Prompt 추천 기반의 이미지 생성 서비스

유민경
홍익대학교 게임학부
mingyeongyu8@gmail.com

Image Generation Service Based on Prompt Recommendation

Yu Min-kyeong
Hongik University, School of Games

요 약

본 논문은 게임 일러스트 이미지를 파인 튜닝한 Stable Diffusion 모델을 활용하여 사용자가 희망하는 분위기의 게임 일러스트 이미지를 생성할 수 있도록 Prompt를 추천하고 이미지를 생성하는 서비스를 제안한다. 게임 장르 인기 순위 10가지 중 3가지 MMORPG, 샌드박스, FPS 중 하나를 택해 이미지를 생성할 수 있다. 10명의 참가자를 모집하여 간단한 예비 실험을 진행한 결과, 서비스에 대해 80%의 만족도를 보였고 Prompt 추천 기능에 대해 70%의 만족도를 보였다. 불만족한 사용자의 설문 조사 결과, Prompt의 인물과 배경 그리고 사물을 세분화할 필요성과 장르적 선택의 폭을 더 넓혀 추천해줄 수 있다면 서비스에 대해 더 높은 만족도를 확보할 수 있음을 확인했다.

1. 서 론

이미지 생성이란 새로운 이미지를 만들어내는 기술로, 최근 Dalle2, GPT-3 등과 같은 텍스트를 이미지로 변환하는 모델의 성능이 향상되며 Text-to-Image 연구가 활발히 진행되고 있다[1]. Text-to-Image는 텍스트로부터 적절한 이미지를 생성하는 멀티모달 러닝의 방식 중 하나이다[2]. 초기 Text-to-Image 연구는 CNN(Convolutional Neural Network)와 LSTM(Long Short Term Memory)를 융합해 사용하여 텍스트에서 특징을 추출하고 GAN(Generative Adversarial Networks)를 사용해 이미지를 생성했다. 이러한 방식은 텍스트의 특징을 이미지에 제대로 반영하지 못해 품질이 저하되는 문제가 있었다. 그 후 bi-directional LSTM등을 사용해 텍스트의 특징을 반영하려 했고 다단계의 개선이 이뤄졌다. 이미지를 표현하는 세분화된 단어들을 수학적으로 표현해 어텐션이란 개념을 도입해 AttnGAN(Attentional Generative Adversarial Networks)알고리즘을 Text-to-Image에 사용해 여러 이미지 생성 모델들을 만들게 되었다[3].

이처럼 텍스트 문장 내용만으로 이미지를 생성할 수 있게 되었으나 부정확한 결과물, 저작권 논란 등을 이유로 '서비스화'와 관련된 연구는 부진한 편이다. 따라서, 본 연구에서는 많은 이미지가 요구되는 게임을 장르로 하여 기획자 혹은 그래픽 작업자가 본격적인 작업에 들어가기 전 레퍼런스 이미지 자료가 필요할 때, Text-to-Image를 사용해 편의성을 제공하는 서비스를 고안한다. 게임 개발에 많이 사용되는 엔진, Unity 내부에서 사용할 수 있도록 하고 무료로 사용 가능하며 모델에서 생성된 이미지에 대한 사용 권한을 자유롭게 부여하고 있는 Stable Diffusion 모델을 사용해 서비스를 개발한다. 또한, 일종의 설문지 체크를 통해 사용자가 희망하는 분위기의 이미지를 생성할 수 있도록 Prompt를 추천하는 기능을 추가한다.

2. 관련 연구

2.1 Text-to-image

Text-to-image의 대표 모델로는 GPT-3를 적용한 Open AI의 DALL-E와 LDM(Latent Diffusion Model)를 적용한 Stable Diffusion, 디스코드 서버에서 이뤄지는 Midjourney등이 있다. Midjourney는 만화적인 이미지 표현에 특화되어 있으며 사용법이 간단하고[4], DALL-E의 경우 수백만개의 스톡 이미지로 학습되어 출력이 정교하고 전문적이란 평가를 받고 있다[5]. 이 두 모델과 달리 Stable Diffusion은 오픈 소스 모델로 일러스트적 표현의 이미지를 생성 시 좋은 결과를 얻을 수 있다[6]. 본 연구에서는 오픈 소스인 Stable Diffusion을 크롤링한 게임 일러스트로 파인 튜닝해 사용한다.

