Question Paraphrasing system

Report By: Prakhar Gurawa

Task Description: To create a question paraphrasing system using the concepts of NLP and machine learning. For example:
f(“who won the Euro contest in 1999?”) → “In 1999, the Euro contest was won by whom?”
Where f is function trained using the provided dataset with a record of 1118 questions and there paraphrases

Approaches toward problem:

Approach 1: Inspired by neural machine translations using the encoder-decoder system, applied the architecture on this problem set where I used LSTM cells in both the encoder and decoder side. I even tried to extend the model by using stacked LSTM layers inspired by [1] on both the side but the results were meaningless. The word embedding used is Glove word embeddings, which help to represent words as a low dimensional embedding. The optimizer used is RMSProp, with loss as categorical cross-entropy loss. Padding is done based on the maximum length sequence found both on the input and output sides.
The basic pipeline for this approach is as described below:

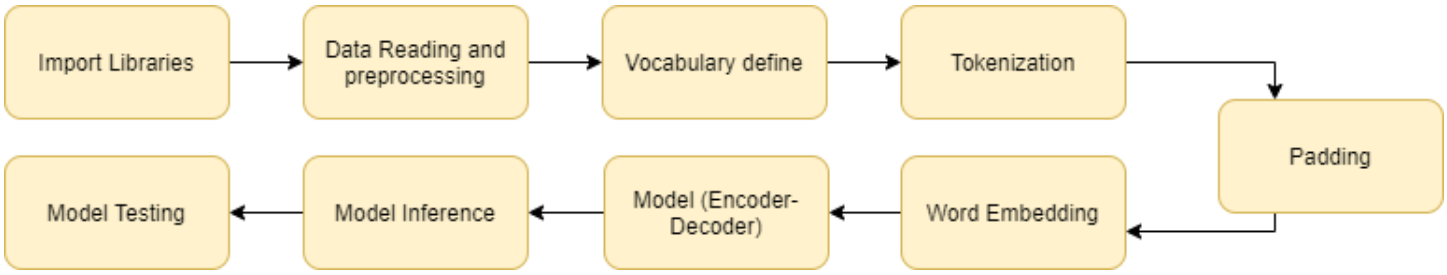

Figure 1. Pipeline for question paraphrasing system using NMT

Both the approach with stacked LSTM layers and single LSTM layers were providing unsatisfactory results such as for the sentence “What kind of glass exists in nature?” it provided “What is the university of force of Chicago?”. This problem has been already stated in some papers like [2].

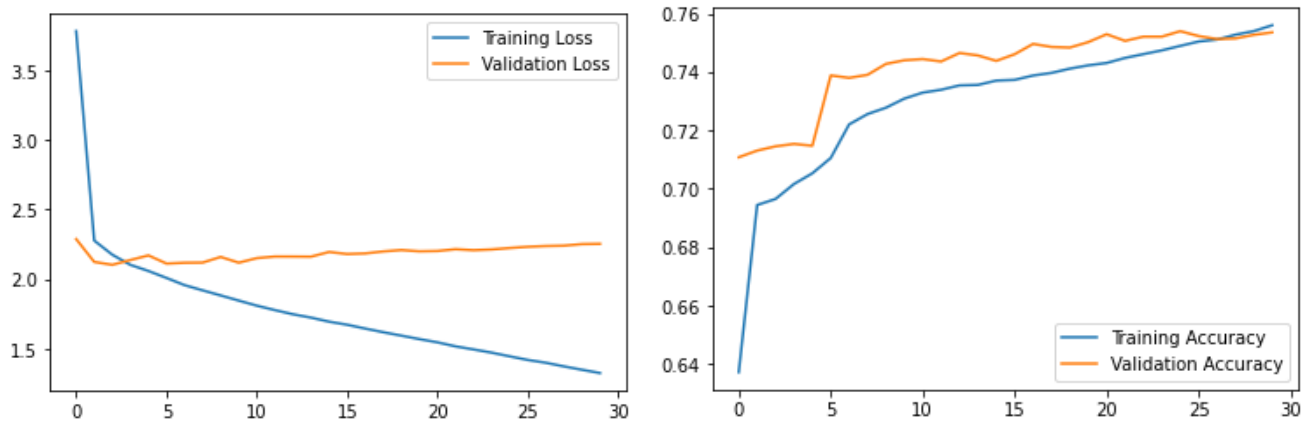

Figure 2. Performance of se2seq stacked LSTM model for 30 epochs

The encoder model is provided with an input vector which provides the state variables as its output which act as input for the decoder model. Also, LSTM cells are used here to learn better for larger sequences and deal with gradient issues.

