Assignment 5

Natural Language Processing (CS563)

Department of CSE, IIT Patna

(Read all the instructions carefully and adhere to them.)

Date: 2nd-April-2021 **Deadline**:- 13th-April-2021

Instructions:

1. **A demo tutorial for the encoder-decoder model implementation will be conducted on 3rd-April-2021 (4:00 PM), on the following link:**
   https://meet.google.com/mfb-irxr-wut
2. Markings will be based on the correctness and soundness of the outputs.
3. Marks will be deducted in case of plagiarism.
4. Proper indentation and appropriate comments (if necessary) are mandatory.
5. You should zip all the required files and name the zip file as: **<roll_no>_assignment_<#>.zip, eg. 1501cs11_assignment_01.zip.**
6. Upload your assignment (the zip file) in the following link:
   https://www.dropbox.com/request/ppJbaaYjDuEr1R1XM12t

For any queries regarding this assignment contact:

Zishan Ahmad (zeeman.zishan@gmail.com) and

Deeksha Varshney (deeksha.varshney2695@gmail.com)

Machine translation: MT is a very challenging task that investigates the use of software to translate text or speech from one language to another.

Example: it is raining outside → बाहर वर्षा हो रही है

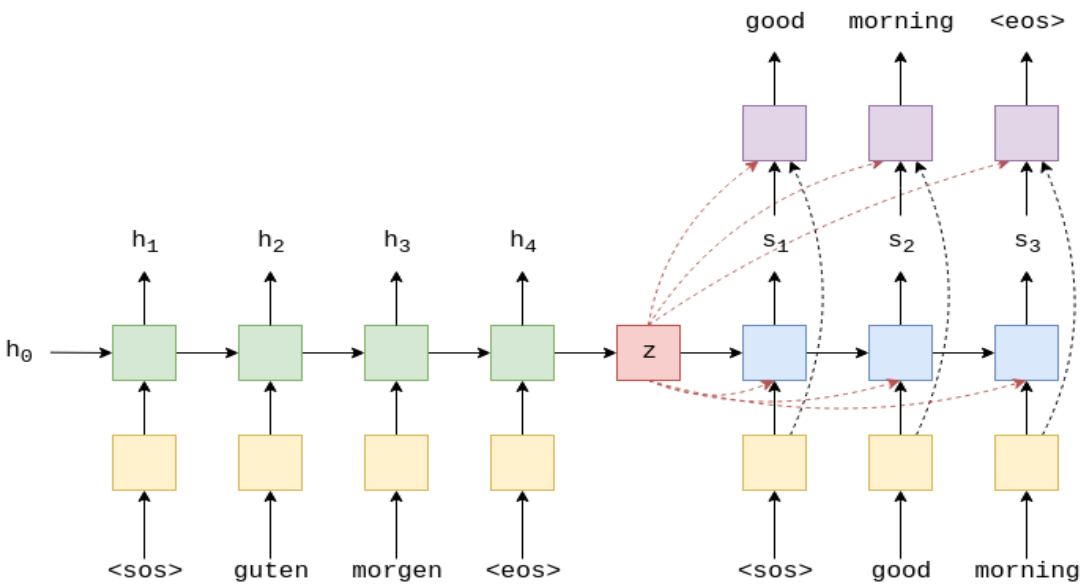
Sequence to Sequence Learning with Attention mechanism for Neural Machine Translation

- **Problem Statement:** The objective is to convert a English sentence to its Hindi counterpart using a Neural Machine Translation (NMT) system using attention.

- **Input:** Given Sentence in English. A start of the sentence (<sos>) and end of the sentence (<eos>) token needs to be appended.
  - ‘<sos>’, ‘it’, ‘is’,’ raining’, ‘outside’, ‘<eos>’
- **Output:** Corresponding translated sentences in Hindi. A start of the sentence (<sos>) and end of the sentence (<eos>) token needs to be appended.
  - ‘<sos>’, ‘बाहर’, ‘वर्षा’,’हो’, ‘रही’, ‘है’ ‘<eos>’

- You may consider the following details for the implementation.
  - Input Vec(Wi input): The word embeddings of the words from the input sentences will be the input to the model. You can use the Word2Vec or GLOVE embedding.
  - Output Vec(Wo Input at the decoder): The word embeddings of the words from the input sentences will be the input to the model. You can use the Word2Vec or GLOVE embedding.
    - Link → Word2vec: http://vectors.nlpl.eu/repository/20/5.zip or https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing
    - Link→ Glove: http://nlp.stanford.edu/data/glove.840B.300d.zip
  - Steps to use pre trained word embeddings:
    - Prepare a dictionary of all the unique words in the dataset.
    - Load the word2vec or glove embeddings.
    - Get embeddings for each word and save them in a numpy or torch matrix.
  - You may use any deep learning libraries such as TensorFlow, PyTorch, Keras etc. for the implementation. Use 300 dimensions for word embeddings.

- **Neural Model:**



- An LSTM based Encoder
- An LSTM based Decoder
- Attention between encoder and decoder as used by [Bahdanau et al.]

- **Dataset:** Download the dataset for Machine translation from here **:** https://drive.google.com/file/d/1jvZxoMsfVDvupZMqTMx11aHmMQFPyWG4/view?usp=sharing

  - There are 3 files consisting of English and Hindi data
  - Use the data in the files ‘english.train.txt’ and ‘hindi.train.txt’ for training. The sentences in the two files are aligned.
  - Test your model using the files ‘english.test.txt’ and `hindi.test.txt’

- **Evaluation Metrics:** Evaluate your model based on the following metrics:
  - BLEU score: BLEU looks at the overlap in the predicted and actual target sequences in terms of their n-grams. (Use the torchtext.data.metrics for computing bleu)
  - Using the gold samples from `hindi.test.txt’ compute the BLEU score.
- **Loss Function :** Use the CrossEntropyLoss function since it calculates both the log softmax as well as the negative log-likelihood for the predicted tokens.