Text-to-Image는 입력된 텍스트 프롬프트의 시각적 특징을 추출해 이미지를 생성한다. 시각적 특징을 추출하기 위해 CNN-LSTM, bi-directional 등을 사용하고, 추출한 텍스트의 특징을 DCGAN(Deep Convolutional GAN)에 입력하면 이미지가 생성된다. 초기에 사용했던 CNN-LSTM으로 텍스트의 특징을 추출하고 DCGAN으로 이미지를 생성하는 방식은 이미지 품질이 낮고 텍스트의 특징을 나타내는데

어려움이 있었다. 최근에는 bi-directional LSTM과 AttnGAN, DM-GAN등을 사용해 이를 보완하고 있으며, 이미지의 시각적 특징 추출을 위한 수학적인 어텐션 매커니즘 적용 또는 DALL-E처럼 하나의 데이터 소스로부터 transformer를 활용하여 Text-to-Image 태스크를 auto-regressive하게 모델링하는 zero-shot learning[5]을 사용하는 등 성능 보완 및 최적화를 위해 다양한 시도를 하고 있다.

2.2 키워드 추출 연구

키워드 추출은 문장을 가장 잘 설명하는 단어를 텍스트에서 추출하는 기법을 말한다. 초기에는 키워드 추출을 위해 단어 빈도수를 활용했으며, 요즘에는 Textrank를 활용해 두 단어를 기준으로 크기 N 이내의 단어들을 그래프로 그리는 방식을 사용한다. 이처럼 그래프를 그릴 경우, 그래프에 추가된 정점은 명사, 동사 등 특정 어휘만 사용할 수 있는 필터를 적용해 표현할 수 있기 때문에 문장의 관계를 파악하는데 활용될 수 있다[7]. Textrank는 키워드 뿐만 아니라 문장 단위로도 추출할 수 있으며, 두 방법 모두 정점에 필터를 적용하기 위해 텍스트를 토큰화하는 사전 처리 작업을 거친다. 이 단계에서는 단일 단어만을 고려하며 다중 단어는 필터를 통과한 단어를 정점에 추가하고 범위 내에 엣지를 추가하는 후처리 과정에서 재구성한다. 각 정점은 초기값 1을 가지며 정점은 점수의 역순으로 정렬하여 상위 T 개의 정점을 후처리에서 활용하기 위해 유지시킨다. 일반적으로 T는 5~20개의 키워드 범위를 갖는다.

WordRank는 띄어쓰기가 없는 일본어와 중국어의 단어 인식에 적합한 방법으로 외부에서 제안된 외부 경계값과 내부 경계값을 모두 활용해 단어를 인식하는 비지도학습 방법을 말한다. 그 중 KR-WordRank는 한국어의 특징을 반영하여 비지도학습 기반으로 한국어로 구성된 문장의 키워드를 추출하는 방법이다[8]. 본 연구에서는 lexica에서 게임 장르별 이미지를 검색 후, 적절한 이미지를 수동 선별해야 하고 입력값에 대한 분명한 목표치가 있는 것이 아니기에 지도학습과 강화학습 보다는 비지도 학습이 적합하다. 때문에 데이터가 어떻게 구성되었는지 대략적으로 키워드 추출을 할 수 있는 KR-WordRank를 사용하여 Prompt로 추천할 단어를 추출했다.

