Assignment - 2

Due Date: 30th April 2023 **Max.Marks: 12.5**

The goal of this assignment is to experiment machine translation. The aim is to gain a basic understanding of the translation, language models and their implementation.

In Machine Translation, your goal is to convert a sentence from the *source* language (choose one language of your choice) to the *target* language (e.g. English). Implement a sequence-to-sequence (Seq2Seq) network with attention. The following are important points to consider while implementing this system:

(i) Use a bidirectional LSTM based encoder and a unidirectional decoder

(ii) You may choose character or word embeddings as input with *m* as the size of the input $x_1, x_2, \ldots, x_m$ ($x_i \in R^{ex1}$) and *e* as size of the embeddings

(iii) The input is then fed to the bidirectional encoder. The forwards and backwards versions are concatenated to give hidden states h_i^e and cell states c_i^e

(iv) We then initialize the decoder's first hidden state h_0^d and cell state c_0^d with a linear projection of the encoder's final hidden state and final cell state

(v) After initialization of the decoder, feed it a target sentence. On the t^{th} step, look up the embedding for the t^{th} subword, $y_t \in R^{ex1}$ and then concatenate y_t with the *combined-output vector* $o_{t-1} \in R^{hx1}$ from the previous time step to produce $\bar{y}_t \in R^{(exh)x1}$. Note that for the first target subword (i.e. the start token) o_0 is a zero-vector. Then feed $\bar{y}_t$ as input to the decoder.

$$h_t^d, c_t^d = decoder(\bar{y}_t, h_{t-1}^d, c_{t-1}^d), where\ h_t^d, c_t^d \in R^{hx1}$$

(vi) Use h_t^d to compute multiplicative attention over $h_1^e, \ldots, h_m^e$. Use W_{att} as the weight matrix for attention.

(vii) Concatenate the attention output a_t with the decoder hidden state h_t^d and apply a linear layer with *tanh* and a dropout to attain the combined-output vector o_t.

(viii) Produce a probability distribution P_t over target subwords at the t^{th} timestep using softmax

(ix) Finally, to train the network, compute the cross entropy loss between P_t and one-hot vector of the target subword at timestep *t*

Perform training and testing of your model and give BLUE Score. It should be greater than a threshold.

Reference: Kishore Papineni, Salim Roukos, Todd Ward, Wei-jing Zhu: BLEU: a Method for Automatic Evaluation of Machine Translation, (2002)

Implementation

(i) Arrange your code in three code files 1) MT.py – for main machine translation code, 2) embed.py – for embedding related stuff, initialization, etc., and 3) utils.py – for utility functions.

(ii) In a batch, all the sentences should be of the same length, apply padding to achieve it. Write a function in utils.py

(iii) Write a function in embed.py to initialize source and target embeddings

(iv) A function in MT.py to initialize necessary model layer for MT system

(v) An encoder function in MT.py that generates hidden and cell states of the encoder and initial states for decoder

(vi) A decoder function in MT.py which constructs $\bar{y}_t$ for every timestep

(vii) A function step in MT.py applies the decoder's LSTM cell for a single timestep computing the encoding of the target subword h_t^d, attention distribution α_t, attention output a_t, and combined output o_t.

(viii) A function in MT.py that generates meta data (batch size, max source sentence length, masks, etc.) about the MT system

Analyzing MT System

Perform the following or any other innovative analysis over your MT system

- Explore the difference between dot product, additive and multiplicative attention
- Look at the vocab file for some examples of phrases and words in the source language vocabulary. When encoding an input source language sequence into "pieces" in the vocabulary, the tokenizer maps the sequence to a series of vocabulary items, each consisting of one or more characters. Given this information, how could adding a 1D Convolutional layer after the embedding layer and before passing the embeddings into the bidirectional encoder help your MT system.
- Find different errors in the English translation obtained by your MT system and gold English translation. Provide possible reason(s) why the model may have made the error (either due to a specific linguistic construct or a specific model limitation)
    - Implement one possible way to alter the MT system to fix the observed error. There can be more than one possible fixes for an error. For example, it could be tweaking the size of the hidden layers or changing the attention mechanism.

Datasets: Datasets can be taken from NLTK/WMT2014/2016/etc., etc.

Report: Submit a report with the detailed methodology (with all the equations and shapes) and results with ablation study on or before the due date.

Deliverables: Report, implementation and sample dataset (id1_id2. zip)