IMAGE CAPTIONING WITH SEQUENCE ENCODING AND DECODING

Sashank Bonda	17CS30031
Karanam Tejendhar	17CS30019
Manas Madine	17CS10025
Harsh Pritam Sanapala	17CS30016

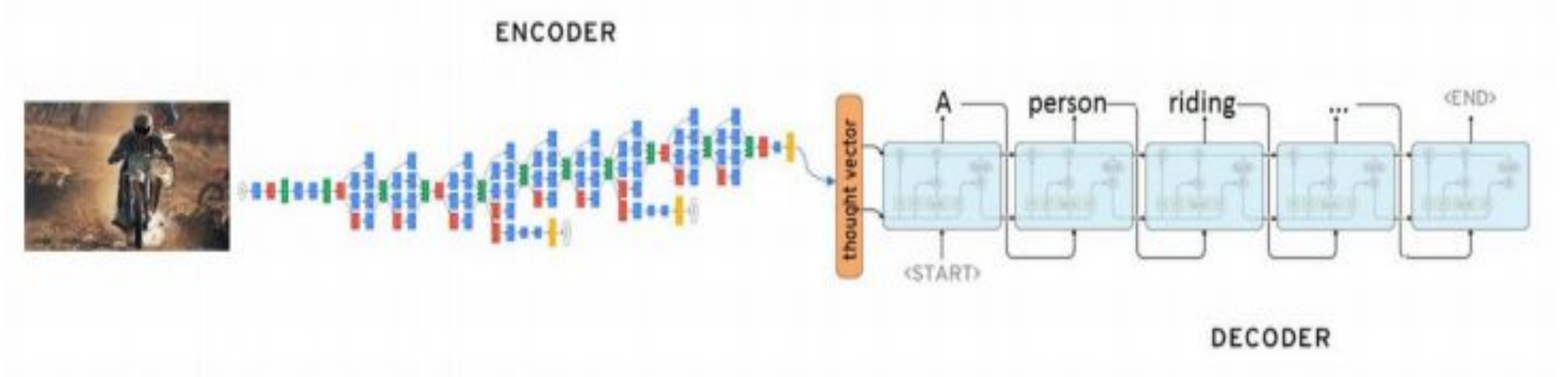
Introduction

- In this work, we implement an image captioning model which takes an image as an input and gives out a description of the image as an output.
- In this work, we try to improve the typical Image captioning model which is made up of a CNN encoder and a RNN decoder.
- We achieve this by employing an attention layer as an interface between the encoder and the decoder in addition to using a Mask RCNN as an encoder and a GRU as a decoder(which is nothing but an advanced and more time efficient form of LSTM) instead of the typical components.
- We use beam search in our prediction model for improved results.
- Our results are measured using BLEU score as a metric.

RELATED MODELS

1) ENCODER DECODER BASED CAPTIONING

- The encoder is a CNN which converts input picture to a feature vector.
- Decoder is a RNN which then generates the captions based on image feature vector.[1]



RELATED MODELS

2) ATTENTION BASED CAPTIONING

- **The problem of previously described model is the feature vector that CNN generated represents the whole input picture.**
- **As a result the RNN generates each word by looking at whole input picture.**
- **This is inefficient and can also lead to wrong results.**
- **The problem is mitigated by adding an attention layer which is a small neural which can learn where to focus in the picture when RNN is generating the target word.**[2]

RELATED MODELS

3) SEQUENCE BASED CAPTIONING

- The other problem of the base CNN+RNN model is the imbalance between representations of the image and the representations of the words.
- The RNN decoder is generating the caption sentences in different time steps however the CNN feature of the image is one time step only .
- In Chang's paper[3] in order to fix this problem an image is converted into a sequence of detected objects using object detection model.
- The encoder is changed from CNN to R-FCN. In this way a sequence of objects can be translated to a sequence of words using same model as machine translation.

DATASET USED

COCO DATASET has large scale of images along with multiple captions to each image.There are total of 81219 unique images in total.we used 80/10/10 split for training , testing and validation respectively. The <start> and <end> token were added to captions as part of preprocessing step. For image preprocessing the images were resized to 299*299 pixels to be given as input and is normalized so that it's pixels contain values between -1 and +1.

