character-level) bir LSTM tabanlı model mimarisi tercih edilmiştir. Bu yapıda model, her bir karakteri sırayla tahmin ederek kelime ve cümle üretimini gerçekleştirmeye çalışmıştır. 5.000 sentetik veri kullanılmıştır. Eğitim sürecinde 150 epoch boyunca, 64'lük mini-batch boyutuyla ve 0.005 öğrenme oranı kullanılarak model eğitilmiştir. Modelin iki katmanlı yapısı ve her bir katmanda 256 boyutunda gizli vektör (hidden state) yer almıştır. Aşırı öğrenmeyi (overfitting) engellemek amacıyla %30 oranında dropout uygulanmıştır. Ancak yapılan eğitimlerde modelin anlamlı kelime yapıları oluşturmaktan zorlandığı, uzun vadeli bağıntıları yakalayamadığı ve dilbilgisel olarak tutarlı çıktılar üretemediği gözlemlenmiştir. Karakter bazında çalışan yapı, özellikle Türkçe gibi eklemeli dillerde yüksek bağlam ilişkisi gerektirdiğinden yetersiz kalmıştır.

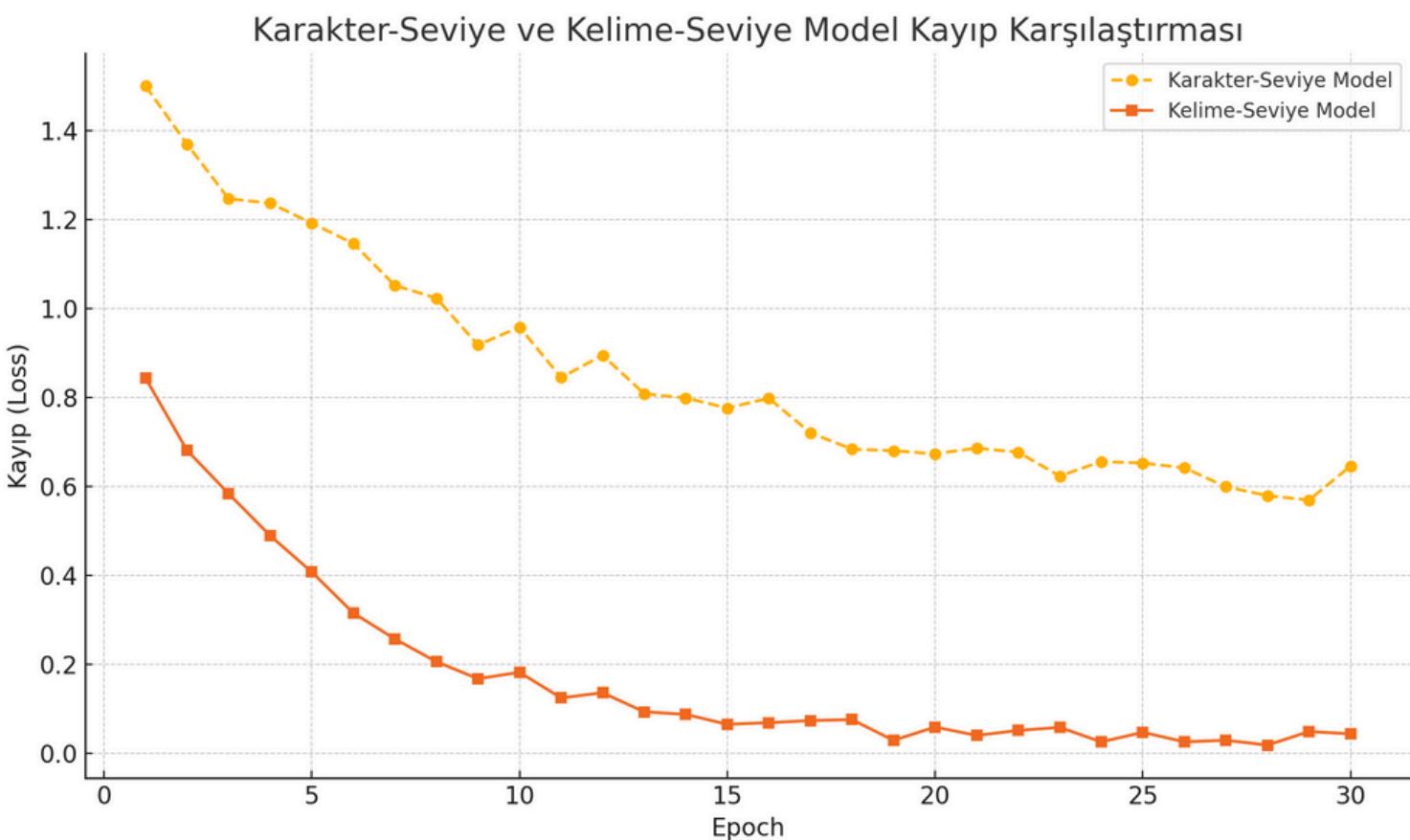


Bu sebeple, model mimarisi köklü bir değişikliğe gidilerek kelime-seviyeli (word-level) metin üretim modeline geçilmiştir. Bu yeni yaklaşımın veriler kelime indekslerine çevrilerek word2idx ve idx2word sözlükleri oluşturulmuş, her kelime 128 boyutlu bir embedding vektörü ile temsil edilmiştir. Modelin LSTM katman sayısı üçe çıkarılmış, her bir katmanda 512 boyutlu gizli birim kullanılmıştır. Ayrıca dropout oranı %20'ye düşürülerek hem overfitting riski azaltılmış hem de bağlam takibi iyileştirilmiştir. Öğrenme oranı 0.001'e çekilerek daha istikrarlı