ID 217

Image Captioning in Tamil Language with Merge Architecture

G Rajalingam# and WU Wickramaarachchi

Department of Computing, Rajarata University of Sri Lanka, Mihintale, Sri Lanka

#gobigarajalingam@gmail.com

Abstract—Image Captioning is the process of describing the content of an image using a natural language. This task that involves computer vision and natural language processing has been attempted on the English language with enormous success, owing to the presence of massive image-caption paired corpora as Flickr and Microsoft Common Objects in Context (MS-COCO). However, such developments in this arena have been a novelty for non-English languages with the exception of a few such as Chinese, Turkish, German and Arabic. In the case of Tamil language, this premise has been barely touched upon, due to the lack of a large, paired corpus. In this work, a paired corpus inspired from Flickr30K dataset has been created in Tamil language for the image captioning purpose. Along with it, this paper includes the experiments with an image captioning model, using a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architecture; specifically the Merge model for Tamil language caption generation. This methodology incorporates the image vectors in a layer following the LSTM layer. The results of the research have proven satisfactory in the evaluation with a Bilingual Evaluation Understudy (BLEU) score of 0.37, and this indicates further development with the presence of a more refined and improved dataset.

Keywords: ***Tamil caption generation, convolutional neural network, long short-term memory, natural language processing***

I. INTRODUCTION

Image Captioning refers to describing an image on what is portrayed in it including the entities present and the actions performed by identifying the objects, their attributes, and their relationships in the image. This task involves two of the major fields in Artificial Intelligence: Computer Vision and Natural Language Processing. For an input of an image, an output of syntactically and semantically correct and meaningful sentence is expected as a caption in a typical image captioning task. Image captioning has several applications in the natural language processing domain such as recommendations in editing applications, in virtual assistants, usage in social media and more. This is incredibly useful in aiding the visually impaired to provide them with an understanding of their surroundings. Image Captioning is a promising premise with various applications in its wake involving Natural Language Processing. Several remarkable developments have been made in the past in Image Captioning in the English language, except for a very few non-English languages as Chinese (Zhang C et al. 2018), Turkish (Yılmaz BD et al., 2019) and Arabic (Al-Muzaini HA, Tasniem N and Benhidour H, 2018) using deep-learning in handling the complexities for semantically complex languages.

As the number of large datasets increase, many deep learning-based techniques have come to hold great promise in their performance and accuracy in image captioning tasks. Many of these technologies addressed for English language (Chen X and Zitnick CL, 2015; Karpathy A and Fei-Fei L, 2015; Vinyals O et al., 2015; Xu K et al., 2015), along with the non-English ones, have made use of the Long Short-Term Memory (LSTM) networks, a special type of Recurrent Neural Networks (RNN). The technique utilized in this study involves a combination of Convolutional Neural Network (CNN) and LSTM architecture. Most of these previous works has used CNN as an image encoder by first pre-training it for image classification and using the last hidden layer as an input to the LSTM decoder that generates captions.

Since the Flickr30K (Plummer B et al., 2015) and Microsoft Common Objects in Context (MS-COCO) (Lin T et al., 2014) corpora are addressed for English with English caption dataset, several non-English languages have attempted image captioning by translating the dataset to the corresponding languages. Among such works, the ones on Arabic, Turkish and Chinese stand remarkable for their technique used in creating the image captioning dataset in their respective languages. Al-muzaini HA, Tasniem N and Benhidour H (2018) have created an Arabic version of a combination of Flickr and MS-COCO datasets and have used a deep LSTM-CNN model on the dataset with 'Merge' architecture with promising results. In this work, the authors have surmised that the results could be improved with much larger dataset, with the implication that Arabic is a morphologically complex language compared to English. Considering the similarity in the complexity of the language and the promising performance of Merge architecture in Arabic, this is chosen to be carried out for Tamil as well.

