PixelBytes: Catching Unified Embedding for Multimodal Generation

Fabien Furfaro

2024

Abstract

This report presents PixelBytes, an innovative multimodal embedding model designed for the simultaneous generation of text and images pixel by pixel. We introduce the PxByEmbed algorithm as a key component of our architecture, enabling efficient representation of mixed sequences of text and images. Our approach leverages web scraping, image processing, and advanced machine learning techniques to create a unique dataset and training pipeline. We compare the performance of RNN, Transformer, and Mamba models in this task, providing insights into the effectiveness of different architectures for multimodal generation.

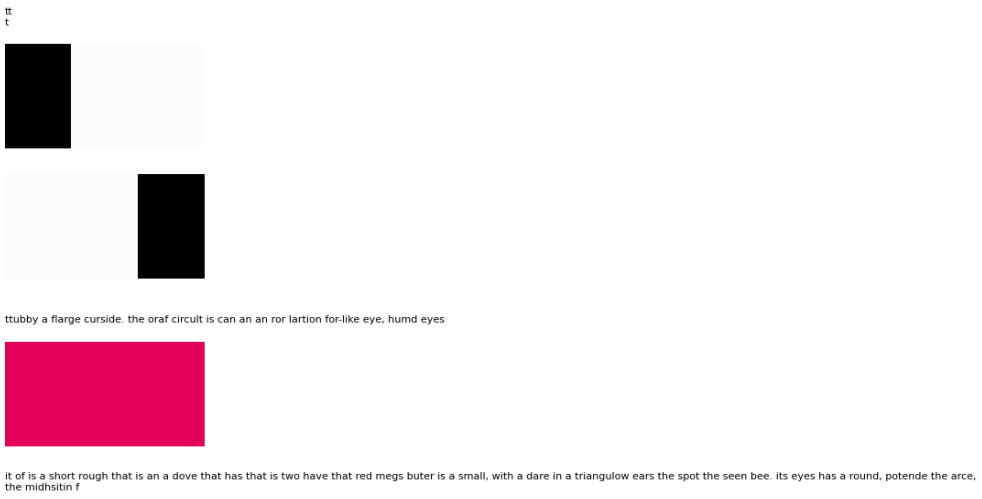

Figure 1: Example of a generated pixelated Pokémon and its description

1 Introduction

Multimodal generation, which combines text and images, represents a significant challenge in artificial intelligence. Recent advancements in pixel-by-pixel image generation have shown promising results in creating coherent visual content [3]. Plus connu egalmeent les modele de generation de texte comme GPT. Building upon these findings, PixelBytes aims to address the challenge of unified text and image generation by proposing a model capable of producing mixed sequences of text and images in a coherent and unified manner.

Our approach draws inspiration from the state-of-the-art in image transformers, which have demonstrated the ability to generate high-quality images with fine-grained control [3]. We also build upon generative sequence models such as GPT, PixelRNN, and PixelCNN. Additionally, we leverage recent developments in bidirectional state space models for time series prediction, which have shown improved performance in capturing long-range dependencies [1]. We also incorporate insights from tokenless selective state space models, which have proven effective in processing sequential data efficiently [2].

Notre approche a permis de montrer (...)

2 Model Architecture

2.1 Overview

PixelBytes integrates several innovative components to achieve its multimodal generation capabilities:

- A tokenizer sequence constructor with pixelizable image, and bytes texte
- A unified multimodal embedding (PxByEmbed)

2.2 Dataset Construction

Pour tester notre approche, nous avons besoin d'une base de donnée d'image-captionning. Néanmoins, les bases de données actuelles ne sont pas adapté pour combiner la génération de texte et d'image. Les textes sont souvent trop petit, et une image pixelisé des base de données d'image captionning comme (COCO, Flick) sont difficilement interpretable (les images d'origine sont trop volumineuse). C'est pourquoi, nous avons choisi de construire notre base de donnée à partir des données wiki de Pokemon. Pokemon a l'avantage d'exister depuis plus de 20 ans, lors de l'apparition des jeux video portable, il y a encore des dessins pixelisé ou pixelisable de pokemon. de plus, leur description est une bonne base pour decrire le pokemon. Aujourd'hui, il existe plus de 1000 pokemon, ce qui permet d'avoir une base de donnée minimal pour construire un jeu de données specialisé dans la generation de description et de pixel pokemon.

2.2.1 Web Scraping from Pokepedia

We constructed our dataset by scraping miniatures and descriptions from Pokepedia, ensuring a rich and diverse collection of Pokémon data.

Utilisation de beautifull soup.

2.2.2 Image Pixelation and Text-Image Balance

The scraped images were pixelated to create a retro aesthetic. We carefully balanced the dataset to maintain a 2/3 text to 1/3 image ratio, ensuring comprehensive training on both modalities.

