|  |
| --- |
| **1. 주제: 만화 번역 프로그램**  **(가)분반, 1팀, 학번: 20241985, 이름: 조혁진** |

|  |  |
| --- | --- |
| **2. 요약**  많은 만화와 이미지가 일본어나 영어로 되어 있어 번역이 필요하지만, 기존 번역 방법들은 비효율적이고 자연스러운 구어체 번역을 제공하지 못한다. 이 문제를 해결하기 위해 이미지 설명 생성(image captioning)과 ChatGPT API를 활용한 번역을 적용하고, 번역된 텍스트가 그림을 가리지 않도록 번호를 붙여 만화 밑에 표시하는 방식을 제안한다. 이 프로그램은 Python으로 구현되며, Android에서 사용할 수 있도록 FastAPI를 통해 API로도 제공할 계획이다. | **3. 대표 그림**    그림 1. 예상 구현 결과 |

|  |
| --- |
| **4. 서론**  만화나 소위 ‘짤’을 찾아보면 대부분이 일본어나 영어로 되어 있다. 예를 들면 ‘pixiv’라는 사이트에서 대부분의 만화가 일본어 또는 영어로 되어 있다. 그러다 보니 ‘아카라이브’와 같은 사이트에서 번역가가 직접 번역한 작업물을 찾아서 보거나 ‘pixiv’내의 한국어판을 결재해서 보곤 했다. 하지만 이는 너무 비효율적인 방법이다. 자료를 찾느라 시간도 많이 들고, 비용도 많이 들고, 심지어 번역된 작업물이나 한국어판이 아예 없는 경우도 존재했다. 그래서 구글 이미지 번역이나 빅스비 비전을 이용해서 번역을 시도했지만 이 역시 문제가 존재했다. 둘 다 구글 번역을 이용해서 번역을 하는데 대화 상황임을 인지하지 못해 구어체가 아닌 문어체로 번역되거나 상황에 맞지 않게 존댓말이 나오기도 한다. 그리고 한글로 번역한 내용이 그림을 가리는 경우도 종종 있었다. 이러한 문제를 해결하기 위해서 현재 제공되고 있는 서비스의 뼈대를 비슷하게 가져가되 이미지를 설명하는 문장을 만들어내는, 즉 image captioning을 하는 신경망을 이용하여 상황에 대한 설명과 문장을 ChatGPT와 같은 LLM에 넣어 번역의 자연스러움과 정확도를 높여 문제를 해결한다. 번역된 문장이 그림을 가릴 경우 말풍선에 번호를 넣고 만화 밑에 공간을 생성하여 번호와 번역 내용을 넣어주는 방식으로 문제를 해결한다. 파이썬 프로그램으로 제작할 것이고 안드로이드에서도 사용할 수 있도록 FastAPI를 통해 API로도 제작할 것이다. |