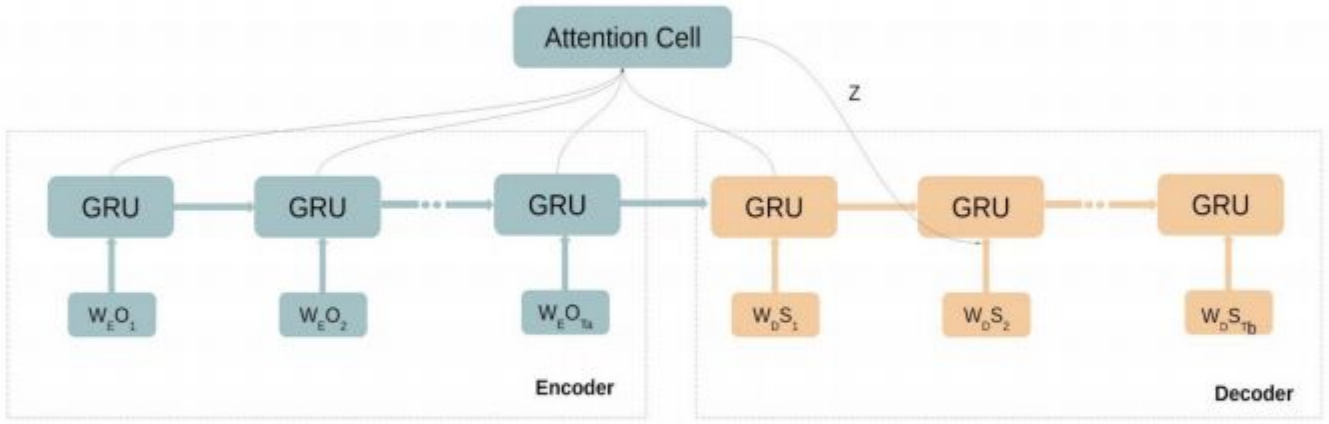
Image A

Image B

Image C

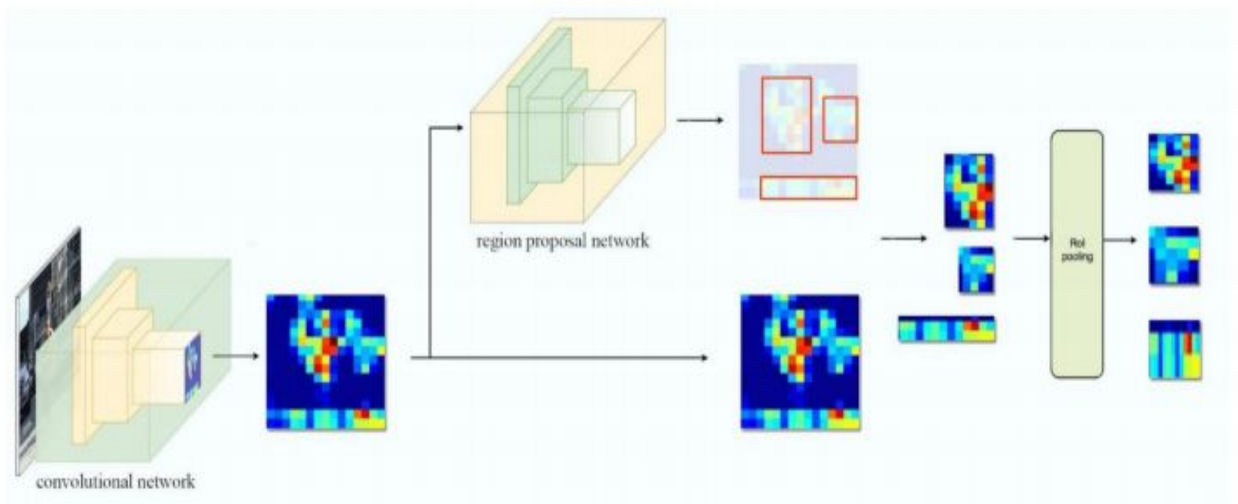
<start> Children are playing baseball on a muddy baseball diamond <end> (Caption A1 for Image A)
<start> A purple bus and a man dressed as a nun on a tall bicycle <end> (Caption B1 for Image B)
<start> A child waits for a pitch during a game of baseball <end> (Caption A2 for Image A)
<start> a wine glass sits in front of some plates of food <end> (Caption C1 for Image C)

OUR MODEL



(Figure 2: Encoder/Decoder of our model)

Sequence Representation of Images

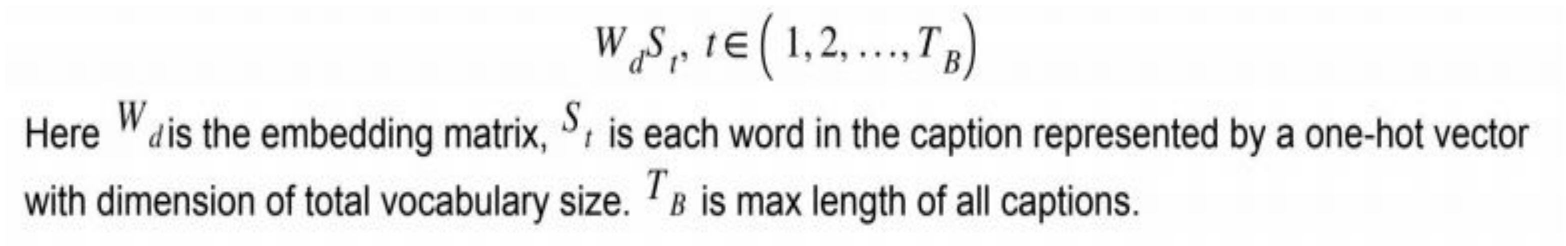


The image sequence can be represented as the following sequence.

$$W_e O_t, t \in (1, 2, \ldots, T_A)$$

Here W_e is the embedding matrix, O_t is the CNN feature of t-th object given by the Mask-RCNN model. T_A is the maximum length of the input sequence.