As for Image Captioning in Turkish language done by Yılmaz BD et al. (2019), their methodology has involved an encoder-decoder model inspired from the work of Vinyals O et al. (2015) consisting respectively CNN and RNN. They have defined the CNN portion of the model to extract the features of the image dataset and the RNN part to generate the Turkish captions. To build the Turkish dataset, the authors have utilized machine translation on MS-COCO dataset. This approach is fairly like the work on Arabic except for the difference on the architecture. Zhang C et al. (2018) have described a Recurrent Attention LSTM (RAL) model for the image Chinese Caption generation. This model has utilized Inception-v4, a CNN model, to extract image features and the RAL model mechanism determines feature weights. In these works, the models used are based on the CNN-LSTM architecture to extract features and generate captions with a few variations. This often-used CNN-LSTM architecture, also referred to as Encoder-Decoder, was first proposed by Vinyals O et al. (2015), based on a CNN acting as an encoder, which is followed by an RNN which generates the caption for English, thus becoming the Decoder. The non-English works stated above utilize machine translation and human effort to build and refine their dataset and then proceed to apply the deep learning model to the dataset. Due to its consistent performance with non-English languages, this architecture which is referred to as 'Neural Image Caption' (NIC) generation is used in the research study for a comparative analysis against the Merge architecture in its performance with Tamil dataset.

A few other non-English works have compensated for the lack of large datasets in their languages by utilizing techniques as Unpaired Image Captioning by Language Pivoting and Image Captioning using Multilingual Data, which enables them to make use of English datasets to suit their requirements. However, these techniques often involve parallel corpora in large scale which are unaffordable resources for this project at this stage. Gu J et al. (2018) have attempted a method of capturing the characteristics of an image captioning component from the source language and align it to the target language using another source-target parallel corpus. The proposed framework is composed of an encoder-decoder model that can describe images in the pivot language and another encoder-decoder model (Neural Machine Translation model) to translate sentence from pivot language to target language. In this work, the authors have assumed of Chinese as a resource-rich language and English to be the resource-scarce, target language, wherein Chinese is used as the Pivot language and the results have outperformed the baseline methods on MS-COCO and Flickr30K databases. This involves the use of two different datasets to train both the image caption generation model and the machine translation model along with a Pivot-Target parallel corpus. Due to the heavy requirement of large datasets for the two models involved, this method is unsuitable for the scope this research aims for.

Beyond the use of pivot language technique, Jaffe A (2017) has proposed the use of a training corpus composed of both German and English captions to generate image captions in German, while ignoring the English output during evaluation. In this work, the German caption dataset has been created not necessarily as the direct translation of its English counterpart. Mostly, the German captions have been manually created to suit the image rather than to be a translation of the

English dataset. In the case of Tamil, this implies the necessity of manually curated set of datasets.

II. METHODOLOGY

As aforementioned on the existing methodologies for image captioning for English and non-English languages, the method analysed by H. A. Al-muzaini, N. Tasniem and H. Benhidour (2018) in the creation of an Arabic version of a combination of Flickr and MS COCO corpus and the usage of a deep Long Short-Term Memory Network and Convolutional Neural Network (LSTM-CNN) model on the dataset with 'Merge' architecture has been chosen as the high-level methodology for this study as well.

Dataset Pre-processing

Figure 1. Flickr30K Sample Image

Flickr30K dataset contains 31,783 images and each comes with five English sentences, forming around 150,000 sentences which have been translated to Tamil, the target language. Figure 1 is a sample image from the Flickr30K image dataset. Its paired text corpus in English have been translated to Tamil respectively without losing the core meaning to be used in the training.

i. A man in a blue baseball cap and green waders' fumbles with a fishing net in a blue boat docked beside a pier: ஒரு நீல பேஸ்பால் தொப்பியில் ஒரு மனிதன் ஒரு நீலப் படகில் ஒரு மீன்பிடி வலையுடன் தடுமாறினார்.

ii. A bright blue fishing boat and fisher at dock preparing nets: பிரகாசமான நீல மீன்பிடி படகு மற்றும் கப்பல்துறையில் மீனவர் வலைகள் தயாரித்தல்.

iii. A man in a small boat readies his net for the day ahead: ஒரு சிறிய படகில் உள்ள ஒரு மனிதன் தனது வலையை அடுத்த நாளுக்குத் தயார் செய்கிறான்.

iv. A lone fisher is on his boat checking his net: ஒரு தனி மீனவர் தனது படகில் தனது வலையை சரிபார்க்கிறார்.

v. Man in blue boat holding a net: நீலப் படகில் வலையை வைத்திருக்கும் மனிதன்.

