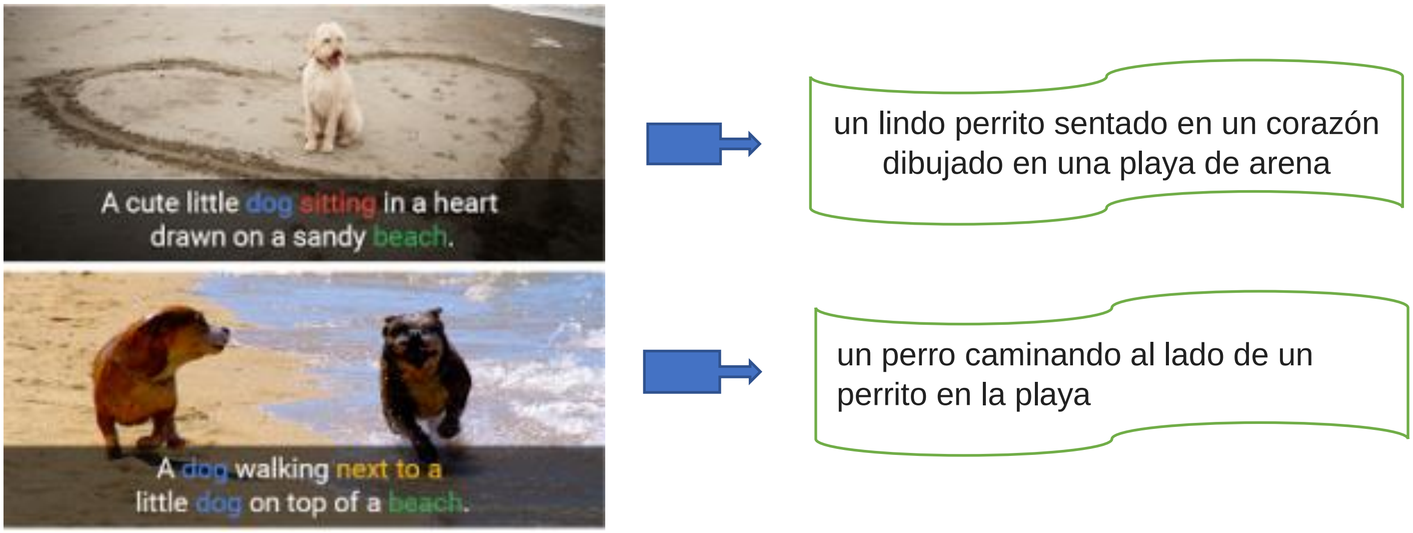
IDS 576 – PROJECT

Automatic Content Creation using Image Captioning and Translation

YOUTUBE LINK: https://www.youtube.com/watch?v=ux-lV_EM-cA


A cute little dog sitting in a heart drawn on a sandy beach.
A dog walking next to a little dog on top of a beach.

un lindo perrito sentado en un corazón dibujado en una playa de arena

un perro caminando al lado de un perrito en la playa

Nimisha Asati - 678028458

Sai Spandana Ettireddy - 652846670

Gughanraj Selvaraju - 662075168

PROBLEM STATEMENT:

Automatic content creation (ACC) is one of the promising areas of research in Artificial Intelligence. Generating a sentence describing an image combines recent advances in computer vision, Natural Language Processing and machine translation. But the improvements have been more concentrated to English language alone. In parallel, there has been huge strides in language translation achieving near human perfection. Both areas of research use deep learning and neural networks as one of the viable options to achieve the desired target. In this project, we have combined the modules of image captioning and language translation to establish effective story creation in a language other than English.

