Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Midterm Milestone Report
on
“GO GO Nihongo”

[Code No: COMP 488]

Submitted by:

Abiral Adhikari (Roll No.02-CE)
Jayash Bhattarai (Roll No.08-CE)
Prashant Manandhar (Roll No.30-CE)
Hridaya Pradhan (Roll No.38-CE)

Submitted to:

Prof. Dr. Bal Krishna Bal

Department of Computer Science and Engineering

Submission Date:2024-11-13

Chapter 1 Introduction

Although, Japanese language is an East Asian language spoken predominantly in Japan, it has huge impact in international context due to Japanese media like manga, anime etc. blowing up as the global phenomenon in recent times. So, with our project, we aim to provide translation of Japanese texts from different sources like comments, manga, posters etc. to English language. We will be focusing on translating text by extracting it from images. English language is chosen to be destination language for translation being the global language of communication, from which the translated text can be easily converted to other languages.

As the huge number of the raws (manga with the Japanese dialogue and text) are released per day, the manual translation for all those raws is time consuming due to lack of translators. In order to address this problem, we plan to promptly provide initial machine translation of the raws. In our project, we plan to extract the Japanese text from images using third party library like EasyOCR, Manga OCR. The model for translation based on encoder-decoder architecture will be trained using “Verah/tatoeba_dedupe_en-jp_2024-March-01” dataset available in Hugging Face. After translation of the text by our model, the translated text will overlayed upon the original image. With this project we aim to make the consumption of Japanese media convenient especially the manga which is popular among young adults.

Chapter 2 Proposed Method and Experiments

2.1 Preprocessing

For the preprocessing phase, we have mainly worked on removing emoji, special words and keeping only the characters like a-z, A-Z, Kanji, Katakana, Hiragana. English might contain accent like è but Japanese doesn't have any accent alphabets, so we have removed the accent for the uniformity. This was done by defining a function to ascii for Japanese and English. The special characters were also removed, and space was created between the word and punctuation. For example, "hello #@...123world." was then processed as 'hello 123world'. The all the Zen character fonts were changed to Han character fonts for uniformity. The text is converted to ASCII and start, and end token are established to tokenize the texts. These texts are then stored in separate lists which the function returns as the respective lists. In the tokenizing phase, we used Janome Tokenizer, which is a Lattice based tokenizer to generate tokens for Japanese language. For English language, we have used Keras Tokenizer as it was sufficient. On proceeding to creating clean dataset, we have tokenized English and Japanese sentences. The sentence length was calculated for padding and splits the data for further training sets. These were mapped back to words for verification at the end. These help to clean and structure the dataset for training models.

2.2 Model Development and Training

For the task of neural machine translation, we considered two options: seq2seq model or transformer model, which are both based on encoder-decoder architecture using attention mechanism. We started the experimentation with seq2seq model, for this we proposed a LSTM based encoder and decoder, which is better at handling long term dependencies than RNN layers. In encoder and decoder architecture, the encoder component converts the input Japanese text to context vector (embeddings). The output state of encoder is used as the initial state of the decoder. Attention Mechanism was used in model to focus on specific parts of the input sequence when generating each word in the output. We opted for the Bahdanau (Additive) Attention mechanism which more effective for language pair of (Japanese - English) having significant structural differences than Luong (Multiplicative) Attention mechanism. The Attention Mechanism calculates attention weights which are used to obtain context vector (weighted sum of all encoders hidden state). Thus obtained, context vector is combined with the decoder's current input to predict the next word in the output sequence (English). As, we are using a larger dataset with 100k Japanese - English sentences, the model training in Kaggle and Colab, required huge hardware resources. We trained the model up to 7-10 epoch which resulted in final loss of 0.28 and BLEU score of 0.48.