Machine translation using Google Translator was used to translate the text corpus to Tamil and owing to the inaccuracy in the translations, they were reviewed and cleaned as required. The translated sentences were reviewed by native Tamil speakers to rectify the issues and discrepancies in the text by removing redundant words, untranslated English words, meaningless characters and rephrasing the text to make it more meaningful by clearly explaining the entities and their actions. Majority of the efforts were invested in translating the words which were unable to be translated by the Google Translator during the bulk translation process. The translated sentences were stored in a text file with UTF-8 encoding and the pre-processing process was conducted. The dataset was split into two for training and validation purposes in the 75:25 ratio, respectively. This resulted in 23837 images in the Training dataset and 7946 images in the validation set.

Model

The difference in Neural Image Captioning (NIC) discussed in Vinyals O (2015) in Figure 2 and the Merge model analysed by Tanti M, Gatt A and Cammileri KP (2017; 2018) as in Figure 3 is based on the variations in performance when the feeding of image dataset to the neural network, either by directly incorporating it in RNN or in a layer following RNN (Merge). Although Merge and NIC architecture differ with regards to where the image is inserted, Merge architecture has been stated to make better use of the RNN memory and they require less regularization than the others.

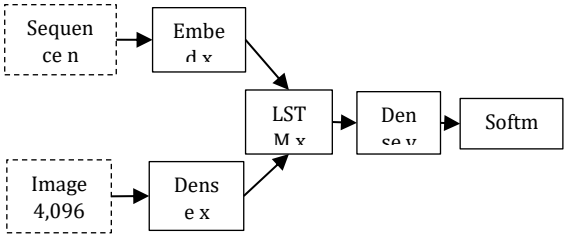

Figure 2. Flow of Neural Image Captioning Architecture (Vanilla CNN-LSTM).

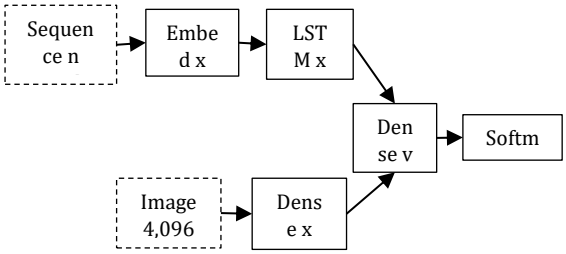

Figure 3. Flow of Merge Architecture used in Arabic and Turkish Image Captioning.

Rather than combining image features together with linguistic features from within the RNN, merge architectures delay their combination until after the caption prefix has been vectorized. This keeps the image out of the LSTM and would be capable of training the part of the neural network that handles images and the part that handles language separately. To be succinct, the RNN is not exposed to the image vector directly at all. Instead, it opts to introduce the image vector into the language model after the prefix has been encoded by the RNN to the entirety. To develop the Deep learning model in Merge architecture, the datasets (image and text) were loaded, and the vectorization of text was done with Keras Tokenizer class. Then the pretrained model and the sequence processor (a word embedding layer handling the text input with LSTM followed by it), result in a fixed length vector which are merged and processed by a Dense Layer. Then the model is fit to the dataset and is evaluated. The architecture of the of the "Merge" approach in the model is shown in Figure 2. An experiment was conducted with the existing architecture with and without the inclusion of mapping of input layer to the 300-d embedding vectors for Tamil from fastText (Bojanowski P et al., 2016). As displayed in figure 3, after the FastText 300-d Tamil embedding, the dropout layer follows which is then fed into the LSTM for processing the sequence. The attempt with the model created with the inclusion of FastText embedding vectors resulted in more comprehensive text sequences than the one without the Embedding vector.