The objective of the project is to work on integrating two problem statements:

1) Image Captioning:

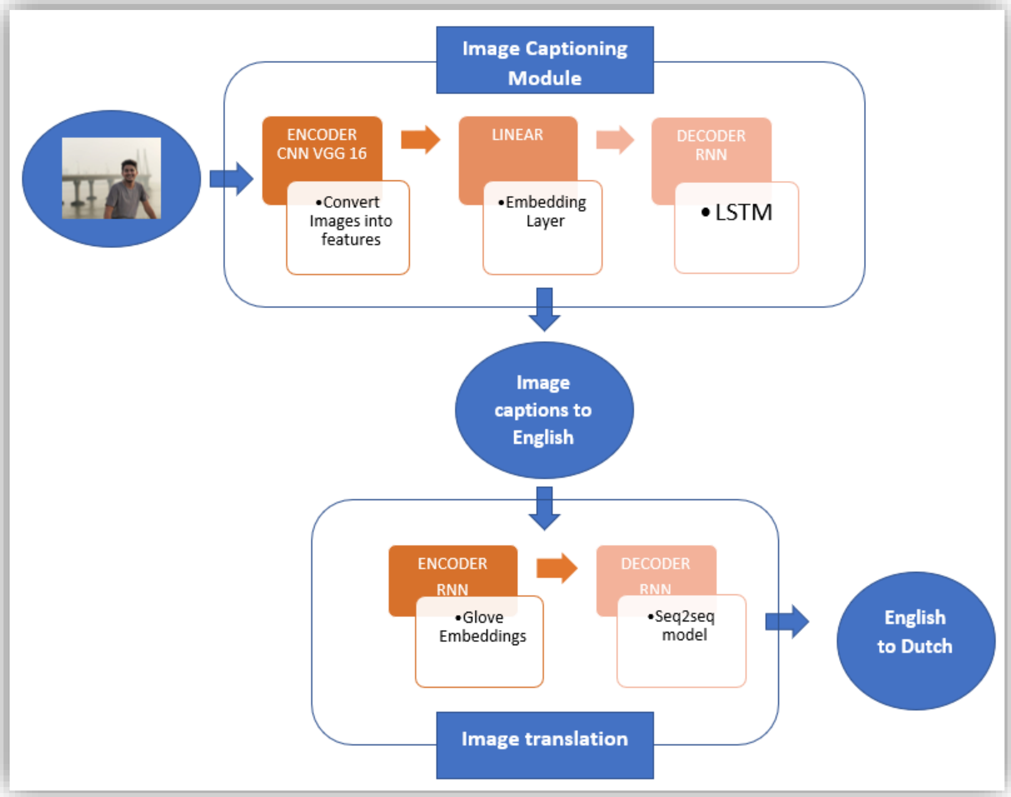
Auto-Generating textual description of images,

2) Language Translation model:

Translating the generated English text into other languages for ease of understanding to the targeted local audience.

SOLUTION ARCHITECTURE:

We present a generative model-based approach on a deep recurrent architecture using encoder-decoder network for image captioning and seq2seq modeling approach for translation.



Architecture for the solution of Automatic Content Creation

DATASET:

We used **Flickr8k dataset** which has 8,000 photos and up to 5 captions for each photo which provide clear descriptions of the entities and features. This has been split into 6000 train images, 1000 validation and 1000 test images.

DATA PREPROCESSING AND CREATION:

First, we loaded data including both captions and images. Then, segregated the data into different folders (train, test) and IDs from captions. We generated the vocabulary using indexes and tokenized to words for training the data. Used NLTK library for the preprocessing, parsing and tokenizing of words.

Pre-processing for Image Captioning: Image captioning preprocess has three parts:

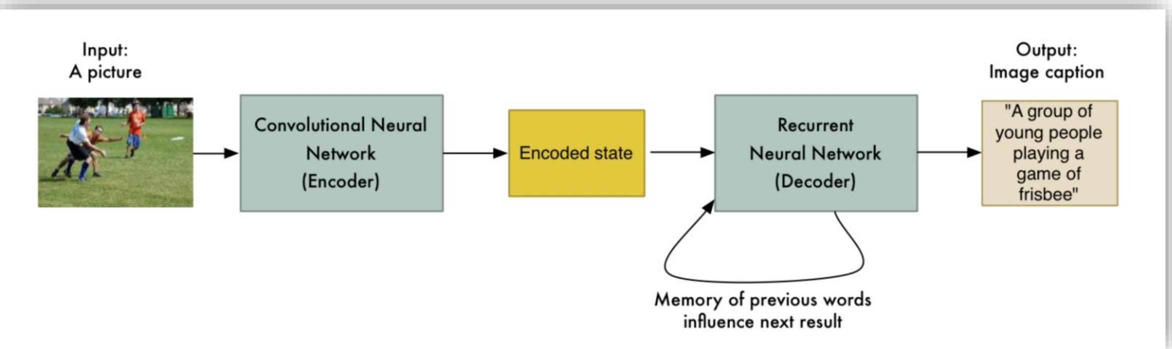
1. **Train-Test split**: 8K images are split into train, test and validation and the its respective captions are split based on the ids and copied to separate folders
2. **Vocabulary Creation**: All the image caption from the Flickr8k dataset is converted into ids-word structured vocabulary to be used in the training process to get **2550 words**
3. **Image Processing:** Any image that is read for training is converted to size of 224x224x3 because of the usage of pretrained network and its caption is converted to ids using the above created vocabulary

Pre-processing for Language Translation: After reading the language pairs file we –

- converted Unicode characters to ASCII and removed punctuations
- splitted into language pairs and trimmed them
- created input tensor and target tensor and appended the EOS token to both sequences.
- created the Glove embedding matrix which has pre-trained weights for the words from the vocabulary.

MODEL BUILDING:

1. Image Captioning


Input:
A picture
Convolutional Neural Network (Encoder)
Encoded state
Recurrent Neural Network (Decoder)
Output:
Image caption
"A group of young people playing a game of frisbee"
Memory of previous words influence next result

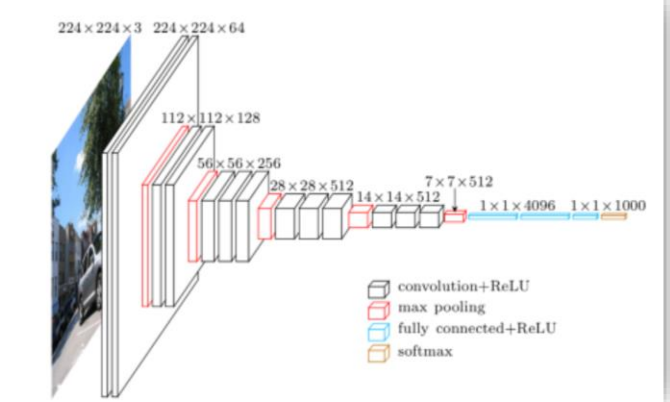
Language translation model architecture: Encoder-Decoder network

- ❖ **Transfer learning approach** → to leverage the pre-trained model's weighted layers to extract features. Then updated the weights of the model's layers during training with new data for the new task.

❖ Image encoder → convolutional neural network (CNN)

Understanding an image largely depends on obtaining image features. For Encoder CNN, we created features for each image using **VGG16** (Oxford Visual Geometry Group) pretrained network. This network takes input of size (224,224,3). The output layer contains 1,000 nodes. As we are not using VGG16 for the sake of the classification, but we just need it for extracting features, **we removed the last layer from the network**.

- We are using VGG16 because it is the simplest with only 3x3 convolution and 2x2 pooling layers.
- VGG shows that depth of the network plays an important role. Deeper networks give better results.
- The width of the network starts at a small value of 64 and increases by a factor of 2 after every sub-sampling/pooling layer. It achieves the top-5 accuracy of 92.3 % on ImageNet.
  - One drawback of VGGNet is that this network is usually big, with around 160M parameters.