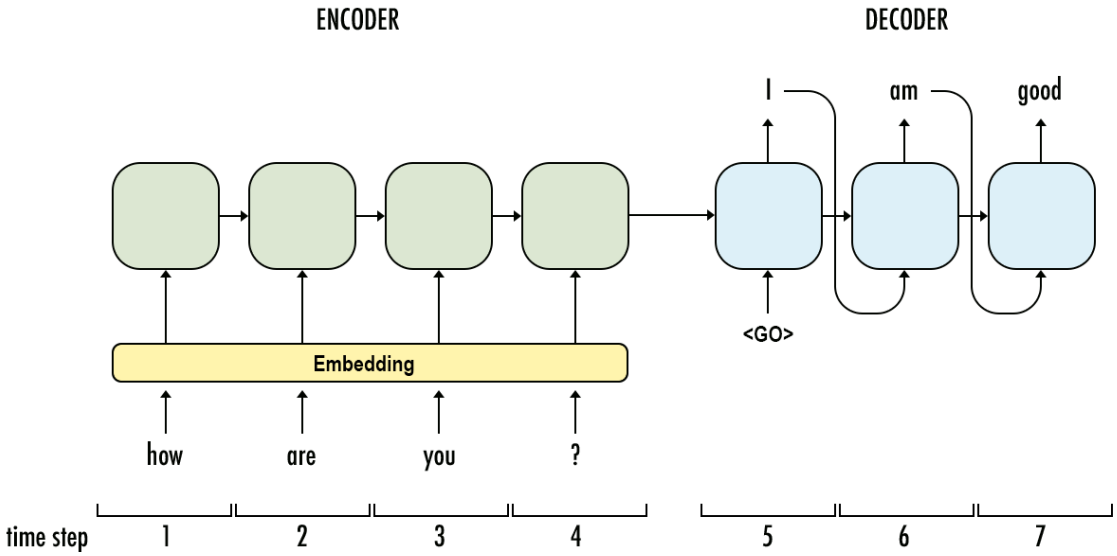

Figure 3. A basic representation of seq to seq encoder-decoder model

Approach 2: Due to the complexity of the problem and the unavailability of a larger corpus I shifted my focus to pretrained powerful models. I used BART [3], which is a denoising autoencoder sequence to sequence transformer model which includes BERT (only used the encoder) and GPT (only uses the decoder). The encoder uses a denoising objective similar to BERT while the decoder attempts to reproduce the original sequence. Performed data cleaning, hyperparameter setting, use of Huggigface's open-source library "simpletransformers" and then running the model for 2 epochs. The results for input questions and paraphrased questions were almost similar such as "What sea was created by the Alps?" to "WhatWhat sea was created by the Alps?". Some potential problems with BART have been already stated as follows:

- The generated sequence is almost identical to the original with only minor differences in a word or two.
- Incorrect or awkward grammar.
- Might not be as good on out-of-domain (from training data) inputs.

Approach 3: Opted for a more powerful pretrained model Google's T5 [4] model which is also a text-to-text encoder-decoder model that works well on out-of-box vocab trained on datasets like C4, RealNews-Like, etc. For the first model, I split the training set in 80:20 ratio as train and validation set which gave me Harmonic Mean: 0.085 (1) BLUE: 0.104 (2) PINC: 0.322 (1). As the dataset was small I split the train set as 95:5 ratio which resulted in Harmonic Mean: 0.104 (1) BLUE: 0.113 (2) PINC: 0.378 (1). Finally with some minor tweaks in hyperparameters as defining the input length as 100, learning rate, train batch size, eval batch size, number of epochs I got a final result as :

Harmonic Mean: 0.108 , BLUE: 0.127 and PINC: 0.327

Overall transfer learning technique on this powerful model gave quite good results and the results looked very real and organic for example:

Was the landing successful? → Is the landing successful?