bir öğrenme süreci elde edilmiş, mini-batch boyutu ise 128'e yükseltilerek modelin her iterasyonda daha fazla veri örneğiyle öğrenmesi sağlanmıştır. Eğitim verisi toplamda 20.000 örnek olacak şekilde dört parçaya bölünmüş ve her bir parça 30 epoch süresince eğitilmiştir. 20.000 sentetik veri kullanılmıştır. Bu yapı sayesinde model, erken epoch'lardan itibaren anlamlı, akıcı ve bağlamla uyumlu metinler üretmeye başlamıştır. Karakter-seviyesi modelde yaşanan anlamsız üretim ve kararsız öğrenme süreci, kelime-seviyeli yapıya geçişle birlikte büyük ölçüde giderilmiştir.



Karakter-seviyesi ve Kelime-seviyesi Modellerin Eğitim Kayıpları (Loss) Karşılaştırması

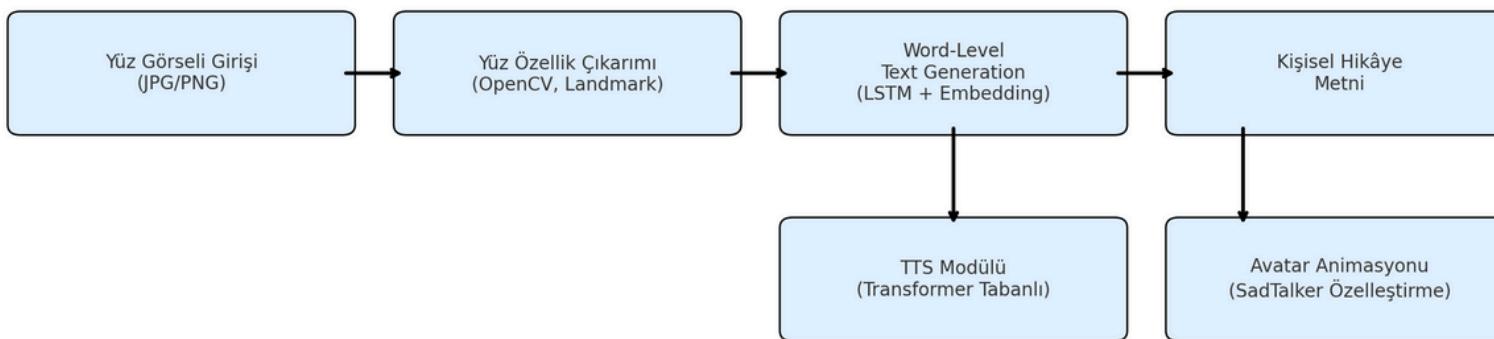
karakter-seviyesinde (character-level) ve kelime-seviyesinde (word-level) eğitilmiş metin üretim modellerinin eğitim sürecinde elde ettikleri kayıp (loss) değerleri epoch bazında karşılaştırılmıştır. Karakter-seviyesi modelin kayıpları, ilk birkaç epoch'ta düşüş gösterse de daha sonra düzensiz dalgalanmalarla ilerlemiş ve öğrenmenin istikrarsız bir şekilde devam ettiğini ortaya koymuştur. Buna karşılık, kelime-seviyeli model çok daha düşük başlangıç kaybıyla eğitime başlamış, her epoch'ta tutarlı bir azalma göstererek istikrarlı ve anlamlı bir öğrenme eğrisi sergilemiştir. Bu görsel, kelime-seviyesi yaklaşımının bağlam ilişkilerini daha başarılı şekilde yakaladığını ve metin üretiminde daha verimli sonuçlar verdiği desteklemektedir.