3. 시스템 설계 및 구현

본 연구의 Prompt 추천 기반의 이미지 생성 서비스는 크게 그림 1과 같이 5단계를 거친다. 서비스가 제공될 환경이 주로 게임이기 때문에, Stable Diffusion을 게임 일러스트 이미지에 맞춰 파인 튜닝하는 작업이 필요하다. 학습에 사용될 이미지를 python의 beautiful soup과 google images download 라이브러리를 사용해 크롤링한다. 그림 2의 크롤링 코드 예시를 적절히 변형해 크롤링을 진행하면, Stable Diffusion Training Data 루트 디렉터리 하위에 만들어두었던 폴더에 이미지들이 저장된다. 파인 튜닝을 진행하기 전, 이상한 이미지들을 수동으로 삭제 후, Stabe Diffusion 학습의 매개 변수 batch size, num workers, max steps, configs등을 조정하고 학습을 진행한다. 다음으로는 Prompt 추천을 위한 텍스트 데이터 크롤링을 위해 lexica 사이트에서 사전에 지정해둔 게임 장르 세 가지, MMORPG, 샌드박스, FPS를 검색한다. 게임 장르 선정 기준은 게임 장르 인기순위 상위 10가지 중 세가지를 택해 사용했다[9].마찬가지로 전자의 방법을 활용해 텍스트를 크롤링하고, 적합하지 않은 이미지에 대한 Prompt는 수동으로 삭제를 진행한다.

그림 3처럼 KR-WordRank를 활용해 키워드 추출하는데 필요한 키워드 추출 함수, 전처리 함수 등을 구현하여 장르별 추출된 키워드를 바탕으로 추천할 tag를 생성한다. 공통적으로 4k, 카메라 렌즈 버전 명시, hyperrealistic, 게임 장르에 해당하는 대표 게임과 같은 단어가 들어갈수록 이미지가 사실적이고 정교하게 나왔다. MMORPG의 경우 배경 이미지와 직업 클래스의 이름이 들어갔을 때 좋은 결과를 보였다. 샌드박스의 경우, isometic, smmoth, row poly를 입력했을 때, FPS의 경우, Screenshot of video game, 총 이름, 자세를 입력했을 때 좋은 결과를 얻을 수 있었다.

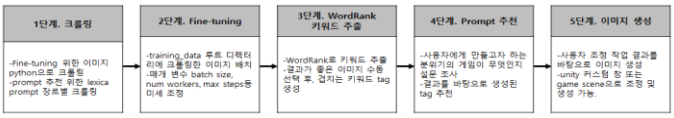

그림 1. Prompt 추천 기반의 이미지 생성 서비스 5단계

```
from bs4 import BeautifulSoup
from urllib.request import urlopen
from google_images_download import google_images_download

response = google_images_download.googleimagesdownload()

arguments = { "keywords":"검색할 키워드 입력",
              "limit" : "최대 개수", "print_urls" : True,
              "format" : "jpg" }

    paths = response.download(arguments)
    print(paths)
```

그림 2. 크롤링 코드 예시

```
from krwordrank.sentence import summarize_with_sentences
from konlpy.tag import Kkma
from konlpy.tag import Okt
import sys
import os
import re
```

생략

```
for idx in str_list[5:]:
    print(idx)
    stop += 1
    tmp = strlist[idx]
    st = ''
    for i in range(len( tmp)):
        texts = tmp[i]
        texts = preprocessing(texts, idx)
        st += texts
    texts = st.split('. ')
    try:
        stopwords = {idx.split(' ')[0], idx.split(' ')[1]}
        keywords, sents = summarize_with_sentences(
                                                texts,
                                                stopwords = stopwords,
                                                num_keywords=100,
                                                num_keysents=10
                                                )
    except ValueError:
        print('key == null')
        print()
        continue

    for word, r in sorted(keywords.items(), key=lambda x:x[1], reverse=True)[:7]:
```

그림 3. KR-WordRank 키워드 추출 코드 예시[10]

3.1 Prompt 추천 시스템 설계 및 구현

프롬프트 추천 시스템은 그림 1의 서비스 5단계 중 4번째에 해당한다. 사용자에게 만들고자 하는 게임의 장르와 분위기가 어떤 것인지 설문 조사를 진행하고, 이를 바탕으로 Prompt에 입력하면 좋을 텍스트를 tag로 추천해준다. Prompt 추천 시스템은 그림 4의 프로세스에 따르며, 완성된 화면은 그림 5와 같다.