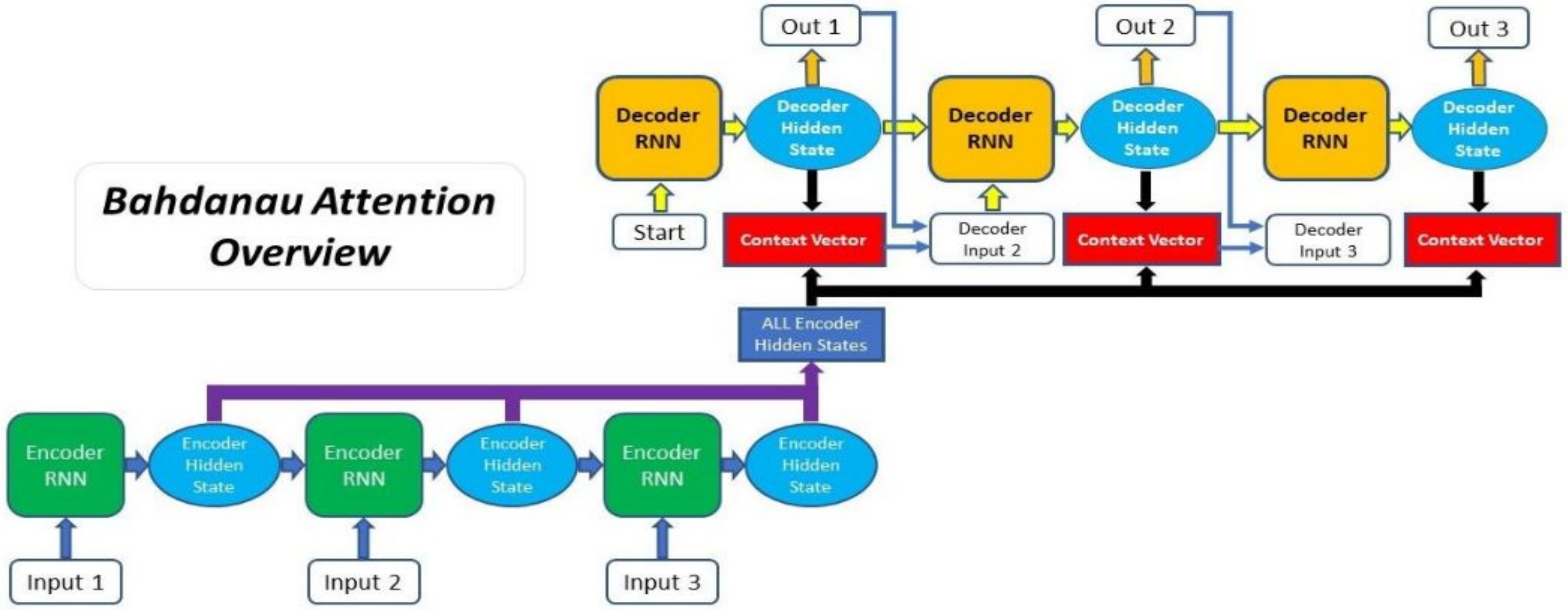
Sequence Representation of captions



$$W_d S_t,\ t \in \left(1, 2, \ldots, T_B\right)$$

Here W_d is the embedding matrix, S_t is each word in the caption represented by a one-hot vector with dimension of total vocabulary size. T_B is max length of all captions.

