LLM-Based Movie Poster Translation

ALISHA BRENHOLT, SAMANTHA KISSEL, YUJIN KI

CSE 895 LARGE LANGUAGE MODEL

Table of Contents

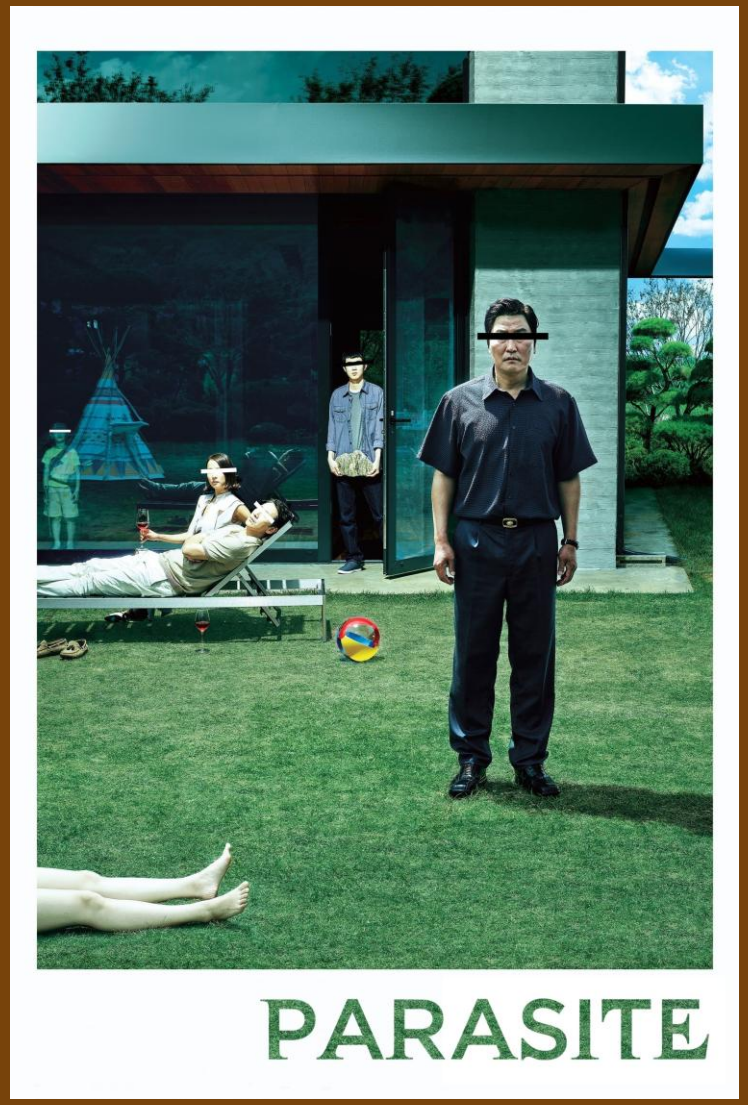
Introduction

WHAT IS THIS PROJECT?

Motivation

Why do we care?

- Transnational cinema has been an increasingly prevalent movie creation ideology
- The international movie market has steadily been providing more revenue year over year
- International movies allow for an international cultural exchange
- The way movies are written and made is being tailored to international audiences
- Translating movie posters can be expensive



What does this mean?

Previous Work – Machine Translation

- Machine translation is a well-studied task in the field of NLP with much historical background
- Approaches to MT have evolved over time with the popularization of transformer models
- We found that most state-of-the-art procedures for tasks related to ours used transformer-based language model architectures
- Transformer encoders are a common approach to encoding input text
- Some newer studies are using a combination of text and image attention in transformer-based LLMs, allowing the LLM to have a greater context when performing translation