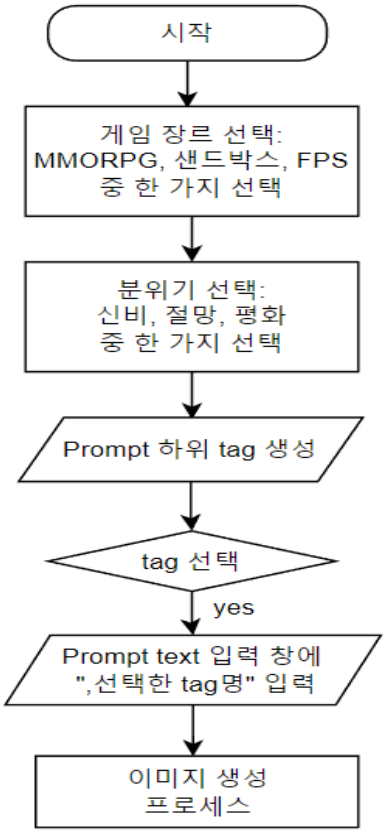

그림 4. Prompt 추천 시스템 프로세스

그림 5. Prompt 추천 위한 설문 조사 창

3.2 이미지 생성 시스템 설계 및 구현

이미지 생성 시스템은 그림 1의 서비스 5단계 중 5번째에 해당한다. 이미지 생성 시스템의 설계 및 구현은 두가지 방식으로 구성된다. 첫번째는 유니티 게임 씬에서 바로 작업할 수 있도록 설계 및 구현한다. 두번째는 유니티 씬 버전과 Editor window를 상속받고 유니티의 Editor GUILayout을 사용해 Toolbar에서 Window/StableDiffusion을 선택해 이미지 생성 작업을 실행할 수 있도록 설계 및 구현한다. 본 논문에서는 이를 유니티 커스터마이징 버전이라고 표현한다. 두 버전 모두 Stable Diffusion 모델을 사용하기 위해 파인튜닝을 진행한 Stable Diffusion 모델의 가중치가 담긴 ckpt 파일을 다운로드 받고, Python을 사용하여 모델을 로드한다. 유니티에서 Python을 실행하려면 유니티를 실행 후 파이썬 실행 프로세스를 코드로 활성화시켜 진행할 수 있다. Python의 환경 경로와 실행할 파일 경로를 입력하는 것에 주의한다.

유니티 씬 버전의 구현 결과는 그림 6과 같다. Prompt를 입력할 수 있는 text창과 Sampling Steps, Width, Height 등을 조정할 수 있는 progress bar 그리고 Sampling Method를 선택할 수 있는 checkbar 총 3가지의 방법으로 구성된다. Generate를 1회 실행 후 버튼은 Retry로 바뀌게 되며, 두 버튼을 누를 경우 클라이언트에서 조정한 이미지 생성 조건을 Python으로 실행한 모델에게 전달해 이미지 생성을 진행한다.

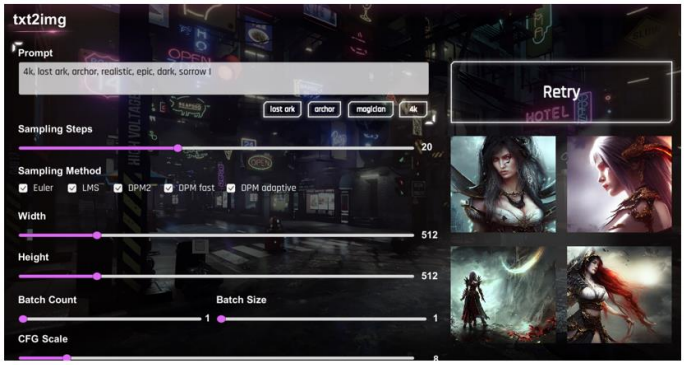

그림 6. 유니티 씬 버전 구현 결과

유니티 커스터마이징 버전의 구현 결과는 그림 7과 같다. 유니티 씬 버전과 마찬가지로 Prompt를 입력할 text창, progressbar, checkbar 총 3가지의 방법으로 구성된다. Editor Window 상속 후, EditorGUILayout으로 위의 3가지 방식을 사용해 사용자가 이미지 생성 조건을 바꿀 수 있도록 틀을 생성한다. 첫번째 방식과 모델의 동장 방식은 유사하며, 핵심 코드는 그림 8과 같다. Editor의 로컬 환경 테스트를 위한 부분은 lstein-stable-diffusion/scripts/dream.py[11]를 참고하여 구현하였다.