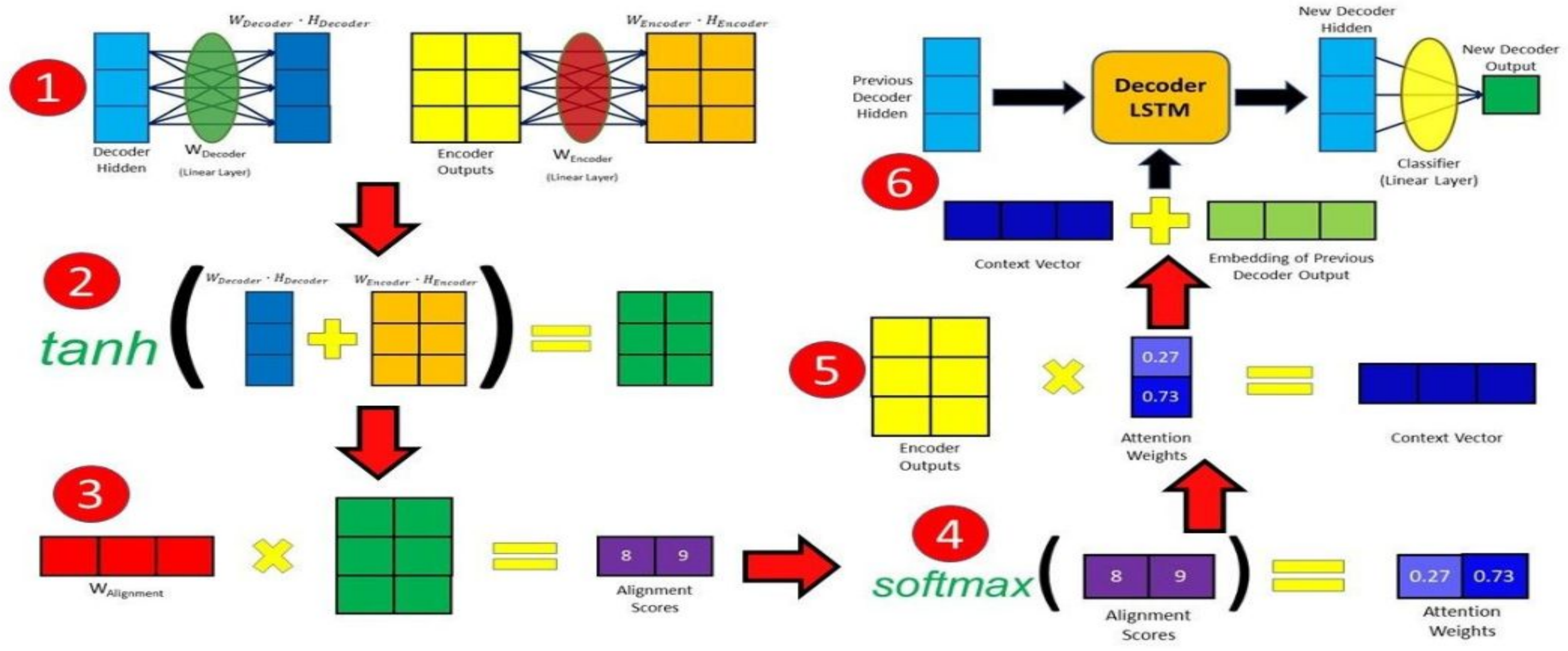
Bahdanau Attention



Overall process for Bahdanau Attention seq2seq model

BAHDANAU ATTENTION

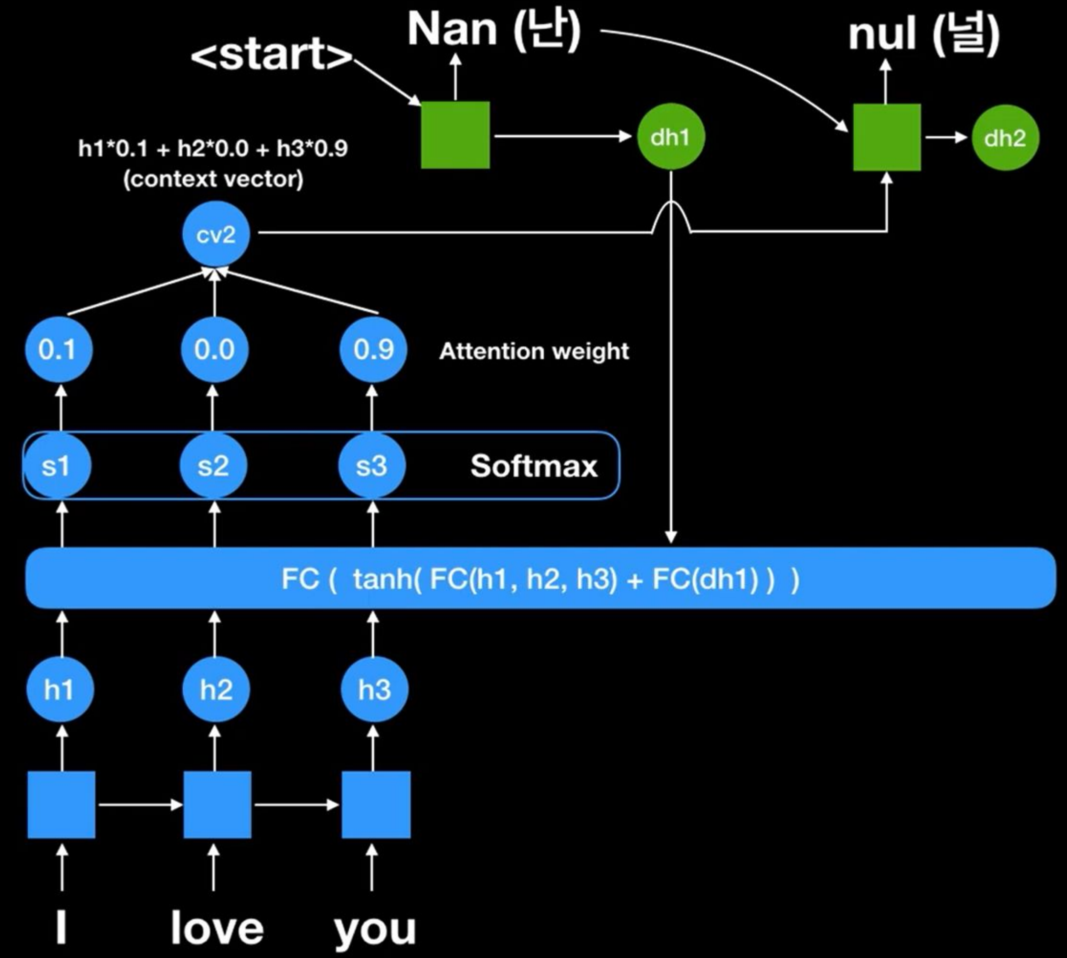
- **Producing the Encoder Hidden States - Encoder produces hidden states of each element in the input sequence**
- **Calculating Alignment Scores between the previous decoder hidden state and each of the encoder's hidden states are calculated**
- **Softmaxing the Alignment Scores - the alignment scores for each encoder hidden state are combined and represented in a single vector and subsequently softmaxed**
- **Calculating the Context Vector - the encoder hidden states and their respective alignment scores are multiplied to form the context vector**
- **Decoding the Output - the context vector is concatenated with the previous decoder output and fed into the Decoder RNN for that time step along with the previous decoder hidden state to produce a new output**