Previous Work – Images

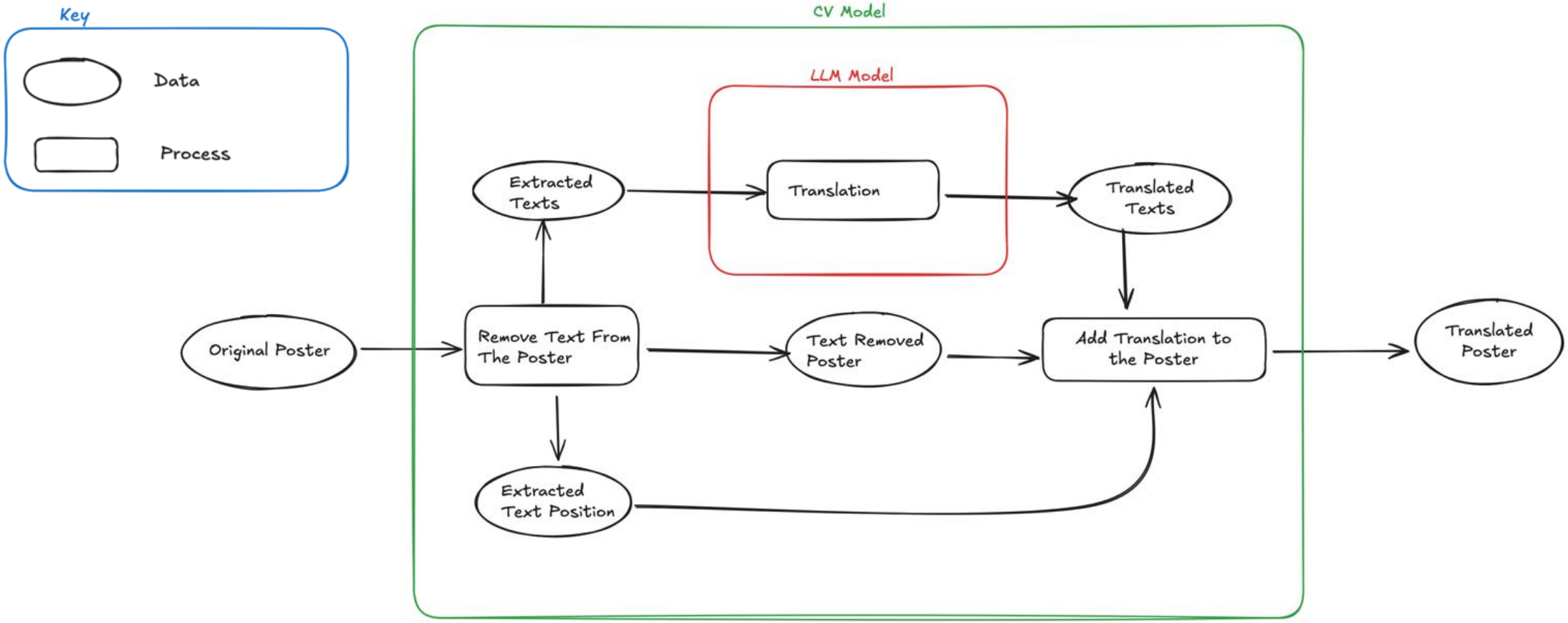
Extracting Text

- For extracting text from images, optical character recognition (OCR) is used
- There are several popular models for OCR including Pytesseract, EasyOCR, KerasOCR, and more
- These models are typically built on neural network deep learning architectures such as CNNs and RNNs

Image Editing

- Today, there are some widely-known text-to-image generation models, but image *editing*, where you input your own image and prompt the model to edit something, is less common
- Generating text is a task that imaging models still struggle with
- Some newer studies are starting to fine-tune and use diffusion models to print text onto images

System Overview

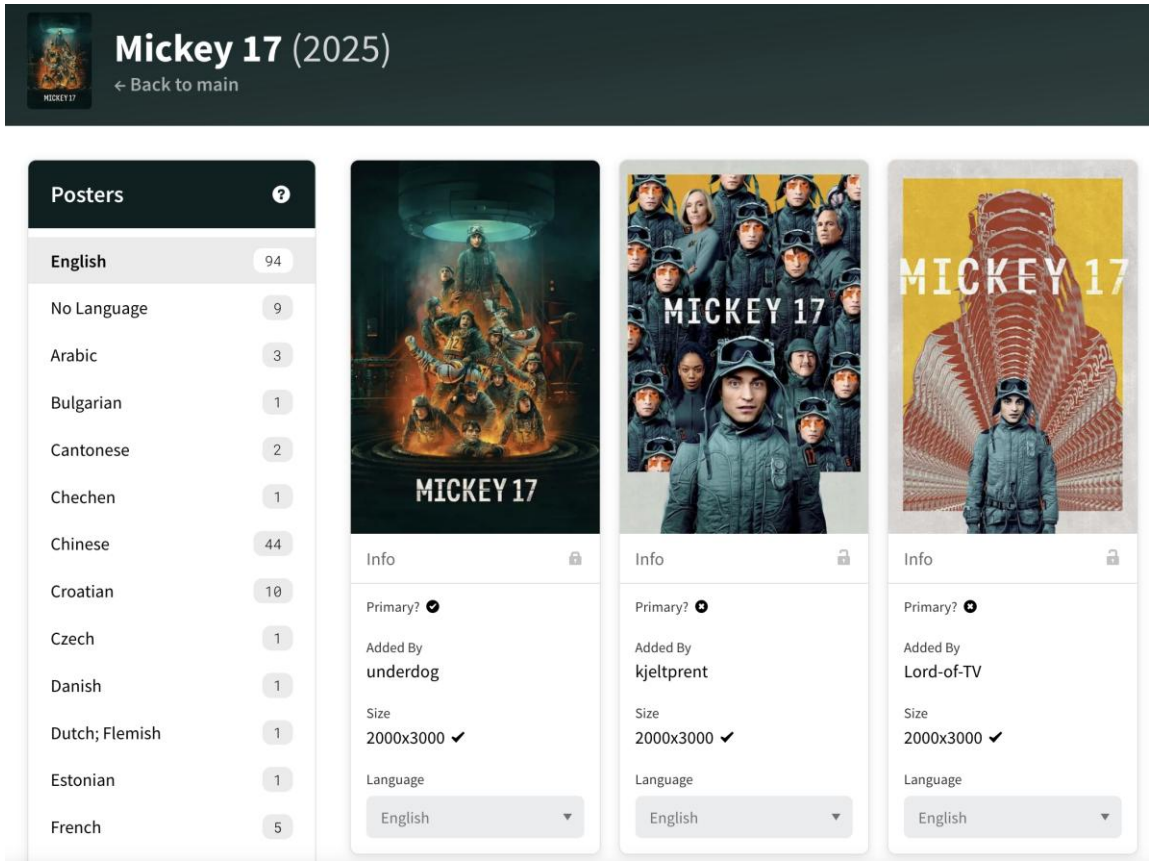


Dataset

TMDb

The most popular Movie database

- Over 1 million movies
- Over 190 thousand TV shows
- Over 5 million TV episodes