|  |
| --- |
| **5. 본론**    만화 번역 프로그램은 크게 세 부분으로 나뉜다. 첫 번째는 이미지에서 글자를 추출하는 부분이고, 두 번째는 추출된 글자를 원하는 언어로 번역하는 부분, 세 번째는 번역된 결과를 이미지에 넣는 부분이다. 이미지로부터 글자를 추출하는 것은 OCR이라는 기술을 사용할 것이다. OCR이란 이미지 텍스트 추출을 통해 기계가 읽을 수 있는 형식으로 변환하는 기술을 말한다. 전처리, 문자 인식, 후처리 순으로 진행되고 최근에 딥러닝 기술이 발전함에 따라 CNN과 같은 신경망을 이용해서 텍스트 분류를 하고 LSTM과 같은 신경망을 통해서 문맥도 고려하며 글자를 후처리 한다. 하지만 이러한 신경망을 직접 구현하기에는 무리가 있고 Google Vision을 이용하거나 PaddleOCR이라는 multilingual OCR 오픈소스를 활용할 수 있다.  글자를 추출했으면 이를 원하는 언어로 번역해야 한다. 언어를 번역할 때 문장을 단어 단위로 끊고 번역할 언어의 그 단어와 뜻이 비슷한 단어를 찾아 넣는 방식으로 할 수 있다. 그런데 글에는 문맥이라는 것이 존재하고 문법 구조도 언어마다 다르기 때문에 알고리즘적으로 구현할 수 없다. 그래서 단어를 벡터화 해서 단어의 의미와 문맥을 수치화 해서 데이터로 만들고 딥러닝 신경망을 학습시켜 언어 모델을 구현하여 번역하면 자연스러운 번역 결과를 얻을 수 있다. 이런 방식으로 번역 부분을 해결할 수 있다. 하지만 이 역시 직접 구현하기에는 무리가 있고 ChatGPT API를 활용할 수 있다. 또한 더 자연스러운 번역을 위해서 image captioning을 하는 신경망을 추가할 것이다. PyTorch 또는 TensorFlow를 통해서 모델을 생성하고 Kaggle에서 image captioning에 사용되는 데이터를 얻어 학습시켜 구현할 수 있다. 마지막으로 번역된 결과를 이미지에 넣기 위해서 pillow라는 파이썬 라이브러리를 활용할 것이다. Google Vision이나 PaddleOCR을 사용할 경우 글자가 추출된 곳의 좌표 값을 알려준다. 그 값을 기반으로 기존 텍스트를 지우기 위해서 흐리게 처리해준 뒤 번역한 텍스트를 넣을 것이다. 그런데 그 텍스트의 길이가 기존보다 길어져 그림을 가리는 경우가 있다. 이러한 경우를 대비해서 만약 좌표 값에서 얻은 한계점을 넘는 경우, 우측 상단에서 시작해서 오른쪽에서 왼쪽으로, 위에서 아래로 1, 2, 3… 번호를 메기고 만화 밑에 공간을 추가하여 번호와 그에 해당하는 번역된 텍스트를 추가하는 방식으로 문제를 해결할 수 있다. Google Vision과 PaddleOCR을 테스트해서 성능이 좋은 것을 택하여 글자 추출 부분을 구현하고 image captioning을 담당하는 신경망은 TensorFlow의 MobileNetV3Small과 2-layer Transformer-decoder를 이용해서 모델 구축 후 학습시켜 사용하고, 만약 성능이 좋지 않을 경우, 다른 모델을 이용하거나 API를 찾아볼 것이다. 이미지 수정을 위해서 pillow 라이브러리 사용하여 구현하고 좌표 값에서 얻은 한계점을 넘을 경우 사진 밑에 텍스트를 추가하는 것은 OpenCV 라이브러리를 이용하거나 pillow 라이브러리를 사용하여 개발할 계획이다. 그리고 FastAPI를 통해 API 형태로도 개발할 계획이다. |

|  |
| --- |
| **6. 결론**  만화 번역 프로그램은 크게 세 부분으로 나뉜다.  **글자 추출**: 이미지에서 글자를 추출하기 위해 OCR 기술을 사용하며, Google Vision이나 PaddleOCR을 활용할 것이다. 신경망 기술을 통해 전처리, 문자 인식, 후처리를 진행하여 글자를 기계가 읽을 수 있는 형식으로 변환한다.  **번역**: 추출된 글자는 ChatGPT API를 사용해 원하는 언어로 번역된다. 자연스러운 번역을 위해 image captioning 신경망(MobileNetV3Small과 2-layer Transformer-decoder)을 이용해 학습하고, 성능이 좋지 않을 경우 다른 모델이나 API를 사용할 것이다.  **이미지에 번역된 텍스트 삽입**: Pillow 라이브러리를 사용해 원래 텍스트가 있던 곳을 흐리게 하고 번역된 텍스트를 추가한다. 텍스트 길이가 기존보다 길 경우, 우측 상단에서 오른쪽에서 왼쪽으로, 위에서 아래로 번호를 부여하고 사진 밑에 번호와 텍스트를 추가해 해결한다. OpenCV 또는 Pillow를 이용해 이 부분을 구현한다.  또한, FastAPI를 사용해 API로도 구현할 계획이다.  Google Vision과 PaddleOCR을 테스트하기, ChatGPT API 이용 방법 학습, TensorFlow로 image captioning 모델 구축 및 Kaggle에서 얻은 데이터로 학습시키기, pillow, OpenCV 라이브러리 학습, FastAPI 학습이 추후 학습할 내용이다. 배운 내용을 토대로 프로그램의 세 부분에 해당하는 3개의 모듈을 제작 및 테스트하고 이를 통합하고 API로도 구현하는 것이 추후 할 일이다. |

**7. 출처**

[1] PaddleOCR, github, <https://github.com/PaddlePaddle/PaddleOCR/blob/main/README_en.md>

[2] OCR 최신 동향, 2022\_02\_08, github pages, <https://yongwookha.github.io/MachineLearning/2022-02-08-current-ocrs>

[3] MRN for Incremental Multilingual Text Recognition, github, <https://github.com/simplify23/MRN/blob/main/README.md>

[4] OpenAI API reference, OpenAI Platform, <https://platform.openai.com/docs/api-reference/introduction>

[5] pillow 이미지에 텍스트 추가하기, tistory, <https://daco2020.tistory.com/832>

[6] OpenCV 이미지 테두리 만들기, VirusHeo, <https://virusheo.blogspot.com/2022/05/220528.html>

[7] 점프 투 FastAPI, 박응용, <https://wikidocs.net/book/8531>