Text2Illustration: Sampling and Composing Creatives With Text

Geethu Miriam Jacob[1] Sagar Sahu[1] Chisato Ito[2] Masashi Kawamura[2] Björn Stenger[1]
[1] Rakuten Institute of Technology, Rakuten Group, Inc.
[2] Creative Design Strategy Department, Rakuten Group, Inc.
geethu.jacob|sagar.a.sahu|bjorn.stenger@rakuten.com

Figure 1: Our Text2Illustration system generates visuals from user text input, using graphical components.

ABSTRACT

We introduce Text2Illustration, a user-friendly system that generates illustrations from plain text input. This system addresses the need for an accessible tool to create illustrations that vividly portray a range of human activities. Text2Illustration uses a large language model (LLM) to select relevant components from an SVG library based on the input text, seamlessly composing new visuals. To ensure ease of use, we have developed a simple text-based interface, allowing users to describe their desired illustration.

CCS CONCEPTS

• **Applied computing** → *E-commerce infrastructure*; • **Human-centered computing** → **Text input**; **Visualization toolkits**.

KEYWORDS

Scalable Vector Graphics, Large Language Models

ACM Reference Format:
Geethu Miriam Jacob[1] Sagar Sahu[1] Chisato Ito[2] Masashi Kawamura[2] Björn Stenger[1] . 2024. Text2Illustration: Sampling and Composing Creatives With Text. In *7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD) (CODS-COMAD 2024), January 4–7, 2024, Bangalore, India.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3632410.3632484

CODS-COMAD 2024, January 4–7, 2024, Bangalore, India

ACM ISBN 979-8-4007-1634-8/24/01.
https://doi.org/10.1145/3632410.3632484

1 INTRODUCTION

Illustrations serve as valuable tools for simplifying complex concepts and enhancing the appeal of presentations and websites. Traditionally, creating illustrations has relied on commercial tools tailored for Scalable Vector Graphics (SVGs), demanding expertise and substantial time investments in manually selecting and assembling various components. To tackle this challenge, automated systems capable of generating illustrations from textual descriptions have garnered significant attention. In this context, we introduce *Text2Illustration*, a novel approach to illustration generation driven by textual input. Text2Illustration harnesses a meticulously curated library of brand-approved visual components, all in the form of Scalable Vector Graphics (SVGs). These components, contributed by designers, ensure a consistent visual style across the library.

At the heart of Text2Illustration lies its capacity to discern the most pertinent components from the library based on user-supplied textual descriptions. This selection process is facilitated by a large language model (LLM) adept at identifying graphical elements that best represent the user's specified activities or attributes. These components are seamlessly merged to create visually coherent and engaging illustrations.

Our method empowers users to simply describe their desired use-case in plain text, eliminating the need for manual component selection. The approach comprises a well-defined pipeline encompassing data pre-processing, prompt engineering, part sampling, and part composition. By reducing the time and effort traditionally required for illustration creation, we aim to make this tool accessible to a wide range of users.

In this paper, we provide an overview of our approach, delineating the key components and methodologies involved. Furthermore, we showcase the efficacy of our method with visuals from our demonstration system, highlighting its ability to generate digital illustrations adhering to brand standards, such as color schemes and visual consistency.

2 BACKGROUND

Vector Representation. Scalable Vector Graphics (SVG) is a web-friendly vector file format that defines graphics using XML format. SVG images are not only scalable and zoomable but also easily indexable and compressible. The components in our library are shapes defined by paths using control points. SVG files can be directly opened in any web browser. In our work, we combine multiple SVG files into one to create illustrations.

Large Language Models. Large language models (LLMs) [1], trained on extensive text corpora, have demonstrated excellent performance on zero-shot tasks, particularly in conversation systems. In this context, a prompt is a set of instructions given to an LLM to obtain customized outputs. The quality of prompts significantly impacts the usefulness of the outputs generated by a conversational LLM. Several recent LLMs have been introduced [6, 10], and API calls simplify the process of interfacing with LLMs, making application development more accessible and convenient.

Text to Illustration Generation. Previous work has employed pre-trained vision-language models to aid in the creation of vector graphics. For instance, VectorAscent [4] and CLIPDraw [3] optimize CLIP's image-text similarity metric to generate vector graphics based on text prompts. StyleCLIPDraw [9] extends CLIPDraw by incorporating an auxiliary style loss with a pretrained VGG16 model. Arnheim [2] utilizes a neural network to parameterize SVG paths, while VectorFusion [5] employs stable diffusion [11] to generate SVG images from text. Our method, however, does not create SVG images from scratch but rather selects existing components from a library of parts using LLMs.

3 RAKUTEN ILLUSTRATION SYSTEM

The Rakuten Illustration System [8] comprises of a library of unique assets crafted by experts [7] using specialized commercial tools. These assets empower users to create human figures engaged in various activities, enhancing brand recognition. These versatile creatives can be seamlessly integrated into web pages to elevate brand representation.