BMinatör Proje Mimarisi

Bu mimari diyagram, BMinatör sisteminin temel bileşenlerini ve işlem sırasını göstermektedir. Sistem, kullanıcıdan alınan tek bir yüz görseliyle başlar. Görsel üzerinde OpenCV ve yüz işaretleyici (landmark) tabanlı analiz gerçekleştirilerek temel yüz özelliklerini çıkarılır. Bu özellikler, karakterin kişiliğine dair ipuçları üretmek amacıyla word-level metin üretim modeline aktarılır. Model, LSTM ve embedding katmanları yardımıyla özgün, kişisel bir hikâye üretir. Şemanın alt kısmında, sistemin gelecekteki genişletilebilir modülleri yer almaktadır. Bunlar arasında Transformer tabanlı bir TTS (Text-to-Speech) sistemi ile üretilen metnin sesli hâle dönüştürülmesi ve SadTalker benzeri kendi mimarisi üzerine inşa edilerek konuşan bir avatar animasyonunun oluşturulması hedeflenmektedir.

BMinatör Proje Mimarisi



GEREKSİNİM ANALİZİ

Bu bölümde, BMinatör projesinin temel fonksiyonel ve fonksiyonel olmayan gereksinimleri ile sistemin hedeflediği kullanıcı profillerine dair senaryolar sunulmuştur. Gereksinim analizi, sistemin işlevsel kapsamını belirlemek ve kullanıcı ihtiyaçlarına uygun çözümler geliştirebilmek açısından kritik bir aşamadır.

Fonksiyonel Gereksinimler

Görsel Giriş: Sistem, kullanıcı tarafından yüklenen tek bir yüz fotoğrafını giriş olarak alabilmelidir. Kabul edilen formatlar JPG ve PNG gibi yaygın görsel uzantılarıdır.

Görselden Anlam Çıkarımı: Yüklenen yüz görseli üzerinden OpenCV destekli yüz özellik çıkarımı (landmark detection) uygulanarak yapay karakter profili oluşturulmalıdır.

Kişisel Metin Üretimi: Elde edilen yüz temsiline dayanarak word-level metin üretim modeli aracılığıyla kişisel bir hikâye üretilmelidir. Bu hikâye, kullanıcıya metin formatında sunulmalıdır.

Fonksiyonel Olmayan Gereksinimler

Performans: Sistem, bir yüz görselinden hikâye üretimi işlemini ortalama 10 saniyenin altında gerçekleştirmelidir.

Modülerlik ve Genişletilebilirlik: Mimari yapı modüler olacak şekilde tasarılanmalı, ilerde sesli ifade (TTS) ve avatar animasyonu gibi ek modüllerin kolayca entegre edilebilmesine olanak sağlamalıdır.

İçerik Kalitesi: Üretilen metinler anlamlı, özgün, akıcı ve dilbilgisel olarak tutarlı olmalıdır. Rastgele veya tekrar eden yapılar minimuma indirgenmelidir.