Which two Portuguese wines are especially enjoyed around the world? → Which two Portuguese wines are extremely popular around the world?

Github Link/Code: https://github.com/prakhargurawa/Question-Paraphrasing-System
Model3: Encoder-Decoder from scratch system
Model4: BART based system
Model1, Model2, Model5: T5 based system where model 1 is a simple pretrained model and model 2 and 5 retrained on provided dataset

Insights: Complex pretrained models will mostly provide much better results especially on NLP tasks as it requires much more data than normal machine learning programs. Data cleaning, data preprocessing and hyperparameter tuning can create a huge impact on the accuracy and efficiency of models. Representing a word as an embedding is in most cases a better choice than representing them as a single integer as single integers can't represent the relationship between the words. Adding stacks of layers will never guarantee a better model as my stacked LSTM encoder-decoder was performing slightly lower than the lower one. A much robust model can be created by extending the given dataset by using other paraphrasing datasets like the Quora question pair dataset [6] and Google's PAWS dataset [7]. Transfer learning is and will surely make a crucial impact in the field of Natural language processing with the more flexible, robust and powerful model coming every year.

References :

1. Neural Paraphrase Generation with Stacked Residual LSTM Networks by Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri2: https://arxiv.org/pdf/1610.03098.pdf
2. A Theoretical Analysis of the Repetition Problem in Text Generation by Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi: https://arxiv.org/pdf/2012.14660.pdf
3. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer: https://arxiv.org/pdf/1910.13461.pdf
4. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer by Colin Raffel, Noam Shazier, Adam Roberts, Katherine Lee, and Sharan Narang: https://arxiv.org/pdf/1910.10683.pdf
5. SQuAD: 100,000+ Questions for Machine Comprehension of Text by Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang: https://arxiv.org/pdf/1606.05250.pdf
6. https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs
7. https://github.com/google-research-datasets/paws#paws-wiki

Other References/Tutorials for research purposes:

1. https://www.reddit.com/r/MachineLearning/comments/5x1l8e/d_what_is_the_state_of_the_art_in_paraphrase/
2. https://towardsdatascience.com/t5-text-to-text-transfer-transformer-643f89e8905e
3. https://medium.com/dataseries/text-to-text-transfer-transformer-e35dc28bae14
4. https://towardsdatascience.com/paraphrase-any-question-with-t5-text-to-text-transfer-transformer-pretrained-model-and-cbb9e35f1555
5. https://huggingface.co/mrm8488/t5-small-finetuned-quora-for-paraphrasing
6. https://www.kaggle.com/c/quora-question-pairs/code
7. http://jalammar.github.io/illustrated-bert/
8. https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

9. https://huggingface.co/transformers/training.html
10. https://medium.com/@aniruddha.choudhury94/part-2-bert-fine-tuning-tutorial-with-pytorch-for-text-classification-on-the-corpus-of-linguistic-18057ce330e1
11. https://www.tensorflow.org/tutorials/text/nmt_with_attention
12. https://www.youtube.com/watch?v=W2rWgXJBZhU
13. https://www.youtube.com/watch?v=TQQlZhbC5ps
14. https://github.com/kingchloexx/GPT2-Question-Answering/blob/master/paper.md
15. https://towardsdatascience.com/natural-language-generation-part-2-gpt-2-and-huggingface-f3acb35bc86a
16. https://gist.github.com/GeorgeDittmar/5c57a35332b2b5818e51618af7953351
17. https://github.com/topics/paraphrase-generation
18. https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/
19. https://github.com/farizrahman4u/seq2seq
20. https://github.com/tlatkowski/multihead-siamese-nets
21. https://www.researchgate.net/publication/336996372_Transformer_and_seq2seq_model_for_Paraphrase_Generation
22. https://tianjun.me/static/essay_resources/Paraphrase_Generation/main.html
23. https://towardsdatascience.com/finding-similar-quora-questions-with-word2vec-and-xgboost-1a19ad272c0d
24. https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html
25. https://www.pragnakalp.com/nlp-tutorial-setup-question-answering-system-bert-squad-colab-tpu/