CS563-NLP

ASSIGNMENT-3: Machine Translation

(Read all the instructions carefully and adhere to them)

Date: April 06, 2022 **Deadline**: April 13, 2022
Scores: 20

Instructions:

1. Markings will be based on the correctness and soundness of the outputs.
2. Marks will be deducted in case of plagiarism.
3. Proper indentation and appropriate comments (if necessary) are mandatory.
4. You should zip all the required files and name the zip file as:
5. <roll_no>_assignment_<#>.zip , eg. 1701cs11_assignment_01.zip.
6. Upload your assignment (the zip file) in the following link: https://www.dropbox.com/request/RliXspRdi3pnp6zhYGk7

For any queries regarding this assignment contact:

Aizan Zafar (aizanzafar@gmail.com), Soumitra Ghosh (ghosh.soumitra2@gmail.com)

Machine translation: MT aims at translating from one language to the another.

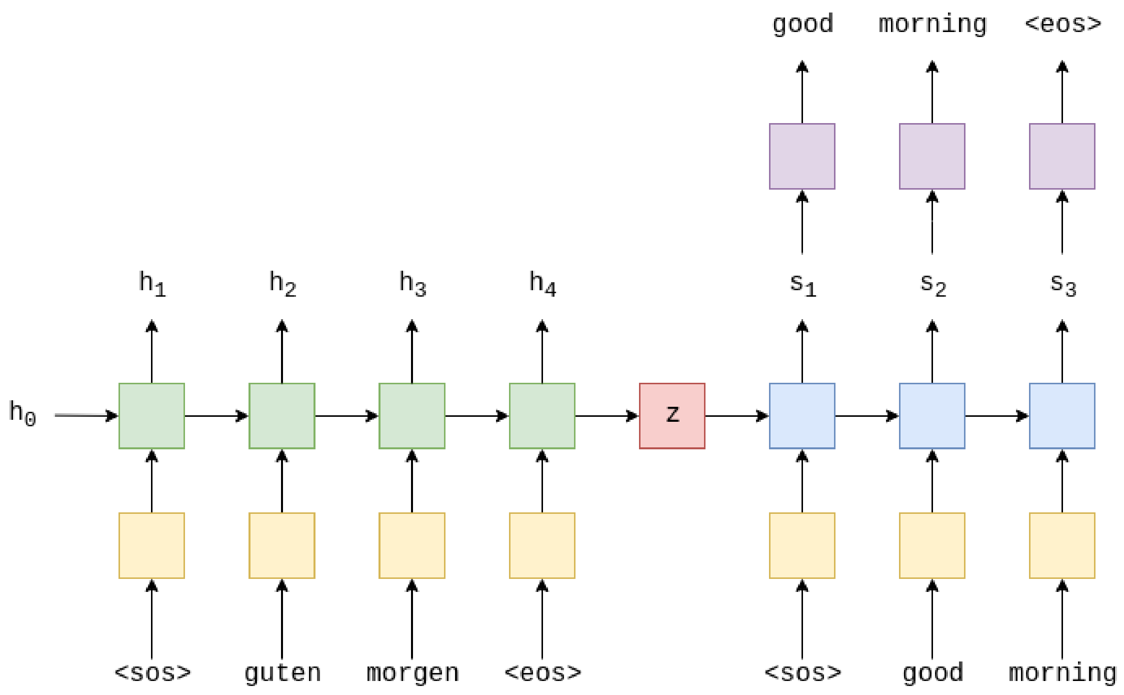
Example: it is raining outside → बाहर वर्षा हो रही है

Sequence to Sequence Learning for Neural Machine Translation

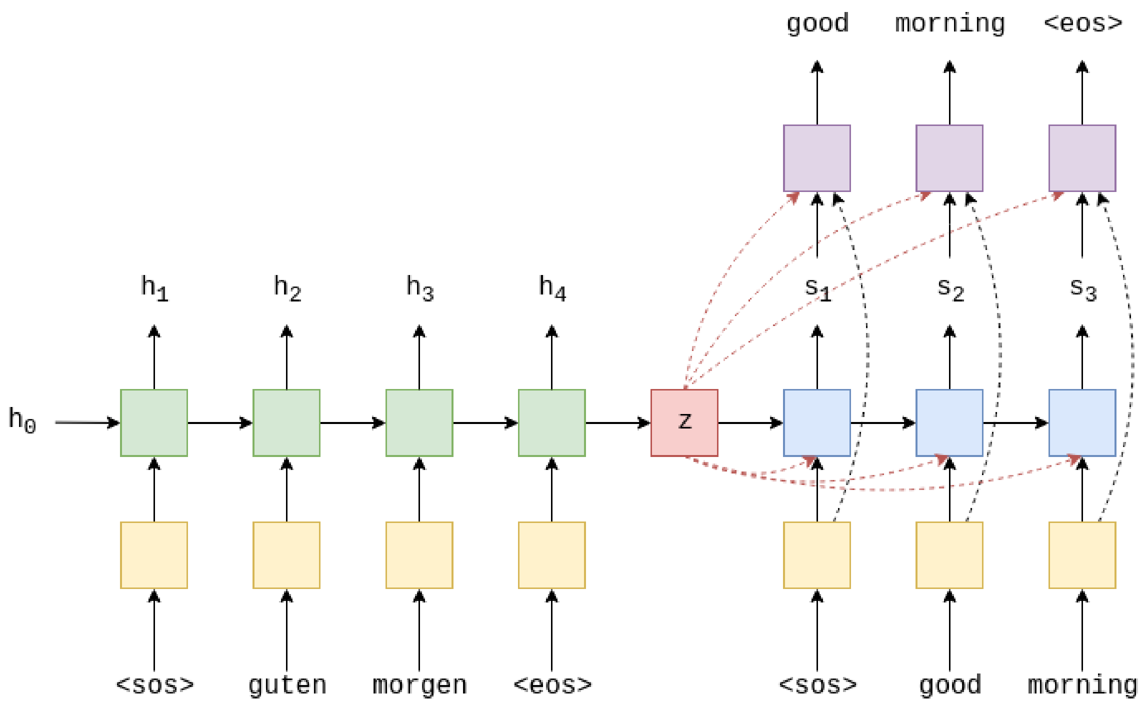
- **Problem Statement:** The objective is to convert an English sentence to its Hindi counterpart using a Neural Machine Translation (NMT) system.
    - **Input:** Given Sentence in English. A start of the sentence (<sos>) and the end of the sentence (<eos>) token needs to be appended.

        - *‘<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 (W_i input at the encoder):** The word embeddings of the words from the input sentences will be the input to the encoder model. You can use the Word2Vec or GLOVE embedding.
    - **Output Vec (W_o input at the decoder):** The word embeddings of the words from the output sentences will be the input to the decoder model. You can use the Word2Vec or GLOVE embedding
        - Link → Word2vec: http://vectors.nlpl.eu/repository/20/5.zip
        - 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 1: Encoder-Decoder with LSTMs**



- **Neural Model 2: Encoder-Decoder with Attention** (Bahdanau attention)



- **Dataset:** Download the dataset for Machine translation from here :

  https://drive.google.com/file/d/1bEK6RCdnXIqg8JGrJIMvDaAM-baalGwt/view?usp=sharing

  - There are 4 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.