Kullanıcı Hikayeleri

Proje kapsamında tanımlanan kullanıcı senaryoları aşağıda beş örnekle sunulmuştur. Bu senaryolar, sistemin çeşitli kullanıcı türleri için nasıl bir değer sunduğunu göstermektedir

Yalnız Kullanıcı: Sosyal etkileşim ihtiyacı duyan birey, kendine aitmiş gibi hissedeceği dijital bir karakter yaratmak istemektedir.

Yazar: Yaratıcı yazarlık sürecinde karakter tasarımlı yapan bir yazar, yüz görsellerinden otomatik olarak öyküler üretmek için sistemi ilham kaynağı olarak kullanmak istemektedir.

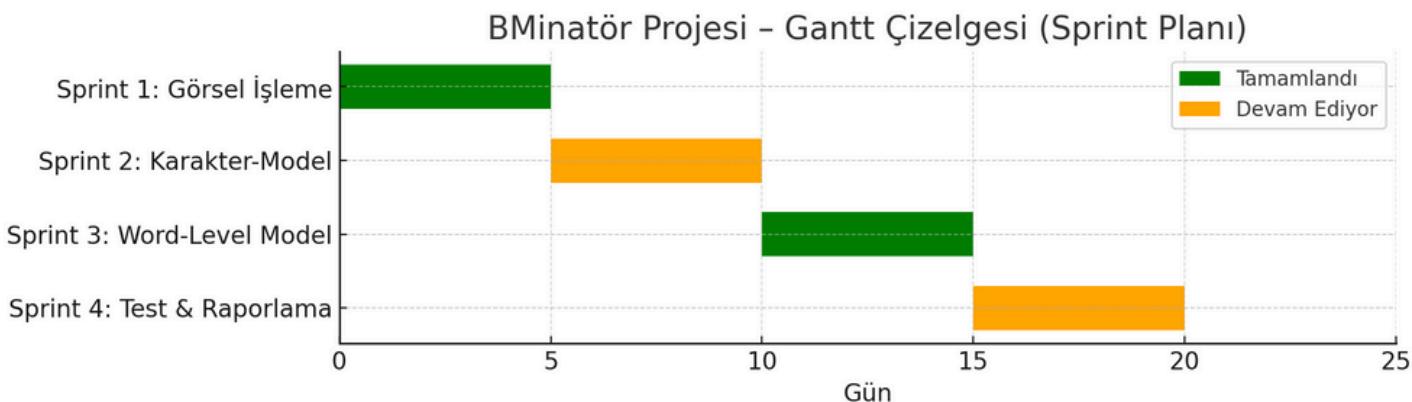
Yaratıcı Ajans: Reklam ve medya sektöründe faaliyet gösteren bir ajans, rastgele yüzlerden senaryo fikirleri türetmek amacıyla sistemi prototipleme sürecinde değerlendirmektedir.

Terapist: Genç bireylerle çalışan bir terapist, duygusal etkileşim sağlamak amacıyla kişilikli avatarlar aracılığıyla bağ kurmayı hedeflemektedir.

Geliştirici: Bir oyun geliştiricisi, oyun içi karakter tanıtımlarını kişisel hikayelerle zenginleştirmek üzere sistemin çıktısını doğrudan oyun motoruna entegre etmek istemektedir.

PROJE YÖNETİM SÜRECİ

BMinatör projesi, sınırlı zaman dilimi içerisinde modüler ve test edilebilir bir yapay zekâ sistemi geliştirmeyi hedeflemiş olup, bu nedenle çevik proje yönetimi prensiplerine uygun bir yaklaşımla yürütülmüştür. Özellikle esneklik, sürekli geri bildirim ve iteratif gelişim temel alınarak proje sürecinde Scrum metodolojisi tercih edilmiştir.