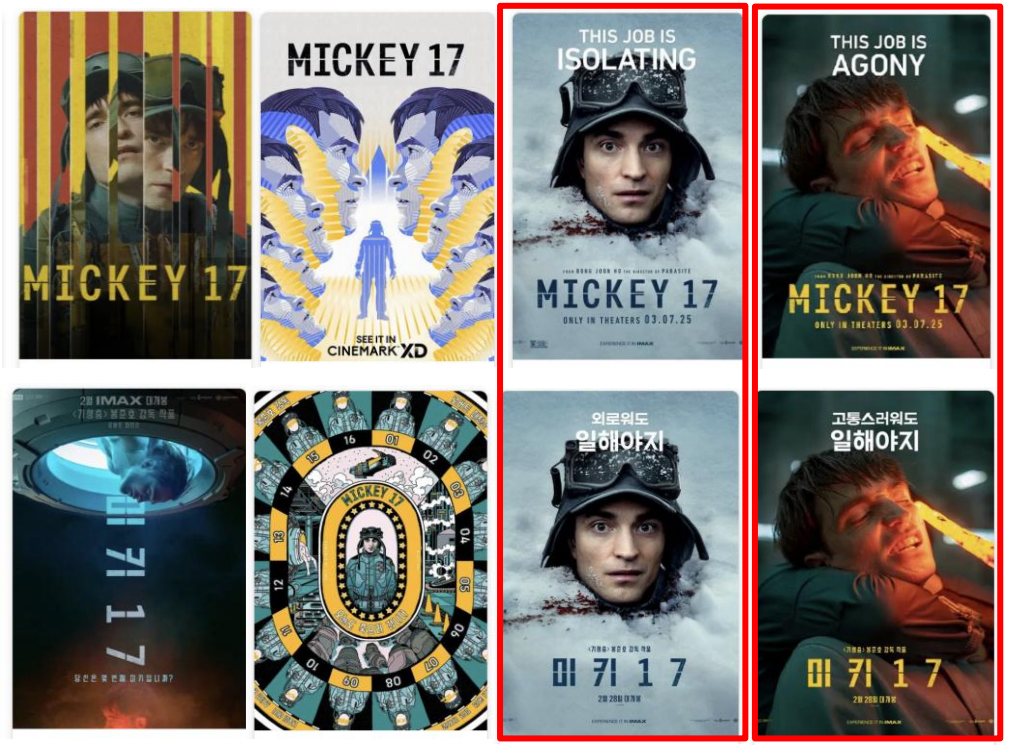


Dataset

1. **Collect 1200 En-Ko posters** of movies that has both language posters from TMDb using API
2. Pick only posters that are actually matching pairs => **Final 87 posters**
3. Manually extract Korean ground truth text data from Korean posters.

English posters

Korean posters



Mickey1_ko_GT.txt

외로워도, 일해야지,
<기생충> 봉준호 감독 작품,
미키 17, 2월 28일 대개봉

Mickey2_ko_GT.txt

고통스러워도, 일해야지,
<기생충> 봉준호 감독 작품,
미키 17, 2월 28일 대개봉

OCR Text Extraction

Easy OCR

EasyOCR struggles to catch artistic texts in movie posters

Pytesseract

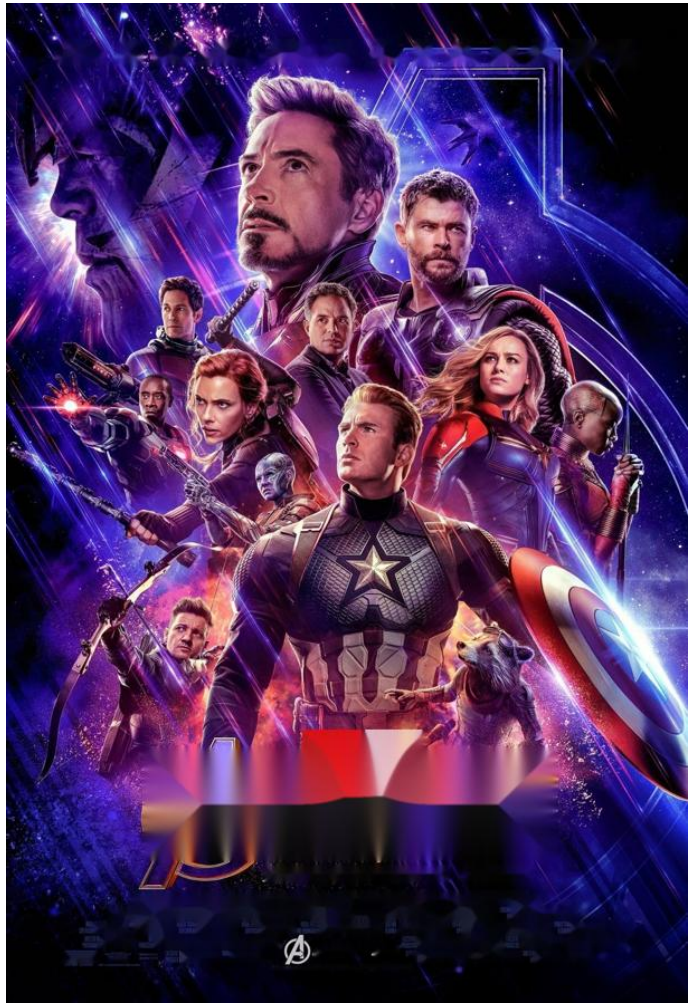
Pytesseract needs fine tuning,
And even then, it struggles to acknowledge only texts from small little detailed pixels of movie posters

KerasOCR Text Extraction

KerasOCR

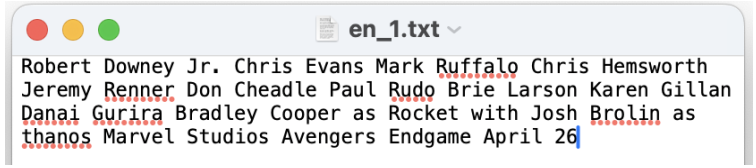
- Good at accurately detecting only correct texts
- Almost no mistakes misreading small pixels to texts.
- Give bounding boxes' location and width/height