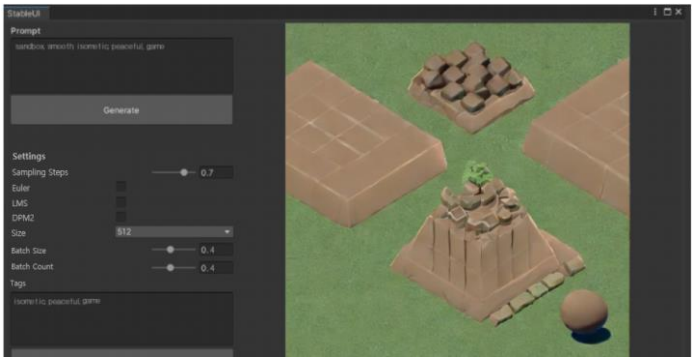

그림 7. 유니티 커스터마이징 버전 구현 결과

```
    EditorGUILayout.BeginVertical("box", GUILayout.ExpandWidth(true));

    EditorGUILayout.LabelField("Results", EditorStyles.boldLabel);

    if (resultsTexture != null) GUI.DrawTexture(tpos, resultsTexture, ScaleMode.ScaleToFit);

    EditorGUILayout.EndHorizontal();
}

참조 1개
void Generate()
{
    Debug.Log("--Generating--");
    SaveResults();

    string fitStr = fit ? "on" : "off";

    //result of image generation, resize
    int width = input_height;
    int height = input_height;

    string initImgObj = "";

    // convert jpeg
    if (initimg == null)
    {
        Object temp = null;
        initImgObj = SimpleJsonConverter.Serialize(temp);
        Debug.Log("--Init Img: null--");
        //return;
    }
    else
    {
        if (initimg.isReadable == false)
        {
            initimg = DuplicateTexture(initimg);
        }

        byte[] bytes = initimg.EncodeToJPG();
        string base64 = System.Convert.ToBase64String(bytes);
        initImgObj = "\"" + base64 + "\"";
    }
    //get prompt -> reference prompt.cs
    string postData = $"{{'prompt':'{prompt}','iterations':'{iterations}','steps':'{steps}','configs':'{c
    postData = postData.Replace("'", "\"");
    //img2img
    var strengthClamped = Mathf.Clamp(strength, 0, 0.9999999f);
    var request = (HttpWebRequest)WebRequest.Create(url);
    var data = Encoding.ASCII.GetBytes(postData);
    request.KeepAlive = true;
    request.Method = "POST";
    //stable diffusion 모델을 돌리려면 webui 연결, dreamstudio 나중에 사용해보기
```

그림 8. 유니티 에디터 커스터마이징 핵심 코드

4. 결과

Prompt 추천 기반의 이미지 생성 서비스를 구현 후, 실제 편의성 개선에 도움이 되었는지 혹은 새로웠는지를 평가하기 위해 기획자 지망생과 그래픽 분야의 총 10명(기획자 4, 그래픽 6, 남 3, 여 7)의 참가자를 모집해 간단한 예비 실험을 진행했다. 참가자들은 서비스를 사용 후, 구글 서베이 폼으로 작성된 설문지에 응답한다. 그 결과, 그림 10에서 확인할 수 있듯 서비스에 대해 80%의 만족도를 보였고 Prompt 추천 기능에 대해 70%의 만족도를 보였다. 유니티 씬 버전과 유니티 커스터마이징, 두 버전의 만족도는 씬 버전이 90%로 높은 결과를 보였다.

불만족한 사용자를 대상으로 한 추가 설문 조사 결과, “Prompt의 인물과 배경 그리고 사물을 세분화할 필요성이 있다”와 “장르적 선택의 폭을 더 넓히면 좋을 것 같다.”란 의견이 있었다. 이들을 보완한다면 서비스에 대해 더 높은 만족도를 확보할 수 있는 것으로 보인다.