SİSTEM TASARIMI

BMinatör projesi, sınırlı zaman dilimi içerisinde modüler ve test edilebilir bir yapay zekâ sistemi geliştirmeyi hedeflemiş olup, bu nedenle çevik proje yönetimi prensiplerine uygun bir yaklaşımla yürütülmüştür. Özellikle esneklik, sürekli geri bildirim ve iteratif gelişim temel alınarak proje sürecinde Scrum metodolojisi tercih edilmiştir.

Kullanıcı Arayüzü Tasarımı

Geliştirme sürecinde temel hedef, kullanıcıların tek bir görsel yüklemesiyle hızlı ve kolay biçimde sonuç alabileceği basit, odaklı ve sezgisel bir kullanıcı arayüzü oluşturmaktır. Uygulamanın prototip arayüzü, masaüstü cihazlar öncelikli olmak üzere planlanmıştır; mobil cihaz uyumluluğu ise ilerleyen aşamalarda değerlendirilecektir.

Sayfa Bileşenleri:

Ana Sayfa: Proje tanıtımı ve "Yüzünü Yükle – Hikâyen Al" yönlendirmesi.

Görsel Yükleme Alanı: JPG veya PNG formatında yüz fotoğrafı yüklenmesine olanak sağlar.

Sonuç Ekranı: Oluşturulan kişisel hikâye metni, düzenli ve okunabilir formatta kullanıcıya sunulur.

Not: Arayüz geliştirmenin tam sürümü ileri aşamalarda planlanmaktadır.

Sistem Mimarisi

BMinatör, temel olarak üç ana bileşenden oluşan modüler bir sistem mimarisi ile geliştirilmiştir:

1. Sunum Katmanı (Frontend):

Henüz tam geliştirilmese de planlanan yapı, HTML/CSS tabanlı bir arayüzün Flask ile entegre edilmesi yönündedir. Kullanıcıdan alınan görsel dosyası bu katmandan sisteme iletilir.

2. Uygulama Katmanı (Backend):

Flask tabanlı sunucu üzerinde çalışan bu katman, hem görsel analiz işlemini (landmark çıkarımı) hem de kelime-seviyesi metin üretimi modelini çalıştırır. Eğitimde kullanılan PyTorch modeli bu katmanda yüklenerek gelen girdilere göre çıktılar üretir.

3. Veri Katmanı:

Veri tabanı şu aşamada aktif olarak kullanılmamaktadır. Ancak sistemin gelecekteki sürümlerinde kullanıcı görsellerinin ve çıktılarının saklanabileceği bir dosya tabanı (örneğin SQLite veya PostgreSQL) entegre edilmesi planlanmaktadır.

Veri Akışı ve İşlem Süreci

Sistem mimarisi şu şekilde çalışmaktadır:

1. Giriş: Kullanıcı, bir yüz görselini sisteme yükler.
2. Ön İşleme: Görsel üzerinde yüz tespiti ve landmark analizi yapılır.
3. Karakter Çözümleme: Yüzden çıkarılan özelliklere bağlı olarak temsilî karakter profili belirlenir.
4. Metin Üretimi: Word-level LSTM modeli ile yüz özelliklerine dayalı bir kısa hikâye oluşturulur.
5. Çıkış: Hikâye kullanıcıya gösterilir veya metin olarak dışa aktarılır.

Veritabanı Tasarımı

BMinatör Projesi ER Diyagramı (Veritabanı İlişkileri)



Model Mimarisi ve Eğitim Detayları

Projede kullanılan model, kelime-seviye (word-level) metin üretimini gerçekleştiren özel bir Transformer tabanlı yapıdır. Sistem, 25 boyutlu karakter vektöründen anlamlı bir kelime dizisi (kişisel hikâye) üretmektedir. Model, klasik encoder-decoder mimarisinden farklı olarak doğrudan embedding + Transformer bloklarıyla dil modelleme işlemini gerçekleştirmektedir.