The Illustration System is built upon a diverse collection of asset components, each designed by design experts. These components are categorized into four groups: head, body, legs, and objects, with a wide array of options available within each group. Users can browse these groups, selecting components that best align with their creative requirements. With assets offering a broad spectrum of poses and scenarios, users can easily craft a diverse range of visuals, with a wide range of physical attributes and cultural backgrounds. Designers navigate these folders of parts, select and compose them on a canvas manually. With over 300,000 unique combinations available, navigating the extensive folder structure and selecting matching components can be time-consuming and overwhelming, especially for novice users.

4 METHOD

We present Text2Illustration — a method that streamlines the process of generating illustrations. It accepts textual descriptions as input and generates corresponding visual output. Text2Illustration extracts attributes from the input text and proceeds to sample components from the part library. These components are then combined to produce the final visual creative. The workflow is shown in Figure 2.

Our method introduces several noteworthy contributions: Firstly, we address the challenge of generating assets from pre-approved part-assets using versatile textual input. This approach accommodates a wide range of creative descriptions, offering flexibility. Moreover, selecting the most pertinent part-assets based on the provided textual description and combining them to generate a new asset. Additionally, our approach allows users to apply different colors, enabling modifications to the dominant color or specific body part hues, thereby ensuring alignment with brand guidelines.

The solution is structured into three distinct steps: component naming, prompt engineering, and text-to-illustration inference.

4.1 Component Naming

To ensure the effectiveness of SVG search, it is important to assign meaningful names to the components. We follow a semantic naming convention, where the names of part-assets carry descriptive information about their contents. We systematically assign specific IDs to the paths within SVG files. These IDs, such as "skin," "hair," "shirt," "skirt," "shoes," and more, help identify individual components. Furthermore, our library encompasses components designed for both front and side-facing visuals. For instance, consider the SVG component named "BodyFront_Holding_Scanner_Apron3.svg" represents a front-facing body component holding a scanner and wearing an apron. Within our library, there is an assortment of attire options, including 8 different types of tops and bottoms, along with 42 distinct top activities and 10 varied bottom activities.

4.2 Prompt Engineering

In our chat-based system, we make use of an LLM (ChatGPT 3.5) to help selecting the appropriate part-asset files based on textual descriptions. We create a simple prompt template that includes all the brand-approved part-assets. This template combines the user's input with our predefined format and feeds it into the LLM. The prompt follows this format:

"Output a JSON with fields, select 'who', 'top_body_activity' and 'bottom_body_activity' from the following lists. keys 'gender', 'who', 'top_body_activity', 'bottom_body_activity', 'top attire', 'bottom attire'. Do not add any new keys, other than the specified, to the JSON. Select most semantically similar values from the lists,

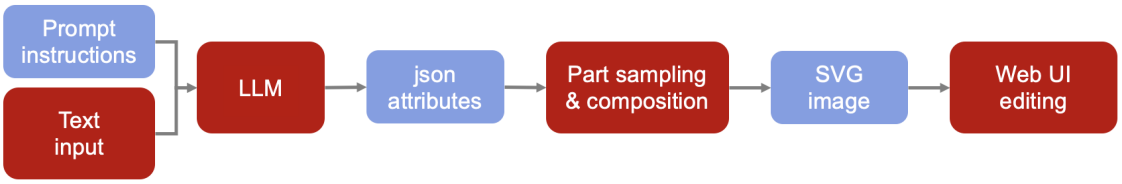

Figure 2: Proposed workflow: Prompt instructions and text input are sent to an LLM, which outputs a JSON file. The components are sampled from the JSON attributes and composed to obtain the final SVG image that can optionally be edited.

and do not change the names of values. Use gender appropriate attire. Select most semantically similar values to the message for all the values from the corresponding lists, and do not change the names of values. The selected values of 'top_body_activity' and 'bottom_body_activity' should be logically consistent.

who: [*list of all head components*]
top_body_activity: [*list of all top body activities*]
bottom_body_activity: [*list of all bottom body activities*]
top attire: [*list of all top attires*]
bottom attire: [*list of all bottom attires*]"

Consider an example where the user inputs "A woman in formal wear, thinking." The following is a JSON provided as output:

```
{
  "gender": "Woman",
  "who": "Woman_PonyTail",
  "top_body_activity": "Speaking_Thinking",
  "bottom_body_activity": "Rest",
  "top_attire": "SuitWoman",
  "bottom_attire": "PencilSkirt"
}
```

We use two different prompts for side-facing and front-facing creatives. The instructions of the prompt remain the same, except for the list of components in the fields, 'who', 'top_body_activity' and 'bottom_body_activity'. This is done to avoid mixing up of components of front-facing and side-facing creatives. Thus, user has to select the type of creative he needs, before he enters the prompt. The output JSONs may vary in different runs when the user prompt only includes information about certain fields. Random selection is used for fields where no information is provided, resulting in the possibility of generating multiple outputs.