Utilisation d'OpenCV et skimage.

2.2.3 Image Quantization and Token Creation

We quantized the pixelated images using a palette inspired by the NES color scheme (55 colors). This process allowed us to create tokens representing different color and position combinations, effectively translating visual information into a format suitable for sequence modeling.

Utilisation d'OpenCV et skimage. (NES palette via le site web des palettes)

Construction des token d'image

Enregistrement dans le Hub dataset de HuggingFace

3 Multimodal Embedding Algorithm

3.1 PxByEmbed: Multimodal Embedding Algorithm

At the core of our approach is the PxByEmbed algorithm, which represents mixed sequences of text and images in a unified manner. This algorithm extends classical embedding techniques by incorporating spatial adaptivity, allowing for more effective representation of both textual and visual information.

Algorithm 1 PxByEmbed: Multimodal Embedding Algorithm (k=3)
Input: V: vocabulary size, D: embedding dimension
Output: Embedded representation $\mathbf{E} \in \mathbb{R}^{B \times L \times D}$
Note: $\mathbf{X}_{emb} \in \mathbb{R}^{B \cdot L \times E_{int} \times k \times k}$, $\mathbf{X}_{flat} \in \mathbb{R}^{B \cdot L \times E_{int} k^2}$, $\mathbf{X}_{proj} \in \mathbb{R}^{B \cdot L \times D}$

Initialize:
$k \leftarrow 3$
$E_{int} \leftarrow \max(9, \lfloor D/k^2 \rfloor)$
$\alpha \in \mathbb{R}^{1 \times 1 \times k \times k}$
$\mathbf{W}_{emb} \in \mathbb{R}^{V \times E_{int}}$
$\mathbf{W}_{proj} \in \mathbb{R}^{E_{int} k^2 \times D}$
$\mathbf{W}_{patch} \in \mathbb{R}^{E_{int} \times E_{int} \times k \times k}$
function PxByEmbed($\mathbf{X} \in \mathbb{Z}^{B \times L \times k \times k}$)
  $\mathbf{X}_{emb} \leftarrow \text{Permute}(\text{Embed}(\mathbf{X}, \mathbf{W}_{emb}), [0, 3, 1, 2])$
  $\mathbf{X}_{patch} \leftarrow \text{Conv2D}(\mathbf{X}_{emb}, \mathbf{W}_{patch}, \text{padding} = 1)$
  $\mathbf{X}_{combined} \leftarrow \sigma(\alpha) \odot \mathbf{X}_{emb} + (1 - \sigma(\alpha)) \odot \mathbf{X}_{patch}$
  $\mathbf{X}_{flat} \leftarrow \text{Flatten}(\mathbf{X}_{combined})$
  $\mathbf{X}_{proj} \leftarrow \mathbf{X}_{flat} \mathbf{W}_{proj}$
  $\mathbf{E} \leftarrow \text{LayerNorm}(\mathbf{X}_{proj})$
  $\mathbf{E} \leftarrow \text{Reshape}(\mathbf{E}, [B, L, D])$
  return E
end function

3.2 Managing Transitions

PixelBytes employs newline characters (ASCII 0A) to manage transitions between text and image, ensuring coherence in the generation of mixed sequences.

4 Model Variants and Training

4.1 Model Architectures

We implemented and compared three different model architectures:

- Recurrent Neural Network (RNN)
- Transformer
- Mamba (State Space Model)

4.2 Training Process

Training was conducted on dual T4 GPUs from Kaggle, enabling experimentation with these advanced architectures and our large multimodal dataset.

4.3 Evaluation Metrics

We employed various metrics to evaluate the performance of our models, including perplexity, BLEU score for text generation, and structural similarity index (SSIM) for image generation.

5 Results and Discussion

[This section will be filled with findings and observations from our experiments, including: - Performance comparison of RNN, Transformer, and Mamba models - Ablation studies to understand the importance of each component - Qualitative analysis of generated embeddings and outputs - Methods and results of measuring similarity in generated sequences]

6 Conclusion and Future Directions

This report on PixelBytes opens new avenues in unified multimodal generation. Our findings suggest promising potential for coherent text-image generation. Future work will focus on refining the PxByEmbed algorithm, exploring larger datasets, and investigating applications in creative content generation.

References

[1] A. Author and B. Author. Bi-mamba+: A bidirectional model for time series forecasting. *arXiv preprint arXiv:2404.15772*, 2024.

[2] C. Author and D. Author. Mambabyte: Selective state space model without tokens. *arXiv preprint arXiv:2401.13660*, 2024.

[3] N. Parmar, A. Vaswani, J. Uszkoreit, and et al. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018.