2.3 Text Extraction and embedding using OCR

For the text extraction from the manga, we have experiments with the different python package like EasyOCR, Manga OCR and tesseract. EasyOCR was used to find the coordinate for the bounding box of the characters it tries to recognize. The obtained coordinates of the bounding boxes were checked if completely overlapped rectangle was present or not. If that condition is satisfied then, that rectangle was discarded to avoid repeated processing of the overlapped area. The coordinates of the bounding boxes were used to crop the image so that only the part of the image, that contains the text, was extracted. This avoids the hassle of searching for the text in unwanted area of manga and one of the requirements for the accurate extraction of the Japanese text using Manga OCR.

Chapter 3 Problem Faced

3.1 Preprocessing

1. **Normalization**

   In Japanese language, there are two types of fonts, namely 全(Zen) and 半(Han). For example, the English equivalent for this would be changing Ｂ Ｏ Ｘ(Zen) to BOX (Han). Translating two different fonts proved to be more complex so we decided to change all the Zen fonts to the Han fonts as Han fonts are generally compatible on other platforms and are similar to the English fonts. It also ensures to hold readability in between the texts because of their similarity.

2. **Language Structure**

   Japanese language uses three scripts, hiragana, katakana and kanji and has large vocabulary in the case of Kanji characters which often has multiple meanings for one character. Tokenizing the Japanese text is complex than English because there exist no spaces between words and thus it is difficult to separate the meaningful words to have the respective English translations. Japanese grammar is quite different to that of English including subject-object-verb structure, omissions of subject in the sentences and honorifics. These require the model to learn the complex grammar structure without extensive training data.

3.2 Model Development

1. **Limited resource for training**

   The model was used 100k Japanese-English pair sentences which resulted in longer training times and high GPU resource requirements.

2. **Lacking Context Awareness due to Long Input Sequence:**

The data had sentence of variable length, which resulted in the problem of context awareness and accurate capturing of distant dependencies especially due to long sequence inputs.

3. **Improper Handling of Out-of-Vocabulary Words:**

   The embeddings for both languages were generated using the vocabulary from the dataset, it was challenging for the model to handle out of vocabulary word resulting in translation quality suffering from vocabulary bottle neck.

3.3 OCR Extraction

1. **Incapability of detection of the Japanese text using Tesseract and EasyOCR**

   Tesseract and EasyOCR cannot fully detect the text from the manga, as the text is in Japanese language, which is often written vertically in Mangas, a layout uncommon in other the document type. Even though, EasyOCR is similarly unable to recognize Japanese text in manga, it gives accurate bounding box of the coordinate of the characters which helps to solve the problem faced by Manga OCR. This coordinate was later used to crop the images

2. **Inaccurate detection of the text when whole manga page is processed**

   The Manga OCR cannot handle the processing when whole image was processed. Among the multiple sentence present in the whole image, the Manga OCR was only able to detect the few characters from random place. While experimenting with various images, working with the smaller images was found to be the best. So instead of the processing whole image, the cropped section, which only contain the text, was processed.

Chapter 4 Plan of upcoming activities

4.1 Preprocessing - (Jayash Bhattarai)

The further experiment can be done by using pretrained tokenizer model, especially for the Japanese languages as the simple tokenizer cannot handle it properly due to the lack of space.

4.2 Model Development and Training - (Abiral Adhikari & Prashant Manandhar):

Even though the initial model has been trained, there is the need to optimize the hyperparameters to increase the model capacity. The attention mechanism used can be revised to used multi-head attention for better performance. Also, restructuring of the model using bottleneck layer between encoder and decoder to reduce dimensionality, is topic of interest that can be explored. The evaluation of the trained model with BLEU metrics (Bilingual Evaluation Understudy) is to need to be done for same dataset and other dataset for checking generalization.

4.3 OCR - (Hridaya Pradhan)

The texts are extracted from the cropped images containing the text from the original target image. Those text will be extracted using Manga OCR which will then be translated using our trained model. The outputted translated text from the model, will be replaced into the image. This will be done by erasing/hiding the previous text. For that purpose, the white rectangle is drawn on the coordinate of the bounding boxes boxed, obtained from the EasyOCR. This will hide the original Japanese text. The translated text will be written inside that same bounding box.