score = FC(tanh(FC(encoder_output) + FC(hidden_state)))

attention weights = softmax(score, axis = 1)

context vector = sum(attention weights * encoder_output, axis = 1)

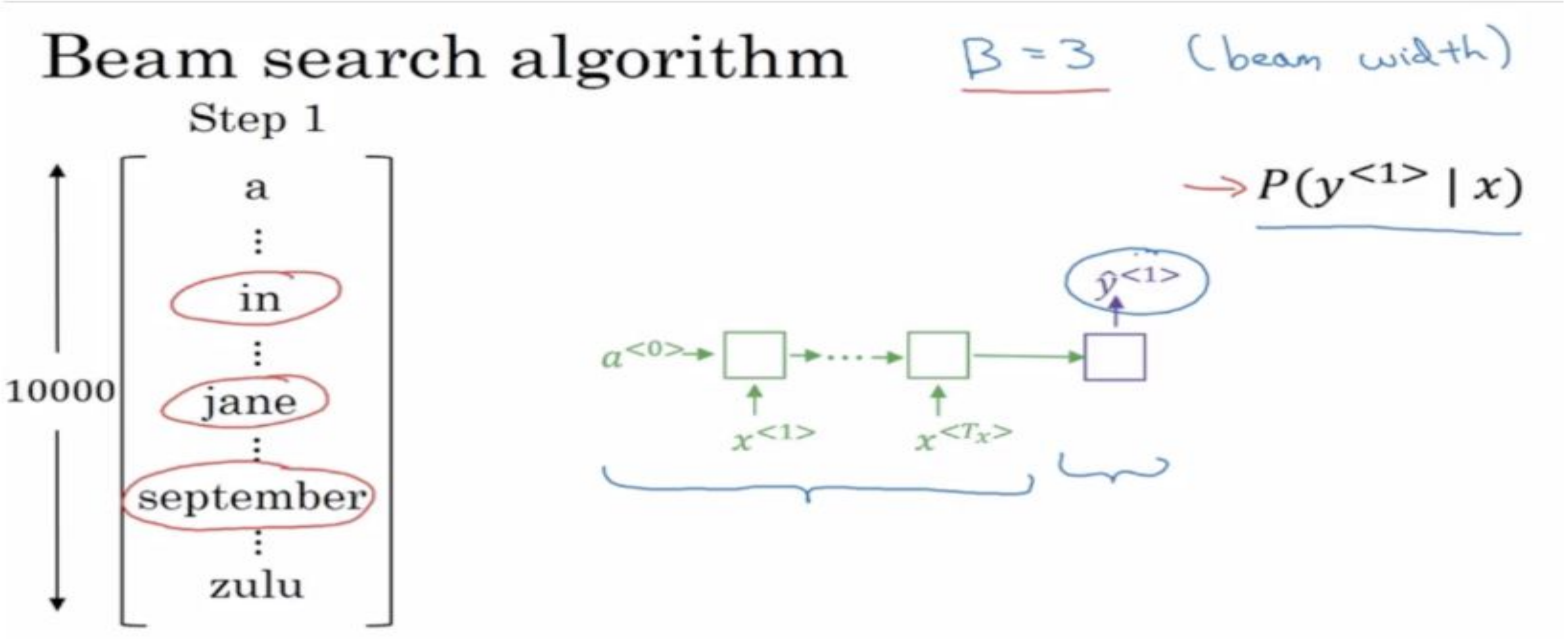


Reference: Neural Machine Translation by Jointly Learning to Align and Translate (https://arxiv.org/pdf/1409.0473.pdf)

BEAM SEARCH

- **In contrast to greedy search, instead of greedily choosing the most likely next step as the sequence is constructed,beam search expands all possible next steps and keeps the *k* most likely, where *k* is a user-specified parameter .**
- **It also controls the number of beams or parallel searches through the sequence of probabilities.**
- **The local beam search keeps track of k-states rather than just one.**
- **It begins with k randomly generated states.**
- **At each step, all the successors of all k states are generated.**
- **If any one is a goal, the algorithm halts.**
- **Otherwise, it selects the k best successors from the complete list and repeats.**