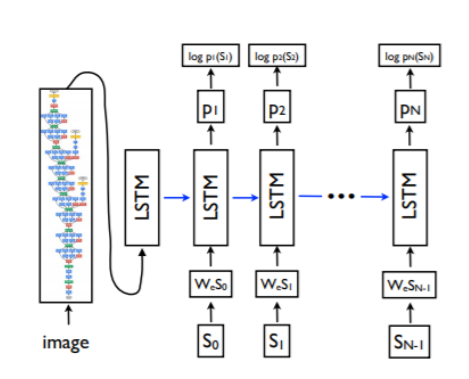


Macro-architecture for VGG16

❖ Image Decoder → long short-term memory (LSTM) network

LSTM is a recurrent neural network which is used in problems with temporal dependencies. It captures information about previous states to better inform the current prediction through gates in memory cell state.

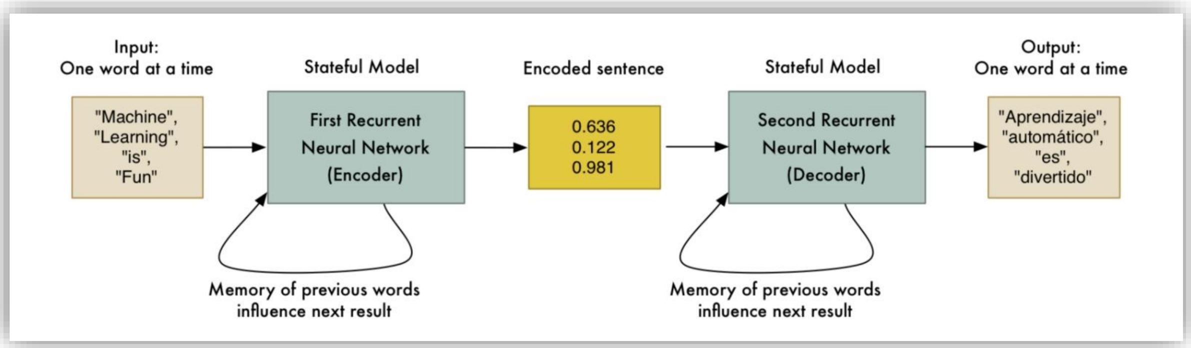
For Decoder RNN, we created **word embeddings** using one hidden layer, weights and biases. We initialized LSTM layer with these parameters and generated single words for the captions. We used **Greedy search** to generate captions for the given image.



LSTM model combined with a CNN image embedder and word embeddings.

The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections.

2. Language Translation



Language translation model architecture: Encoder-Decoder network

- Used pre-trained **Glove Embeddings** to learn word vectors capturing more precise syntactic and semantic relationships. They create linear substructures of the word vector space. We used Glove 100 which has words represented by vectors of 100 dimensions.

- **Seq2Seq translation:**

Unlike sequence prediction with a single RNN, where every input corresponds to an output, the seq2seq model frees us from considering sequence length and order, which makes it ideal for translation between two languages. For the translation, we'll need a unique index per word, to use as the inputs and targets of the networks later. To keep track of all this, we used a helper class called Lang which has word → index (word2index) and index → word (index2word) dictionaries, as well as a count of each word word2count to use to later replace rare words.

- Model consists of **two RNNs called the encoder and decoder**

We used RNN because in RNN inputs are related to each other, which enables the network to predict a better output using memory from all previous words.

- The **encoder** of a seq2seq network is RNN that outputs a vector and a hidden state for every word from the input sentence and uses the hidden state for the next input word.
- The **decoder** is another RNN that takes the encoder output vector(s) and outputs a sequence of words to create the translation.

FINE TUNING:

For Image Captioning: We used supervised learning with **cross entropy loss** & **ADAM optimizer** to train the neural network with mean squared loss.

For Language Translation: We tried **Adam & SGD optimizers** with changing **Learning rates** (0.01,0.05,0.001,0.0001) and changing **momentum** (0.01,0.05,0.1) for both encoder and decoder models.