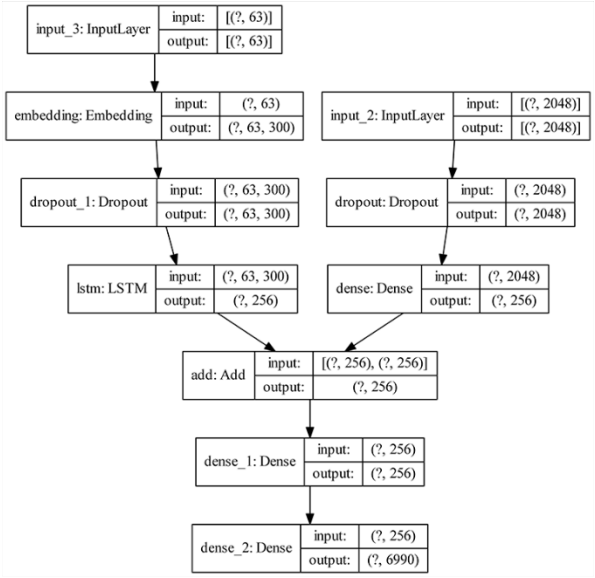

Figure 4. Merge Model Summary with 300-d FastText word vector

The Figure 4 refers to the Merge methodology where the image is left out in the LSTM network such that the LSTM manages only the caption prefix. As the prefix is encoded, the image vector is merged with the prefix vector in a separate layer. The merged vector is handled in a feedforward layer. With the development of the Deep learning models with 20 individual models which were received as an outcome of 20 epochs, the model with less Cross-entropy loss while ensuring that the model does not overfit was chosen to proceed with. The pretrained word vectors of 300-d from fastText for Tamil was chosen to be included in the research. The image captioning process can be grouped into three modules as Image Feature Extractor, Sequence Processor and Decoder. The following subparagraphs give a general introduction of the components of our model.

1 .Image Feature Extractor: VGG16 and Inception-V3 were finalized among the five pretrained models available on Keras. Among them, VGG-16 was used to test the proof-of-concept with 8092 images dataset derived from the Flicker30K dataset. The results were satisfactory considering the performance of VGG-16 for other non-English complex languages. However, Inception-V3 being a more advanced CNN model, has better performance on ImageNet dataset in comparison with VGG-16. Hence, Inception-V3 pretrained on ImageNet dataset was chosen as the finalized CNN model for the research. The images were pre-

processed with the Inception-V3 model without the output dense layer since the research does not involve any image classification functionalities. The extracted features predicted by this model will be used as the input for the SoftMax layer.

2. Sequence Processor: The Sequence Processor is a word embedding layer for handling the text input, followed by a LSTM Recurrent Neural Network layer as stated previously. Word embeddings, that is, the vectors that represent known words prior to being fed to the RNN, consist of vectors that have been randomly initialized. The purpose of the LSTM, a type of RNN, is primarily to take a prefix of embedded words and produce a single vector that represents the sequence. The LSTM neural-language model begins with 'startseq' token, an artificial word placed at the beginning of each generated caption as a prefix when predicting the first word. The same way, there will be an 'endseq' token denoting the end of the caption sentence.

3. Decoder: The Decoder merges the vector output from the extractor and Sequence Processor, wherein the merged output is processed by an output 'SoftMax' layer to make a final prediction over the entire output vocabulary for the next word in caption until the 'endseq' token is reached.