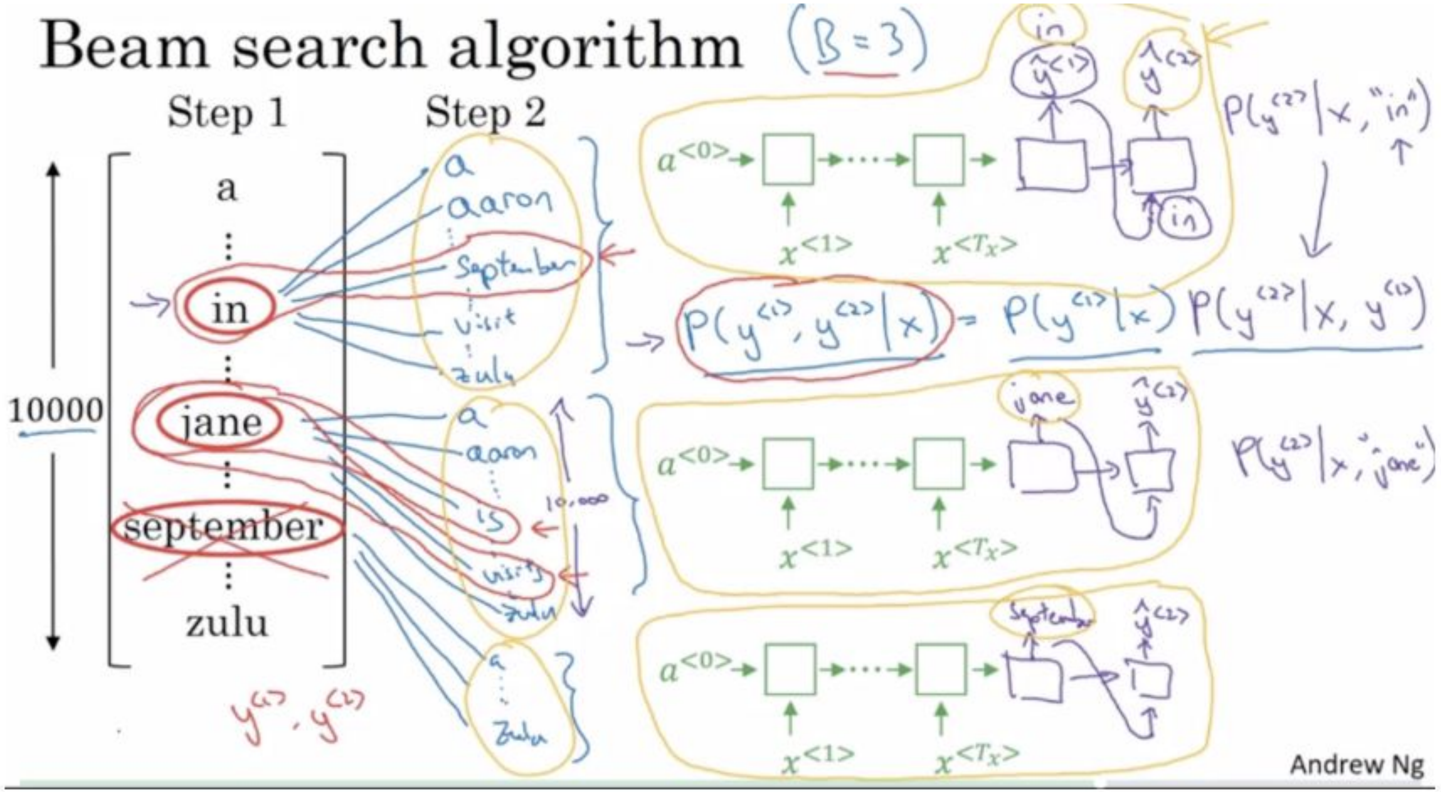
BEAM SEARCH



Beam search algorithm $\underline{B = 3}$ (beam width)