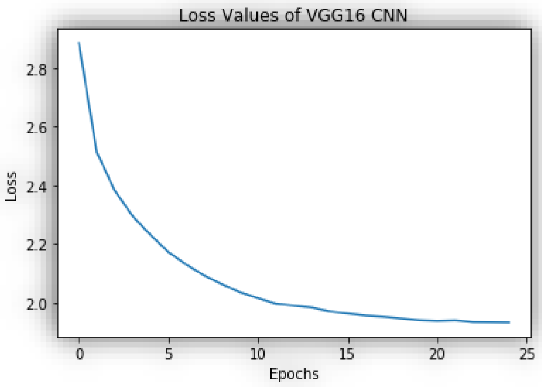
Stochastic Gradient Descent optimizer with learning rate=0.01, momentum=0.05 is better converging with minimal loss of 1.90 after 100000 epochs.

PROGRAMMING ENVIRONMENT:

- Jupter Notebook.
- Numpy for high performance multidimensional array.
- Matplotlib for plotting results.
- PyTorch library for neural network layers and natural language processing.
- pytorch-vision for image transformations.

MODEL EVALUATION AND PERFORMANCE METRICS:

❖ **Loss Values:** Due to the high training time we could run the model for limited number of epochs.



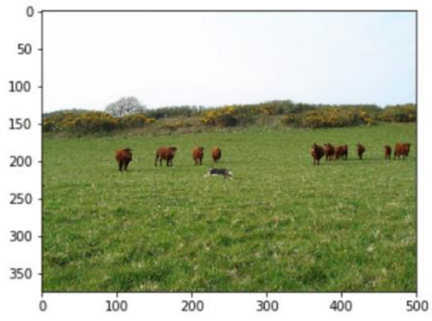
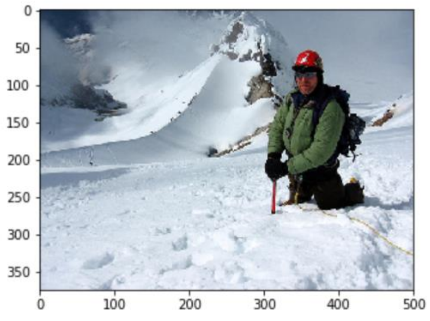
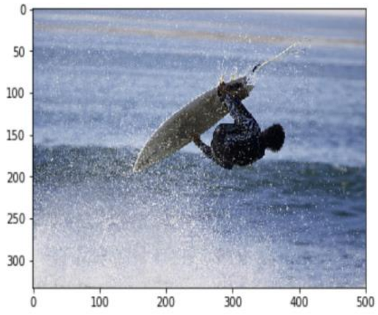
Loss Values for 25 epochs: VGG16 CNN

❖ **Bleu Score:**

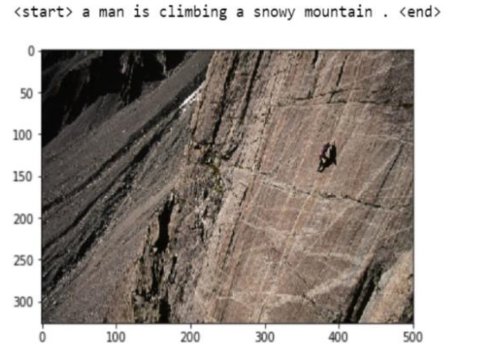
To evaluate our image captioning model, we used Bilingual Evaluation Understudy algorithm (BLEU score) as our evaluation metric. The BLEU score comes from work in machine translation, which is where image captioning takes much of its inspiration. BLEU scores are calculated as network generated output against human-annotated reference captions. So, below are the human annotated captions that we did manually.