4.3 Text-based Editing

We can also provide feedback in the chat-based system, allowing users to easily and intuitively update the illustration. Once we obtain the initial JSON output and observe the SVG created, we have the option to edit the JSON using text. For example, in the above instance, we can provide feedback such as "Change her hairstyle" and "The woman is wearing a cardigan." The JSON is then updated to incorporate the feedback as follows:

```
{
  "gender": "Woman",
  "who": "Woman_CurlyHair_HeadBand",
  "top_body_activity": "Speaking_Thinking",
  "bottom_body_activity": "Rest",
  "top_attire": "CardiganWoman",
  "bottom_attire": "PencilSkirt"
}
```

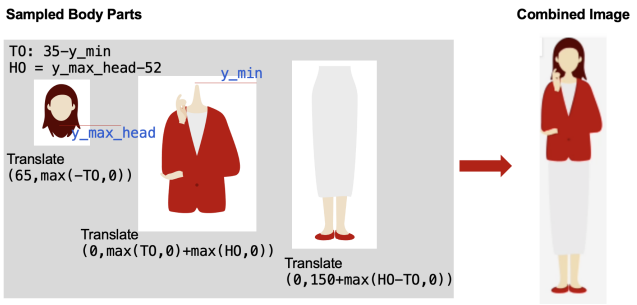

Figure 3: Sampled components are combined by translating them to the correct coordinates to compose the final image.

4.4 Part Sampling and Composition

In our inference pipeline, we take user text input, combine it with predefined instructions, and process it through an LLM. The expected output is a JSON format with values for fields like "who," "top_body_activity," "bottom_body_activity," "top_attire," and "bottom_attire." Using these final field values, we randomly sample components from our library and assemble them. This implies that the same user query may return different illustrations. For sampling, we combine the JSON values to match the format of the assigned names of the components. The selected components are then composed together to obtain the full creative.

Figure 3 illustrates the composition process. Defining the bottom y-coordinate of the skin in the head as $y_{\text{max\_head}}$ and the start of the neck portion of the body component as y_{min}. We compute $T_0 = 35 - y_{\text{min}}$ and $H_0 = y_{\text{max\_head}} - 52$. We translate the head part-asset to $(65, \max(-T_0, 0))$, the top part-asset to $(0, \max(T_0, 0) + \max(H_0, 0))$, and the bottom part-asset to $(0, 150 + \max(H_0 - T_0, 0))$ to create the final SVG.

Additionally, apart from SVG creation and editing, we offer the ability to change asset colors using brand color schemes. Users can

(a) (b) (c)

Figure 4: Text2Illustration examples, showing (a) user-supplied input text, sampled attributes in JSON format, and the output image of a front-facing pose and (b) a sidefacing pose. (c) In an optional editing step, users can change the color scheme or the colors of individual components.

modify the dominant color by selecting from a different color set and also adjust colors for specific parts.

5 USER INTERFACE

Our system incorporates a Gradio interface featuring two tabs: one designated for SVG creation and the other for SVG editing. Within the 'SVG Creation' tab, text descriptions can be entered. The tab displays the prompt, the text input, and the resulting JSON output, allowing for continuous text-input to update the JSON. The 'SVG Editing' tab facilitates adjustments to the dominant color and specific part colors, where colors can be selected from a brand-consistent color palette.

Figure 4 shows screenshots of our demonstration. Figure 4(a) and (b) show the SVG Creation tab, where (a) shows a creative with front-facing pose, and (b) shows a side-facing pose. Figure 4(c) shows the SVG editing tab with the ability to change the dominant color, skin color, and hair color. The illustration on the right reflects the updated colors of skin, hair, and attire.

6 CONCLUSION

Text2Illustration is an innovative system that generates illustrations from text. It leverages a large manguage model (LLM) to select relevant components from a pre-approved illustration library and seamlessly composes them to create diverse illustrations depicting various activities. Users can describe their desired use-case in text, and the system automates the creative generation process, making illustration creation more accessible and efficient for a wider range of users.

REFERENCES

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv:2108.07258* (2021).

[2] Chrisantha Fernando, SM Eslami, Jean-Baptiste Alayrac, Piotr Mirowski, Dylan Banarse, and Simon Osindero. 2021. Generative art using neural visual grammars and dual encoders. *arXiv:2105.00162* (2021).

[3] Kevin Frans, Lisa Soros, and Olaf Witkowski. 2022. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *NeurIPS* 35 (2022), 5207–5218.

[4] Nisha Huang, Fan Tang, Weiming Dong, and Changsheng Xu. 2022. Draw your art dream: Diverse digital art synthesis with multimodal guided diffusion. In *Proc. ACM Int. Conf. Multimedia*. 1085–1094.

[5] Ajay Jain, Amber Xie, and Pieter Abbeel. 2023. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *CVPR*. 1911–1920.

[6] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS* 35 (2022), 27730–27744.

[7] Rakuten Design. accessed Sep 14, 2023. https://www.rakuten.design/

[8] Rakuten Illustration System. accessed Sep 14, 2023. https://corp.rakuten.co.jp/news/update/2022/1007_01.html

[9] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. 2022. Styleclipdraw: Coupling content and style in text-to-drawing translation. *arXiv:2202.12362* (2022).

[10] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv:2302.13971* (2023).

[11] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*. 8798–8807.