Step 1

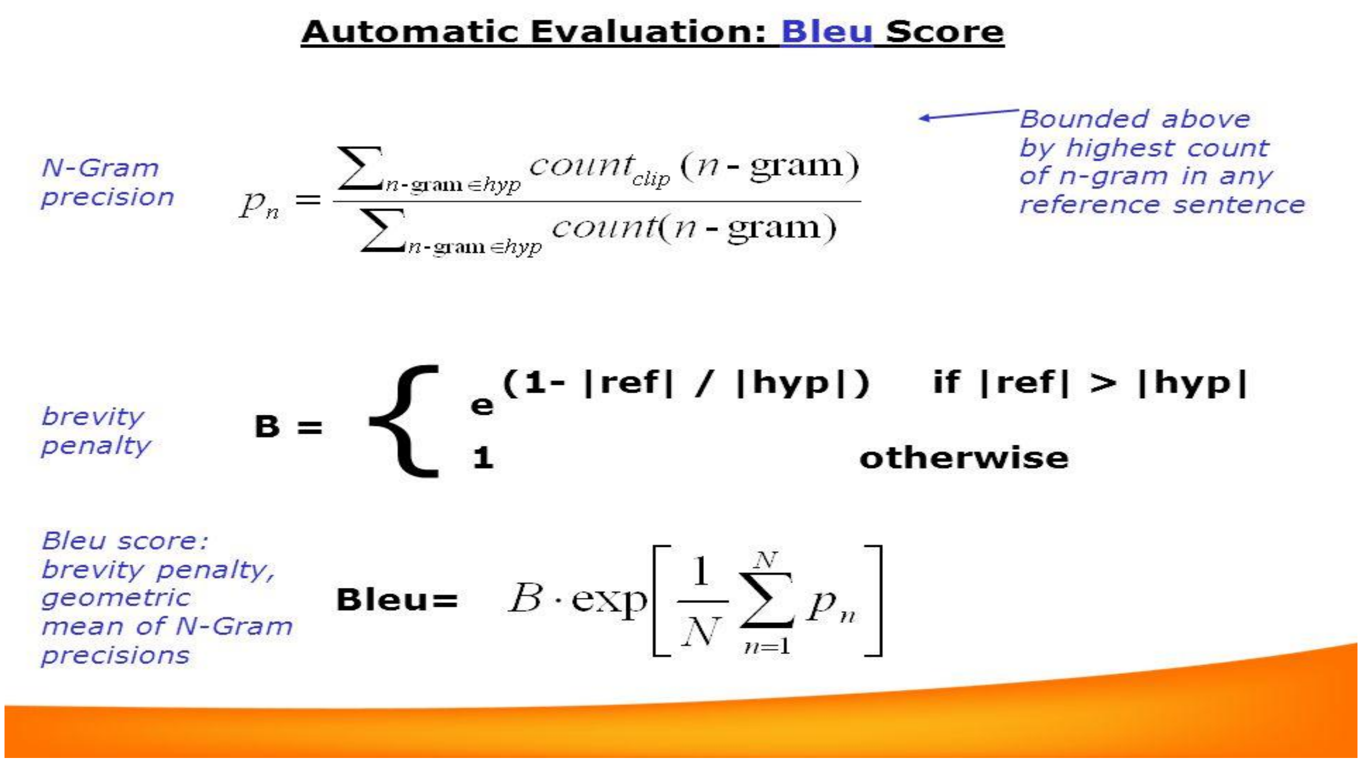
STEP 2


Beam search algorithm
(B=3)
Step 1
Step 2
a
in
jane
september
zulu
10000
$y^{<1>}, y^{<2>}$
a
aaron
september
visit
zulu
a
aaron
is
visits
zulu
10,000
a
zulu
$a^{<0>}$
$x^{<1>}$
$x^{<T_x>}$
in
$\hat{y}^{<1>}$
$\hat{y}^{<2>}$
$P(y^{<2>} | x, \text{"in"})$
$P(y^{<1>}, y^{<2>} | x) = P(y^{<1>} | x) P(y^{<2>} | x, y^{<1>})$
jane
$P(y^{<2>} | x, \text{"jane"})$
september
Andrew Ng

EVALUATION METRIC

N-gram means n sized substring

By producing short captions it is obvious that our bleu score will be high so to penalise that we use brevity penalty



Automatic Evaluation: Bleu Score

N-Gram precision

$$p_n = \frac{\sum_{n\text{-gram} \in hyp} count_{clip}(n\text{-gram})}{\sum_{n\text{-gram} \in hyp} count(n\text{-gram})}$$

Bounded above by highest count of n-gram in any reference sentence

brevity penalty

$$B = \begin{cases} e^{(1 - |ref| / |hyp|)} & \text{if } |ref| > |hyp| \\ 1 & \text{otherwise} \end{cases}$$

Bleu score: brevity penalty, geometric mean of N-Gram precisions

$$\textbf{Bleu} = B \cdot \exp\left[\frac{1}{N}\sum_{n=1}^{N} p_n\right]$$

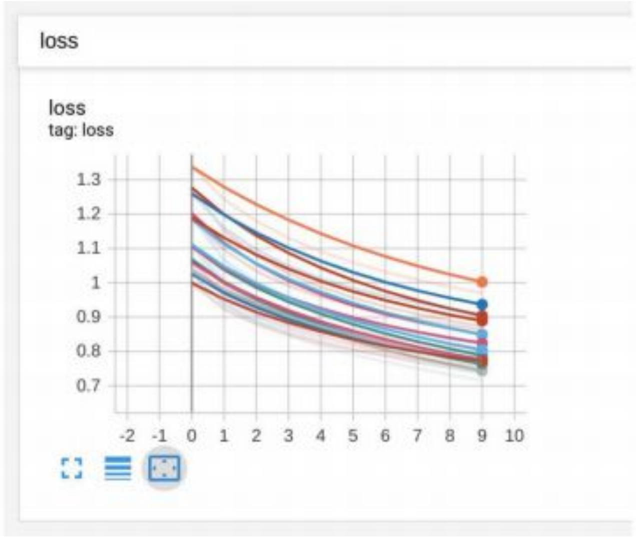
HYPER PARAMETER TUNING

Mini batch size: 16,32,64, Learning Rate:0.1,0.01,0.001,0.0001, hidden-unit size: 256,512,1024