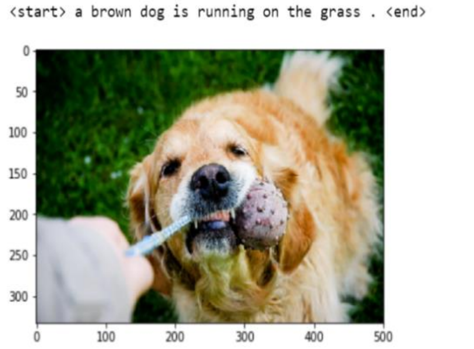
MODEL	PARAMETERS	BLEU SCORE
Image Captioning	Unigram	0.18
Image Captioning	Bigram	0.03
Image Captioning	Trigram	0.008

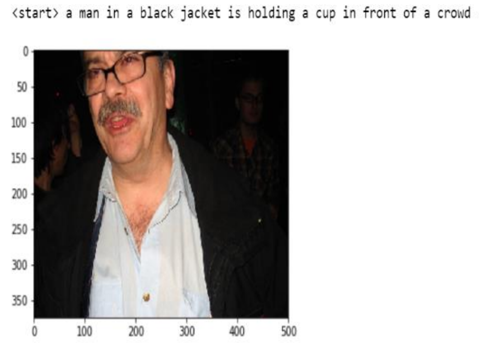
❖ **Human Evaluation:**

Describes Without Error		
`<start> a dog runs through a field . <end>` 	`<start> a man is climbing a snowy mountain . <e` 	`<start> a man in a wetsuit is riding a surfboard . <end>` 

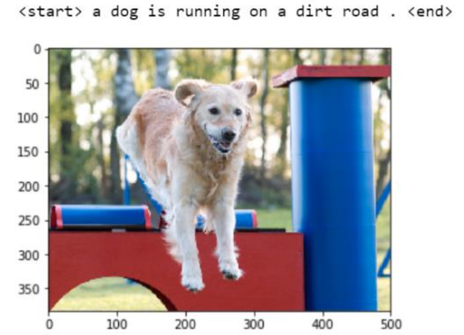
Describes with Minor Error

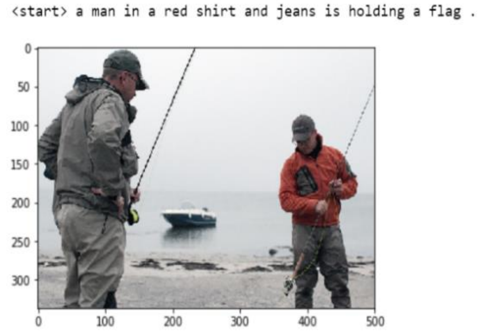

<start> a man is climbing a snowy mountain . <end>

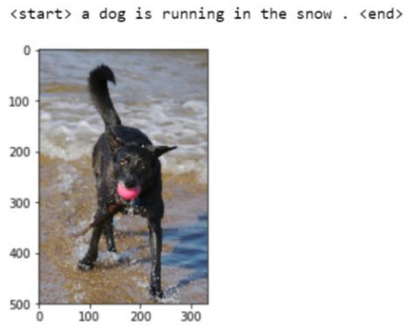

<start> a brown dog is running on the grass . <end>


<start> a man in a black jacket is holding a cup in front of a crowd .