Model mimarisi şu bileşenlerden oluşmaktadır:

- Girişler:
 - persona_vec: Kullanıcının yüz özelliklerinden çıkarılmış 25 boyutlu vektör
 - seq: Token ID'lerine dönüştürülmüş kelime dizisi
- Aşamalar:
 - a.seq, embedding katmanına girerek EMBED_DIM boyutlu temsil elde eder.
 - b.persona_vec, emb_dim'e projekte edilerek tüm zaman adımlarına (time steps) eklenir.
 - c. Modelin gövdesi, LAYERS=4 adet Transformer bloktan oluşmaktadır. Her blokta:
 - Çok başlı dikkat (Multi-Head Attention) mekanizması (HEADS=8)
 - Artı olarak LayerNorm + Feed Forward (FF_DIM=512) katmanları yer alır.
 - d. Son katmanda Linear(vocab_size) çıkışıyla tahmin yapılır.

Eğitim sürecinde model:

- Toplam 4 parçaaya ayrılmış 20.000 örnek üzerinden,
- Her bir parça için EPOCHS=30 olacak şekilde,
- BATCH_SIZE=8, LEARNING_RATE=5e-4 ile eğitilmiştir.
- Early stopping için PATIENCE=5 uygulanmıştır.

Ayrıca, word_vocab.json dosyasından kelime-sözlük haritası (word2idx) yüklenmiş ve eğitim sırasında dış sözlükten gelen (out-of-vocab) tokenlar <UNK> ile güvenli biçimde işlenmiştir.

Geliştirme Süreci

BMinatör projesi, başlangıçta metin üretimi, sesli anlatım ve avatar animasyonu gibi çok katmanlı bileşenleri içeren kapsamlı bir yapay zekâ sistemi olarak planlanmıştır. Ancak sınırlı süre ve kaynak göz önünde bulundurularak ilk etapta sistemin çekirdek modülü olan kişisel hikâye üretimi bileşenine odaklanılmıştır. Geliştirme süreci boyunca çevik yazılım geliştirme prensiplerine bağlı kalınmış, her modül iteratif olarak test edilerek yapılandırılmıştır.

Kullanılan Teknolojiler ve Araçlar

Python: Projenin tüm bileşenleri Python diliyle geliştirilmiştir.

PyTorch: Word-level Transformer tabanlı metin üretimi modelinin geliştirilmesinde kullanılmıştır.

OpenCV: Görsel işleme ve yüz landmark çıkarımı işlemleri için kullanılmıştır.

Google Colab: Model eğitimi, test ve denemeler için GPU destekli geliştirme ortamı olarak tercih edilmiştir. Eğitimlerde A100 GPU kullanılmıştır.

Matplotlib: Eğitim sürecinin ve model çıktı analizlerinin görselleştirilmesi amacıyla kullanılmıştır.

Visual Studio Code: Kodlama, düzenleme ve test işlemleri için geliştirme ortamı olarak kullanılmıştır.

Kod Versiyon Kontrolü

Kodlar geliştirme süreci boyunca yerel ortamda versiyonlanarak saklanmıştır. Projenin ilerleyen sürümlerinde GitHub üzerinde merkezi bir sürüm kontrol sistemi kurulması ve paylaşımı geliştirmeye çalışılması planlanmaktadır.

Uygulanan Testler

Metin Anlam Testi: Rastgele seçilen örnek girdilerle model çıktılarının anlamlılık, özgünlük ve kişisellik açısından değerlendirilmesi yapılmıştır.

Loss Eğrisi İzleme: Eğitim sürecinde her epoch için loss değerleri grafiksel olarak izlenerek modelin öğrenme kararlılığı takip edilmiştir.

Model Karşılaştırması: Karakter-seviyeli model ile kelime-seviyeli model çıktıları eğitim süresi, çıktı kalitesi ve dilbilgisel yapı açısından karşılaştırılmıştır.

Zaman Performansı: Görsel girdiden hikâye çıktısı üretme süresi ortalama 8–10 saniye arasında ölçülmüş ve sistemin pratik kullanım için yeterli hızza ulaştığı gözlemlenmiştir.