KerasOCR Text Extraction



Output

1. Extracted English texts



2. Each text's bounding box position
3. Bounding boxes' width and height
4. Blurred original poster

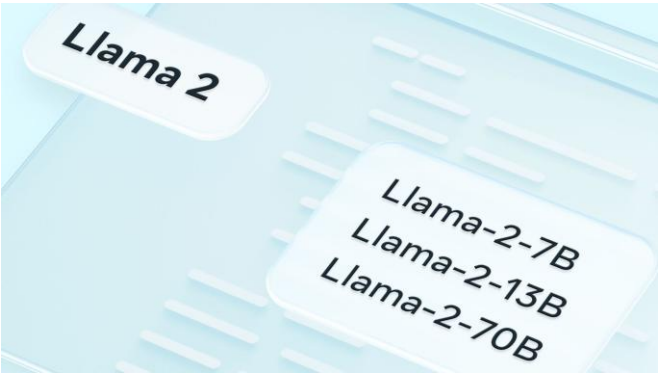
LLM Translation - Models

- Different models were tested
  - Llama2-13b-Language-translate
  - Facebook/mbart-Large-50-many-to-many
  - KETI-AIR/long-ke-15-base-translation
  - Mbart-mmt-mid1-en-ko
  - Mistral-7b-instruct
  - GPT-4o-mini
- GPT-4o-mini cost $5 to run for our project
- Models specialized in instruction or translation

MISTRAL AI_

OpenAI

Meta



LLM translation – Prompting Strategies

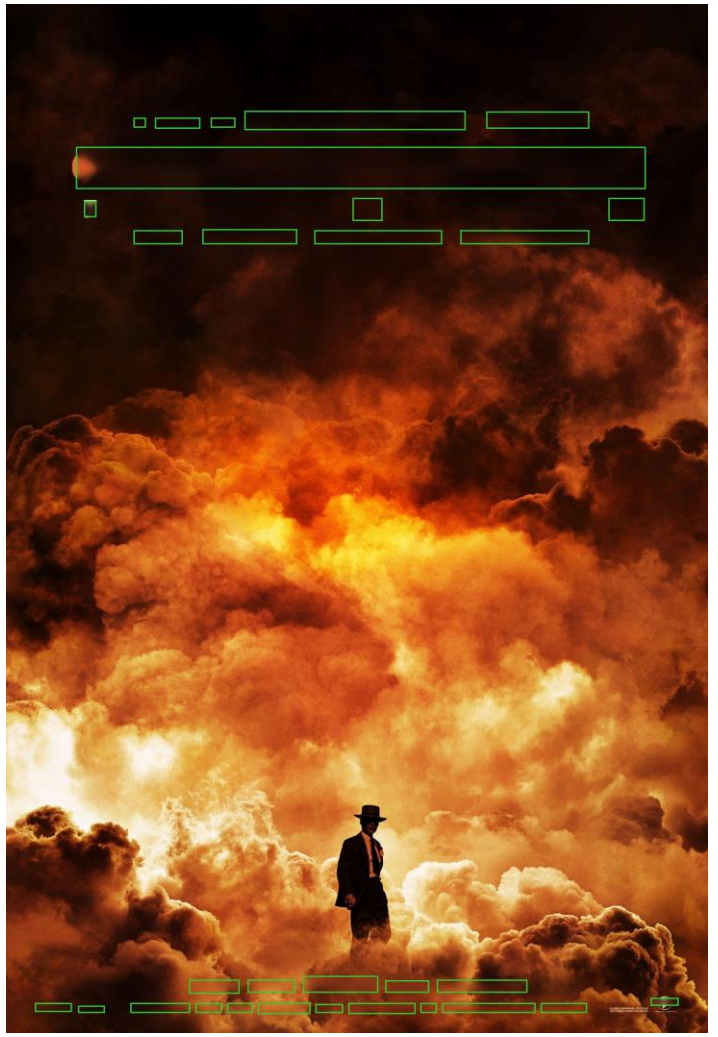
- Several iterations of prompts
  - Raw results – direct output from LLM
    - Often had incorrect tokens or English, or parts of the prompt
  - Simple output filtering – Remove English and prompt if it is repeated
    - Helped results, but sometimes incorrect Korean still existed
  - Single shot with filtering – Filter but also include an example with prompt
    - Increased performance even more, ensured single word Korean outputs
  - Previous techniques with second LLM prompt – input into another prompt to ensure proper results
    - This hurt performance
  - Include sentence to give contextual meaning
    - Including the sentence in the prompting helped ensure the proper form of words was used

Text Insertion Algorithm

- After getting the translation, the pipeline moves onto the text insertion phase