그림 9. 구글 서베이 폼

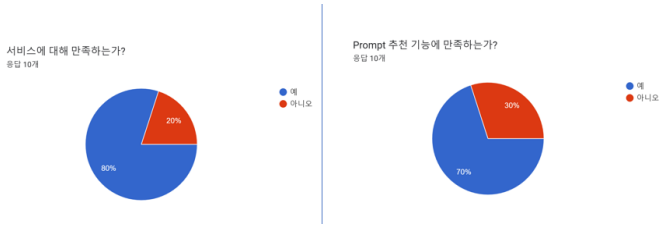

그림 10. 서비스 만족도 조사 결과

4. 결론 및 향후 과제

본 연구에서는 게임을 장르로 하여 기획자 혹은 그래픽 작업자가 레퍼런스 이미지 자료가 필요할 때, Text-to-Image를 사용해 편의성을 제공하는 서비스를 제안했다. 게임 일러스트를 재학습한 Stable Diffusion model을 유니티에 적용하여 유니티 씬 버전과 유니티 커스터마이징 두 버전의 서비스를 배포한 결과, 유니티 씬 버전이 90%의 더 높은 만족도를 보였다. 이미지 생성 서비스와 Prompt 추천 서비스의 만족도 결과는 각각 80%와 70%로 좋은 결과를 보였으나, 현재 택할 수 있는 장르가 3가지라는 점과 인물, 사물, 배경 중 어느 것에 해당하는지 Prompt를 추천 기능을 제공하지 않는 다는 점에서 서비스 보완의 필요성을 보였다.

향후 연구로는 현재 서비스에 적용된 Stable Diffusion의 여러 기능 중 Text-to-Image만 적용했기 때문에 그 외의 Image-to-Image, Inpaint 기능을 만족도가 더 높았던 유니티 씬 버전에 적용시켜 재서비스해볼 예정이다. 또한, 크롤링한 이미지 중 부적절한 이미지를 수동으로 삭제했는데 이 부분을 이미지 분류 모델을 사용해 자동으로 삭제할 수 있도록 고안해 볼 필요성이 있다. 마찬가지로 KR-WordRank의 경우 비지도 학습으로 진행된 사항으로 강화 학습 모델의 적용이 필요할 것으로 보인다.

참고 문헌

[1] Reed, S., Akata, Z., Tan, X., Logeswaran, L., Schiele, B. & Lee, H., “Generative Adversarial Text to Image Synthesis”, International Conference on Machine Learning, 1060-1069, 2016

[2] Gao, J., Li, P., Chen, Z. &Zhang, J., “A Survey on Deep Learning for Multimodal Data Fusion”, Neural Computation, 32, 829-864, 2020

[3] Xu, T., Zhang, P., Hunag, Q., Zhang, H., Gan, Z., Huang, X. &He, X., “AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks”, IEEE, 1316-1324, 2018

[4] “Midjourney”, Midjourney, accessed Nov 29, 2022, https://midjourney.com/

[5] Ramesh, A., Pavlov, M., Goh, G., Gray, S., … Sutskever, I., “Zero-Shot Text-to-Image Generation”, Proceeding of Machine Learning Research, 8821-8831,

2021
[6] “Stability.ai”, Blog-Stability.ai, accessed Nov 29, 2022, https://stability.ai/blog
[7] Mihalcea, R., Tarau, P., “TextRank: Bringing Order into Text”, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 404-411, 2004
[8] 김현중, 조성준, 강필성, “KR-WordRank: WordRank를 개선한 비지도학습 기반 한국어 단어 추출 방법”, 대한산업공학회, 18-33, 2014
[9] “국민트리”, 게임순위-국민트리, 2022년 11월 5일 접속, https://trees.gamemeca.com/gamerank/
[10] “lovit/KR-WordRank”, WordRank, accessed Nov 10, 2022, https://github.com/lovit/KR-WordRank.git
[11] “lstein-stable-diffusion”, lstein-stable-diffusion/scripts/dream.py, accessed Dec 3. 2022, https://github.com/afiaka87/lstein-stable-diffusion.git