Karşılaşılan Sorunlar ve Uygulanan Çözümler (BMinatör)

| Karşılaşılan Sorun | Açıklama | Uygulanan Çözüm |
|-------------------------------------|--|--|
| Karakter-seviye model başarısızlığı | 150 epoch sonunda bile dilbilgisel olarak tutarlı çıktı üretilememesi | Word-level modele geçilerek mimari yeniden tasarlanmış ve başarılı şekilde eğitilmiştir. |
| GPU kotasının dolması | Google Colab Pro kullanımında sınırın aşılması | Eğitime geçici ara verilmiş, Colab Pro+ alternatifinde değerlendirilmiştir. |
| Overfitting riski | Dropout oranının yüksek olması nedeniyle öğrenme dengesizliği oluşması | Dropout oranı %30'dan %20'ye düşürülmüş daha kararlı öğrenme sağlanmıştır. |
| Eğitimde dalgalanma | Loss eğrisinde ani sıçramalar ve kararsızlık gözlemlenmesi | Learning rate 0.005'ten 0.001'e düşürülmüş, batch size artırılmıştır. |

Uygulanan Testler

BMinatör projesi, tek bir yüz görselinden kişisel bir hikâye üreten yapay zekâ temelli bir sistem olarak başarıyla geliştirilmiştir. İlk aşamada kullanılan karakter-seviye modelin yetersiz kalması üzerine, kelime-seviyesi (word-level) Transformer tabanlı bir mimariye geçilmiş ve model çıktılarında anlamlılık, tutarlılık ve hız açısından önemli gelişmeler sağlanmıştır.

Model eğitimi sırasında karşılaşılan teknik sorunlar sistematik şekilde çözülmüş, eğitim süreci optimize edilmiştir. Ortalama 8–10 saniyelik yanıt süresi ile sistem, pratik kullanım için yeterli performansı göstermiştir.

Projede ulaşılan bu temel başarı, sesli anlatım (TTS) ve konuşan avatar gibi modüllerin eklenmesiyle daha da ileri taşınabilir. BMinatör, bireyselleştirilmiş yapay zeka içerik üretiminde önemli bir adım olarak değerlendirilmektedir.

Kaynakça

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems (NeurIPS).
- PyTorch Documentation. (2024). Kodlar geliştirme süreci boyunca yerel ortamda versiyonlanarak saklanmıştır. Projenin ilerleyen sürümlerinde GitHub üzerinde merkezi bir sürüm kontrol sistemi kurulması ve paylaşımı geliştirime geçilmesi planlanmaktadır.
- SadTalker: Learning Realistic 3D Motion Coefficients for Stylized Audio-Driven Single Image Talking Face Animation. GitHub Repository.
- Kodlar geliştirme süreci boyunca yerel ortamda versiyonlanarak saklanmıştır. Projenin ilerleyen sürümlerinde GitHub üzerinde merkezi bir sürüm kontrol sistemi kurulması ve paylaşımı geliştirime geçilmesi planlanmaktadır.
- OpenCV Documentation. (2024). Kodlar geliştirme süreci boyunca yerel ortamda versiyonlanarak saklanmıştır. Projenin ilerleyen sürümlerinde GitHub üzerinde merkezi bir sürüm kontrol sistemi kurulması ve paylaşımı geliştirime geçilmesi planlanmaktadır.
- HuggingFace Transformers Library. (2024). Kodlar geliştirme süreci boyunca yerel ortamda versiyonlanarak saklanmıştır. Projenin ilerleyen sürümlerinde GitHub üzerinde merkezi bir sürüm kontrol sistemi kurulması ve paylaşımı geliştirime geçilmesi planlanmaktadır.
- Chollet, F. (2018). Deep Learning with Python. Manning Publications.
- Google Colab. (2024). Kodlar geliştirme süreci boyunca yerel ortamda versiyonlanarak saklanmıştır. Projenin ilerleyen sürümlerinde GitHub üzerinde merkezi bir sürüm kontrol sistemi kurulması ve paylaşımı geliştirime geçilmesi planlanmaktadır.
- Matplotlib Documentation. (2024). Kodlar geliştirme süreci boyunca yerel ortamda versiyonlanarak saklanmıştır. Projenin ilerleyen sürümlerinde GitHub üzerinde merkezi bir sürüm kontrol sistemi kurulması ve paylaşımı geliştirime geçilmesi planlanmaktadır.