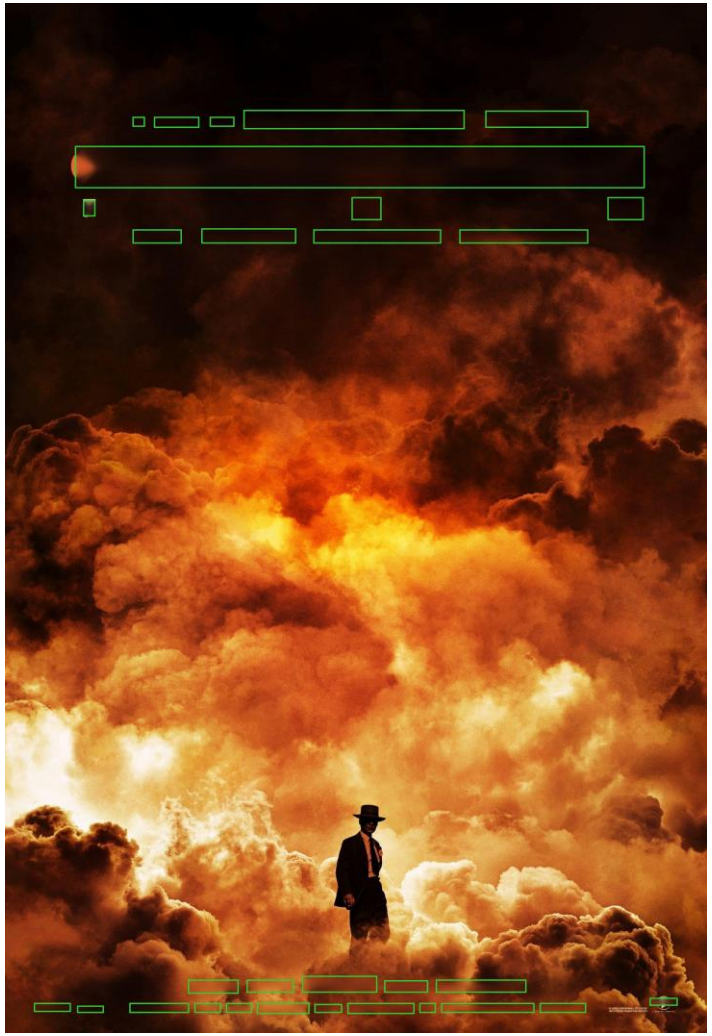
- Input into the text insertion algorithm is:
  1. The poster with English text removed
  2. Korean characters to be inserted
  3. The location of the bounding boxes for the placement of the text

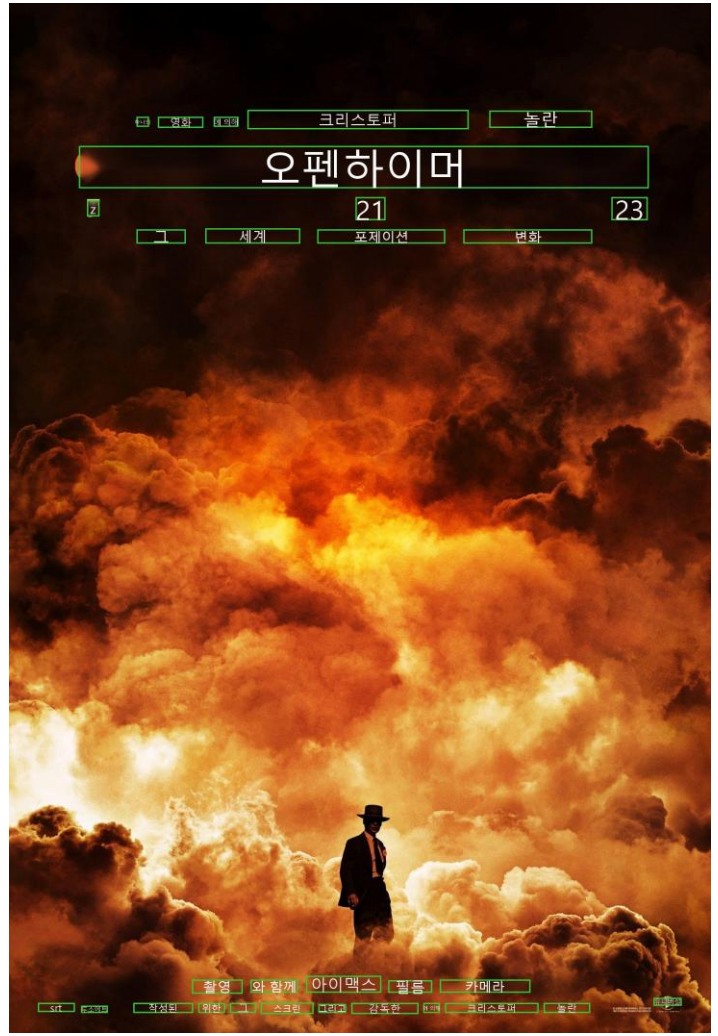
```
Translated Text:
 {'oppenheimer': {'en_2.jpg': [('놀란', array([[1394.5312 ,  199.21873],
       [1754.8828 ,  199.21873],
       [1754.8828 ,  266.60153],
       [1394.5312 ,  266.60153]], dtype=float32)), ('크리스토퍼', array([[ 565.4297 ,  202.14844],
       [1327.1484 ,  202.14844],
       [1327.1484 ,  269.53125],
       [ 565.4297 ,  269.53125]], dtype=float32)), ('영화', array([[319.33594, 222.65623],
       [457.03125, 222.65623],
       [457.03125, 266.60153],
       [319.33594, 266.60153]], dtype=float32)), ('에 의해', array([[468.75   , 222.65625],
       [547.85156, 222.65625],
       [547.85156, 266.60156],
```



Text Insertion – Bounding Boxes

First, the font size for the Korean text needs to be determined so that it fits into the original bounding box on the poster





Text Insertion – Color

- We also need to pick a font color so that the text will show up on the photo regardless of the background color
- We sample the color of the background pixel in the middle of the bounding box and calculate its brightness using a standard formula:

Brightness = 0.2126R + 0.7152G + 0.0722B

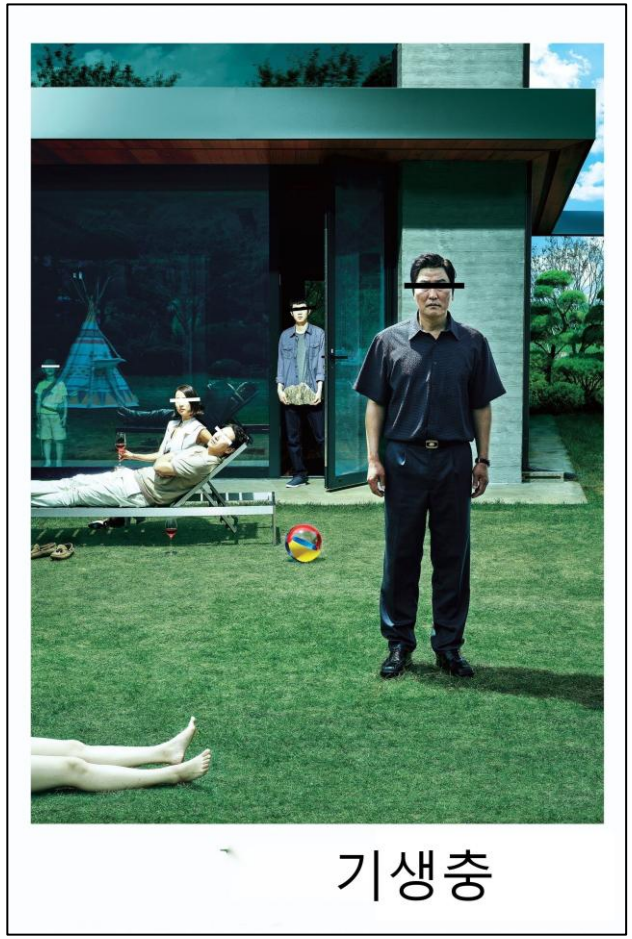
Where R, G, and B are the RGB values of the pixel color

- If the brightness value is > 128 --> the background is "light"; put black text in the box
- If the brightness value is <= 128 --> the background is "dark"; put white text in the box

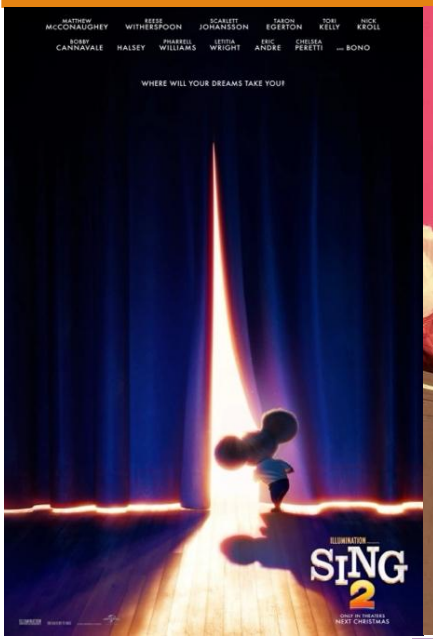
Text Insertion – Final

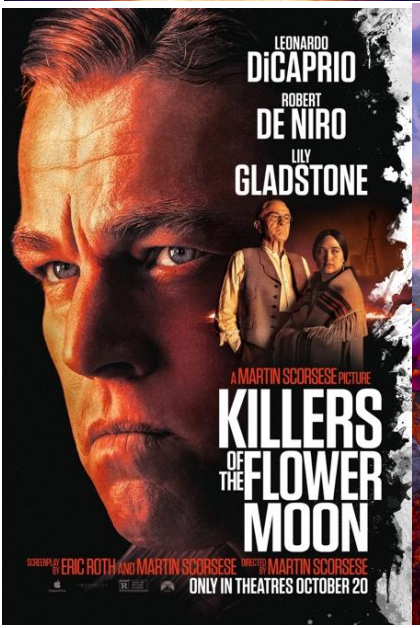
- After selecting font size and color, we now have everything to place the text back onto the poster
- We draw the Korean text on the movie poster image using the Python PIL library