Experiment with a subset of Flickr30K dataset

During the early stage of the research, the Proof-of-Concept was tested with an 8092-image dataset curated from the Flickr30K dataset along with its Tamil paired corpus. The dataset was split into 6092 images for training and 2000 images for testing. The model that was trained on the training dataset in the Merge architecture for 20 epochs with 6092 steps had a loss of 3.04 and had satisfactory outputs. This involved the use of VGG-16 CNN model and the fastText word vectors for Tamil were not used. The loss function that was used is categorical cross-entropy. The aim of this is to minimize the loss to minimize the difference between the distribution of the predicted sentences and the actual captions of the image given in the training data.

The 8K model was evaluated on the test dataset on the Bilingual Evaluation Understudy (BLEU) (Papineni K et al., 2002) score. To this project we calculate BLEU scores for unigrams to 4-grams (BLEU-1 to BLEU-4 respectively) to evaluate the chosen model. As for this experiment, the BLEU scores are as follows:

i. BLEU-1: 0.468239
ii. BLEU-2: 0.288166
iii. BLEU-3: 0.188267
iv. BLEU-4: 0.086652

As per the BLEU metric definitions, the 0.46 score refers to high quality captions. In the trial run with the

8K set, a greedy search was used for the caption prediction. This means the model generates the caption word-by-word, as in, it uses the previously generated words to generate the next word.

Using the results from the 8K dataset experiment, a few changes were made to the training of the 30K dataset by swapping VGG-16 with Inception-V3 and including a word representation vector from fastText for Tamil. The latter is a pre-trained word vector for Tamil language, trained on Common Crawl and Wikipedia using fastText. Besides, the Greedy search was switched with Beam search for predictions. In contrast, Beam Search expands the scope of Greedy search and takes the best 'N' words out of the predictions. The hyperparameter 'N' is known as the Beam width and we used 3, 5, 7 and 9 as the Beam width for generation. But, the evaluation with 30K dataset was conducted with beam width 5.

III. RESULTS & DISCUSSION

Although the results from the 8K dataset experiment had a good BLEU score, it had its limitations due to the use of Greedy search for the caption generation. The generated caption for the Figure 5 is as follows:
ஒரு மனிதன் ஒரு பெரிய கட்டிடத்தின் முன் ஒரு பெரிய கட்டிடத்தின் முன் நிற்கிறான். (A man stands in front of a big building).

Figure 5. A sample image used for testing.

However, there is a noticeable issue in the generated caption regarding the repetition of the underlined phrase in the sequence. Although it does not affect the meaning of the piton, it is an obvious inconvenience which could destroy the meaning of caption in any other circumstances.

Nevertheless, with the observations made from the 8K model, the 30K model training with the inclusion of Inception-V3, Tamil word vector representation from fastText and Beam search algorithm was conducted. The Loss vs Epochs graph in Figure 6 for the 30K model for its first 10 epochs proved that the model has a good learning rate, and it could be improved with a much higher value of epoch. Hence, the model was trained until 20 epochs, resulting in a model with 3.59 loss in the 20th epoch model.

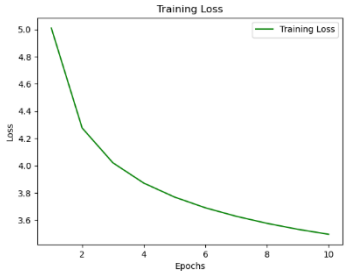

Figure 6. Training Loss vs Epoch graph for 30K model for 10 epochs.

ஒரு கருப்பு சட்டை மற்றும் நீல நிற ஜீன்ஸ் அணிந்த ஒரு மனிதன் ஒரு நடைபாதையில் நடந்து செல்கிறான்
(A man wearing black and blue jeans is walking across the pedestrian cross.)

இரண்டு நாய்கள் ஒரு வயல் வழியாக ஓடுகின்றன
(Two dogs are running through a field.)

ஒரு மனிதன் ஒரு பாறை கடற்கரையில் ஒரு பெரிய பாறையை ஏறுகிறான்
(A man is climbing a huge rock in a rocky beach.)