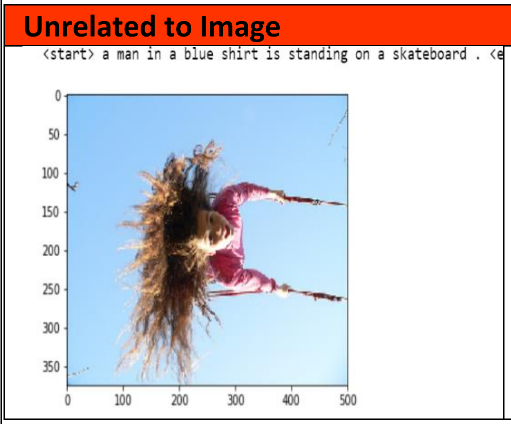
Somewhat Related to Image

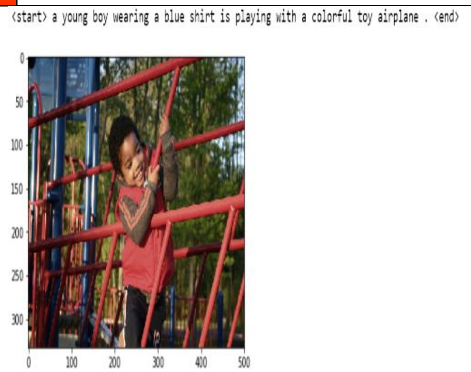

<start> a dog is running on a dirt road . <end>

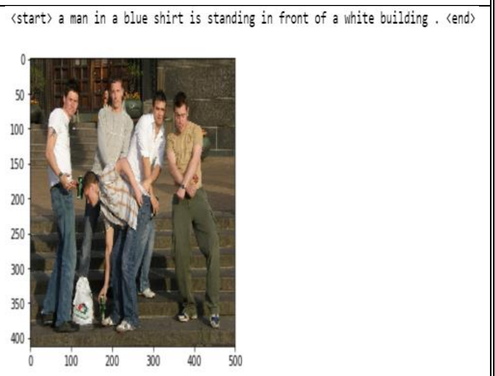

<start> a man in a red shirt and jeans is holding a flag .


<start> a dog is running in the snow . <end>

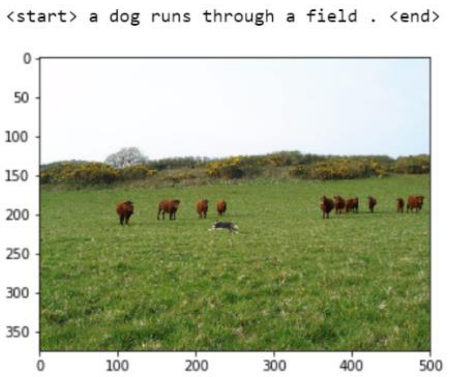
Unrelated to Image


<start> a man in a blue shirt is standing on a skateboard . <e


<start> a young boy wearing a blue shirt is playing with a colorful toy airplane . <end>


<start> a man in a blue shirt is standing in front of a white building . <end>

- **Final Model Output:**


<start> a dog runs through a field . <end>

Caption: a dog runs through a field

Dutch Translation: een hond een een

Back to English: A dog a one

Image captioning

Language Translation

INFERENCE / INSIGHTS:

- Image Captioning module is performing well on the describing the objects in an image, but we feel the actions in the images can be improved further.
- We observed that, for a sentence with more than 3 words the translation is not up to mark but for a sentence with 3 words translation is good.
- LSTM is facing challenges in generating long length sentences because image information is fed only at the beginning of the process, it may face vanishing gradient problems. The role of the words generated at the beginning is also becoming weaker and weaker.
- RNN is unable to learn to connect long range information that is potentially incapable of learning dependencies in a longer sentence.

FUTURE WORK AND IMPROVEMENTS:

- Usage of BLEU alone in evaluating the Image captioning module doesn’t take into the meaning of the sentence. There is a need for better evaluation metrics
- The Captions generated are just standalone sentences for a single image we can extend this idea to a group of images resulting in a neural story teller
- Simple RNNs suffer from the vanishing gradient problem which makes it difficult to learn and tune the parameters in the earlier layers
- Other variants, such as long short-term memory (LSTM) networks, residual networks (ResNets), and gated-recurrent networks (GRU) were later introduced to overcome this limitation
- Hence, using LSTM will improve the language Translation learning the weights better
- We can implement this model to translate to multiple native languages from English for the ease of understanding for the audience.

REFERENCES:

1. Show and Tell: A Neural Image Caption Generator; Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan
2. https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5
3. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION; Karen Simonyan ∗ & Andrew Zisserman +
4. https://github.com/topics/image-captioning
5. https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa
6. Deep Neural Language Models for Machine Translation; Minh-Thang Luon, Michael Kayser, Christopher D. Manning
7. https://github.com/OValery16/Language-Translation-with-deep-learning-
8. Neural Image Caption Generation with Weighted Training and Reference; Guiguang Ding, Minghai Chen, Sicheng Zhao, Hui Chen, Jungong Han, Qiang Liu