Final Poster



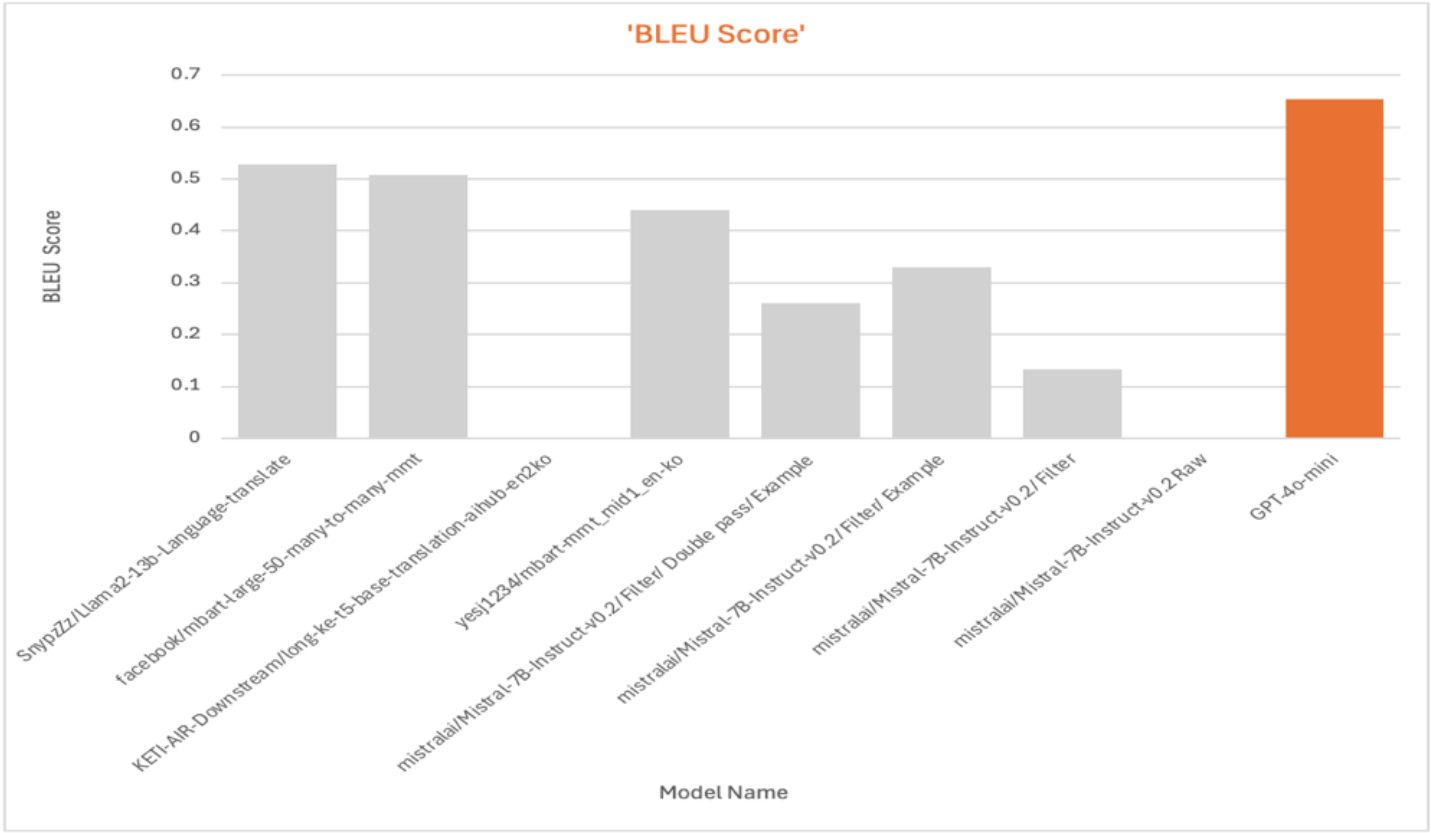
English poster | Our model's result | Korean poster

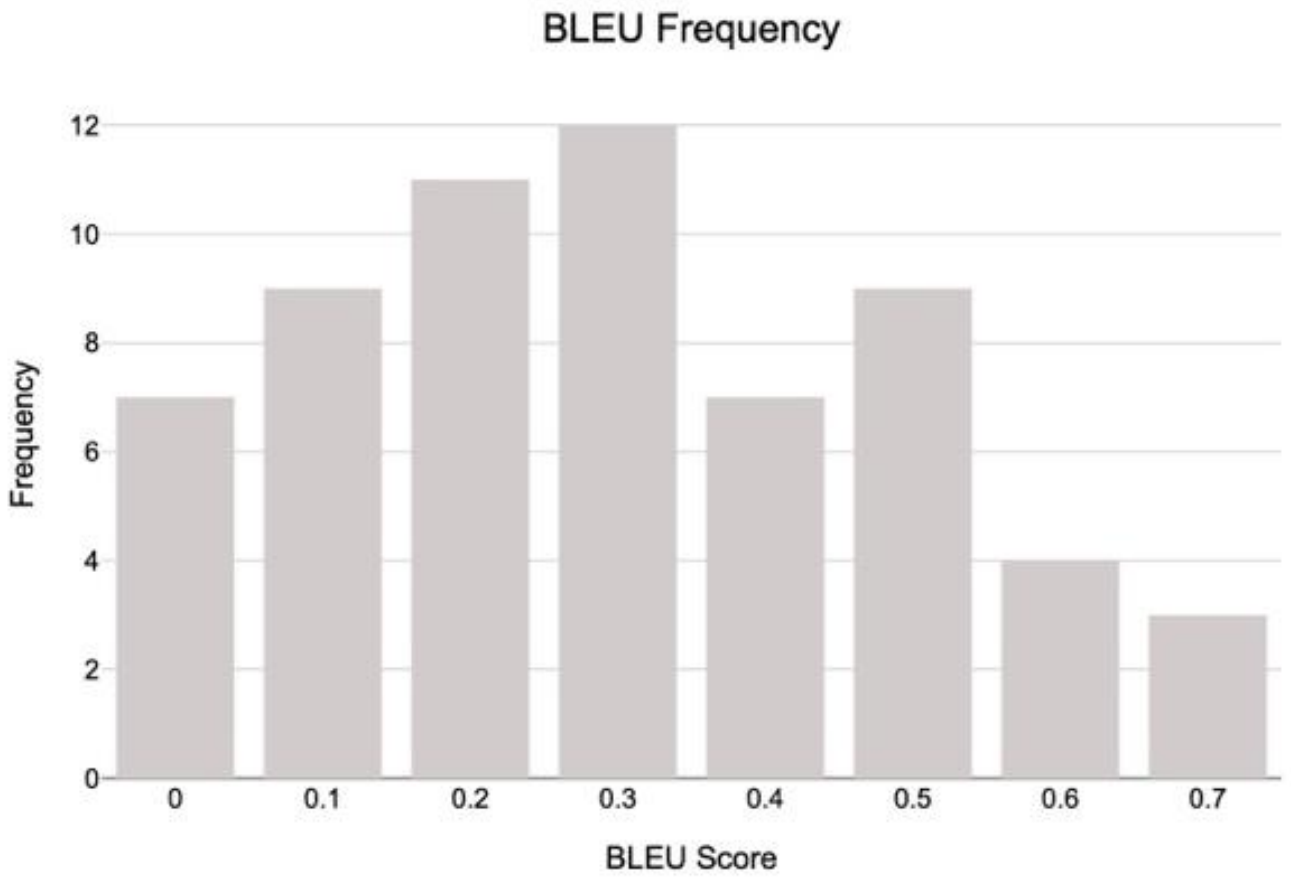




Results
BLEU Scores Model Evaluation

- The independent BLEU scores are shown
- The orange indicates the model with the highest score
- Demo sentences were used to evaluate each model
  - Each sentence was related to movies in some way
  - Some sentences had errors to evaluate how well the model can adjust for incorrect text extraction