Figure 7

A few examples of captions generated by the model can be observed in the Figure 7. The BLEU scores for 30 K using the test dataset are:

i. BLEU-1: 0.370611
ii. BLEU-2: 0.217844
iii. BLEU-3: 0.160439
iv. BLEU-4: 0.077670

The scores are comparatively lower than that of the 8K model, but their performance in the caption generation has been more than satisfactory. Regardless of the performance, this model consists of limitations when faced with images with entities which were not included in the training dataset. An example is the Figure 8 and its generated caption which Figure 7. Best

Tamil Captions Generated by the model. The English translations are provided for the reader's comprehension.

misidentifies the laptop as a book is shown below: ஒரு பெண் ஒரு மேஜையில் உட்கார்ந்து ஒரு புத்தகத்தைப் படிக்கிறாள். (A girl sitting at a desk is reading a book.)

Figure 8. A sample image used for testing.

IV. CONCLUSION

This paper is a preliminary study on the potential of generating captions for images in Tamil language. This research has utilized the "Merge" model architecture proposed by Tanti M, Gatt A and Cammilleri KP (2017) and the methodology carried out for Turkish language in by Yılmaz BD et al. (2019, pp. 1-5). This Merge model is a variant of the CNN-LSTM model proposed by Vinyals O et al. (2015). This study began with the creation of a paired 30K dataset in Tamil inspired from the Flickr30K dataset using machine translation and manual review process. There are a few limitations in the caption generation as to the inaccurate identification of the model in certain unique entities which were not present in the training dataset. The results obtained through this research certainly proves that the performance of the model can be improved further with a more refined corpus.

REFERENCES

Al-muzaini HA, Tasniem N and Benhidour H (2018) Automatic Arabic Image Captioning using RNN-LSTM-Based Language Model and CNN, IJACSA.

Bojanowski P et al. (2016) Enriching Word Vectors with Sub word Information, CoRR.

Chen X and Zitnick CL (2015) Mind's eye: A Recurrent Visual Representation for Image Caption Generation, IEEE Conf. on CVPR, 2422–2431.

Gu J et al. (2018) Unpaired Image Captioning by Language Pivoting, 15th Eur. Conf.

Jaffe A (2017) Generating Image Descriptions using Multilingual Data, WMT.

Karpathy A and Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions, IEEE Conf. on CVPR, 3128–3137.

Lin T et al. (2014) Microsoft COCO: Common Objects in Context, European Conference on Computer Vision, 740-755.

Papineni K et al. (2002) BLEU: A Method for Automatic Evaluation of Machine Translation, Available at: <https://doi.org/10.3115/1073083.1073135>[Accessed 8 June 2021].

Plummer B et al. (2015) Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models.

Tanti M, Gatt A and Cammileri KP (2017) What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator? 10th Int. Conf. on Natural Lang. Generation.

Tanti M, Gatt A and Cammileri KP (2018) Where to put the Image in an Image Caption Generator, 11th Int. Conf. on Natural Lang. Generation.

Vinyals O et al. (2015) Show and tell: A Neural Image Caption Generator, IEEE Conf. on CVPR., 3156-3164.

Xu K et al. (2015) Show, attend and tell: Neural image caption generation with visual attention, Proceedings of the 32Nd Int. Conf. on Mach. Learn, 2048–2057.

Yılmaz BD et al. (2019) Image Captioning in Turkish Language, Innovations in Intelligent Systems and Applications Conference (ASYU), 1-5.

Zhang C et al. (2018) Recurrent Attention LSTM Model for Image Chinese Caption Generation, International Symp. on Advanced Intelligent Systems.

AUTHOR BIOGRAPHIES

Gobiga Rajalingam is reading B.Sc (Honors) in Information Technology at Department of Computing, Rajarata University of Sri Lanka. She is currently working in Data Engineering sector. Her research interest lies in Natural Language Processing.

Wiraj Udara Wickramarachchi is a Lecturer of the Department of Computing, Rajarata University of Sri Lanka. Currently, he is reading for his doctoral degree at Wuhan University of Technology, China. His research interests are Information Security and Image Processing.