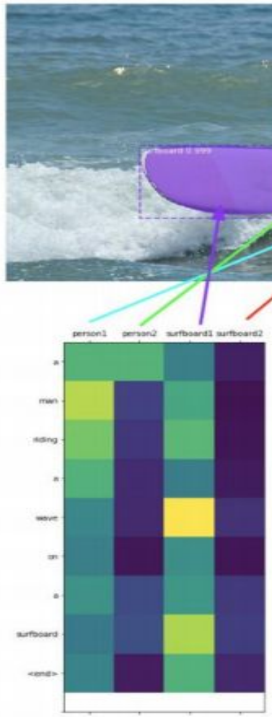
Plot of all 36 combinations using Tensorboard (y-axis: LOSS,x-axis:#of Epoch)

The combination of (64,0.001,512) gave the lowest loss after training over 10 epoch

To avoid overfitting,we tried epoch from 0 to 20,after each epoch we calculated BLEU score of output sentence using validation set,after epoch 18 avg BLEU score starts to decrease



RESULTS AND ANALYSIS



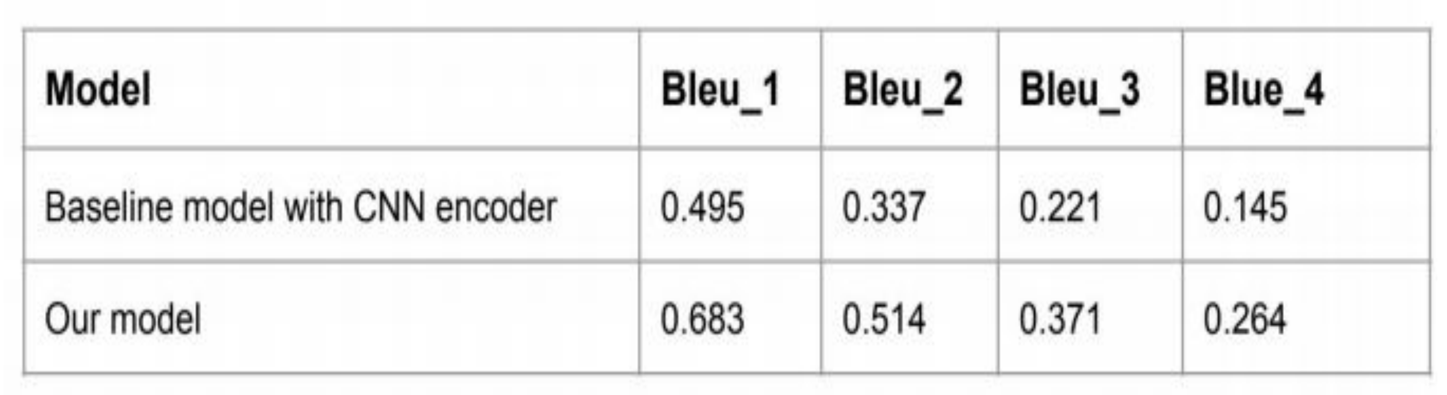
Attention plot

Predicted sentence:
Greedy search:
 a man riding a wave on a surfboard
Beam search: k = 3
 a man riding a surfboard in the ocean
Beam search: k = 8
 a man on a surfboard riding a wave

Reference:
1. A young man riding a wave on a boogie board.
2. A man riding on top of a wave with a surfboard.
3. A young man having fun riding a wave to the shore.
4. A man surfs on his surfboard in the water.
5. A boy is falling off of a surfboard in the water.

Also shown in the left is the output of Mask-RCNN model. We print out the attention matrix at the time our model was generating captions for this image. The output sentence is “a man riding a wave on a surfboard”. From the attention matrix, we can see when the model is generating word “man”, it pays more attention to the first detected object “person1”, and when it is generating the words “wave” and “surfboard”, it pays more attention to the third detected object “surfboard 1”. The model doesn’t pay attention to the person and surfboard in the background. This is a desired behavior, since in all five reference captions none of them mention the other person or the surfboard in the background.

RESULTS AND ANALYSIS



Model	Bleu_1	Bleu_2	Bleu_3	Blue_4
Baseline model with CNN encoder	0.495	0.337	0.221	0.145
Our model	0.683	0.514	0.371	0.264

CONCLUSION

In this project ,we are inspired by Chang’s paper[3] to change the traditional CNN encoder to an object detector encoder. In this way,the input of the model is not the input image itself,but a sequence of objects from image.we also added attention layer so that the decoder uses the right object to generate the right word.

Improvements

1)the Mask -RCNN model we used can only detect 80 kind of objects,we can re-train it to detect more.

2)Number of objects vary from image to image.We can group images having similar number of objects and train in buckets instead of padding all input sequences to have same length.

REFERENCES

1. [1] Vinyals et al. (2014). Show and Tell: A Neural Image Caption Generator. [Online] arXiv: 1411.4555

2. [2] Xu et al. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.[Online] arXiv: 1502.03044

3. [3] Liu et al. (2017). MAT: A Multimodal Attentive Translator for Image Captioning. [Online] arXiv: 1702.05658

4. [4] Bahdanau et al.(2014). Neural Machine Translation by Jointly Learning to Align and Translate. Online] arXiv: 1409.0473

Code available at : https://github.com/manas1999/image-captioning-/tree/master