BLEU Score - Extracted Frequency

- BLEU score frequency when extracting text from movies and translating it. The reference sentences are the hand annotated ground truth posters
- Chart showing the distribution of BLEU scores for movies
- The most common result was a BLEU score of .3 to .4

Results
Image Similarity

- To evaluate our final output, we compare our generated images to the ground truth Korean posters using image similarity metrics

3 different similarity scores

1. VGG16 Cosine Similarity: uses the pretrained VGG16 model to create embeddings for the images and compares the embeddings using cosine similarity
2. ResNet50 Cosine Similarity: same process as VGG16, but with the pretrained ResNet50 model
3. Structural Similarity Index Measure (SSIM): compares structural information of pixel values between two images

Score table for each metric

	VGG16	ResNet50	SSIM
Best	0.947	0.989	0.967
Average	0.742	0.901	0.53

Results-Human Evaluation

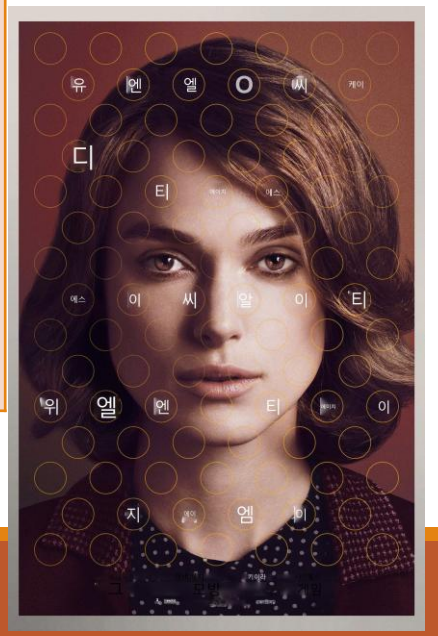
Visually Bad

Meaning Delivery Good

Visually Bad & Delivery Good	Visually Good & Delivery Good
11.5%	56.3%
10/87	49/87
Visually Bad & Delivery Bad	**Visually Good & Delivery Bad**
5.7%	26.4%
5/87	23/87

Meaning Delivery Bad

Visually Good

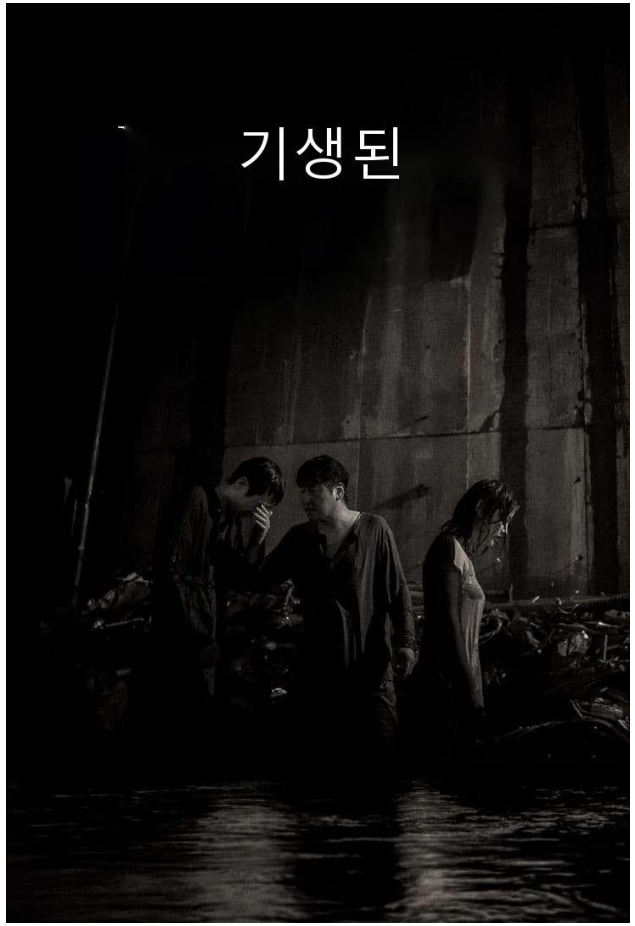


Conclusion

- As our world becomes increasingly globalized, machine translation is an important task to keep us connected
- International movies are reaching new audiences, but not all film production companies can afford the cost to advertise their movies abroad
- Using LLMs, our results show that the process of translating movie posters into new languages can be automated, saving filmmakers time and money on art production and advertising
- Many of our generated images score high on both translation and image similarity

Future Challenges

The translation models can continue to be fine-tuned and improved



Incorrect contextual translation

The OCR model occasionally picks up undesired elements, such as symbols that are part of an image on the poster

Error in OCR

Future Challenges

- Our text placement algorithm can be improved, especially in regards to artistic quality
  - Text color, font style, improving the blurring of text, etc.
- An ideal solution would be using a diffusion model to dynamically generate a font that matches the original English text
  - However, artistically generating text is still a common struggle for many diffusion models

Font style and color is lost in translation

Thank You !