Abstractive Text Summarization

Yalamaddi Abhinav | Dr. D. Sumathi | Scope

Motivation/ Introduction

Abstractive summary is an effective kind of summary in comparison with extractive summary, because it collects information from several texts in order to construct an accurate summary. This has becoming more popular since new sentences may be developed to convey vital information from text documents. In a consistent and linguistically accurate form, an abstractive summarizer delivers summarized information. Readability or language quality is a vital driver to improve a summary's quality.

Methodology

As stated, before this project is implemented using 3 models as a stacked layer.

Model 1

The first model was a basic one-way LSTM encoder decoder with randomly initialized words. As mentioned later, discovered that using preexisting word embedding would be better, and hence iterated Model 2 using this model.

Model 2

The model with a two-way LSTM encoder and a unidirectional LSTM decoder has been implemented.

Model 3

Implementation of a two-way LSTM encoder in the last model and incorporated the LSTM decoder worldwide attention. Previously provided attention scoring functions. The final secret state (forward and reverse) of the encoder is linked and utilized as the initial hidden state of the decoder.

In Model 1, the Word Embedding is randomly initialized and updated by the model. Here, I believed that the word vector representations learnt would be more synthesizable than previously trained Word2Vec or Glove vectors. Soon became aware, however, that machine and time restrictions would impede the achievement of this goal, as many iterations on enormous data sets would be required to generate exact word embedding for this purpose. Since loaded Glove vectors for the embedding matrix for model 2 and model 3 because training in words on a large dataset was not a possibility. Utilized the same embedding and vocabulary both for encoder and decoder to ease implementation. But alternative models such as that provided by Rush et al. (2015) train several embedding matrices for various functionalities in the context in which a word appears and therefore enhance the notion of semantics of a vector.

Attention networks

The use of attention for encoder-decoder neural networks allows a context vector to be established at any given time, given the current hidden state of the decoder and a subset of hidden states of the encoder. The context vector is conditioned on all the hidden states of the encoder for global attention, while a strict subset of the hidden states of the encoder is used by local attention. We employed two separate scoring functions for our implementation of Model 3, which are used as weights for averaging the hidden conditions to build the context vector.

Handling multiple input lengths

Since input and output sequence lengths vary, all inputs have been standardized for a maximum length by padding short phrases and cutting large phrases. In both the score and loss functions this padding was taken into consideration, which prevents the model from refreshing parameters that cannot be modified.

Drawbacks

Our greatest drawback has been the insufficient time after our careful encode decoder RNN model was implemented. In the absence of a better description, abstract summarization is a chaotic process — training and truth data are not well coordinated in comparison to Model 1 Loss, which means that machine translation is almost one-to-one. Therefore, in order to translate input text to summary, the model must learn to interpret nuances. Models that have succeeded in this job often train a lot of workable parameters for days (i.e., from a large vocabulary size, or large hidden state size). As an example, it has taken the 8-day Tesla K40 GPUs to train IBM Watson. Since we wanted to continuously iterate to adjust hyperparameters, it was challenging to train the model for an enough time to achieve meaningful results.

Results

Despite succeeded to build the RNN summarizer encoder-decoder uas a baseline in the model, the lack of enough time for training, trained the Model 3 on poor data for the most part of its training and faced many problems with the Collab GPU. On 100,000 sentence-headline combinations trained Model 3 for 10 times the first time. Used around 200 encoders and decoder cells, 200 Glove vectors and a vocabulary of 5,000. The loss was reduced from 8,667 to 6,584 but found that the model had started producing entirely UNK tokens, for maximum length of output after evaluating the development projections. Later examined the data used to train the model and the manner through which initialized the vocabulary in the model to detect the problem.

Used the most common V-words from the full pre-processed dataset to generate an embedding matrix in order to begin with a vocabulary of size V. In the whole Unannotated English Giga word Corpus although the V = 10.000 most popular words are still significantly fewer than the number of singular words on the initial 100.000 examples. Many of the truth values have therefore been expressed as UNK tokens. Thus, UNK was the widest word in truth values and the model has been able to reduce loss by repeated output of UNK.

Without enough time to use a big dataset to generalize the model to unnoticed information, decided to train the attention-grabber RNN model on a short dataset where the model's vocabulary is the same as the unique terms in training and development sets. This eliminates the problem of repeated imprints of the UNK token and even prints meaningful words, however the results were only examined at the 80th out of 600s.

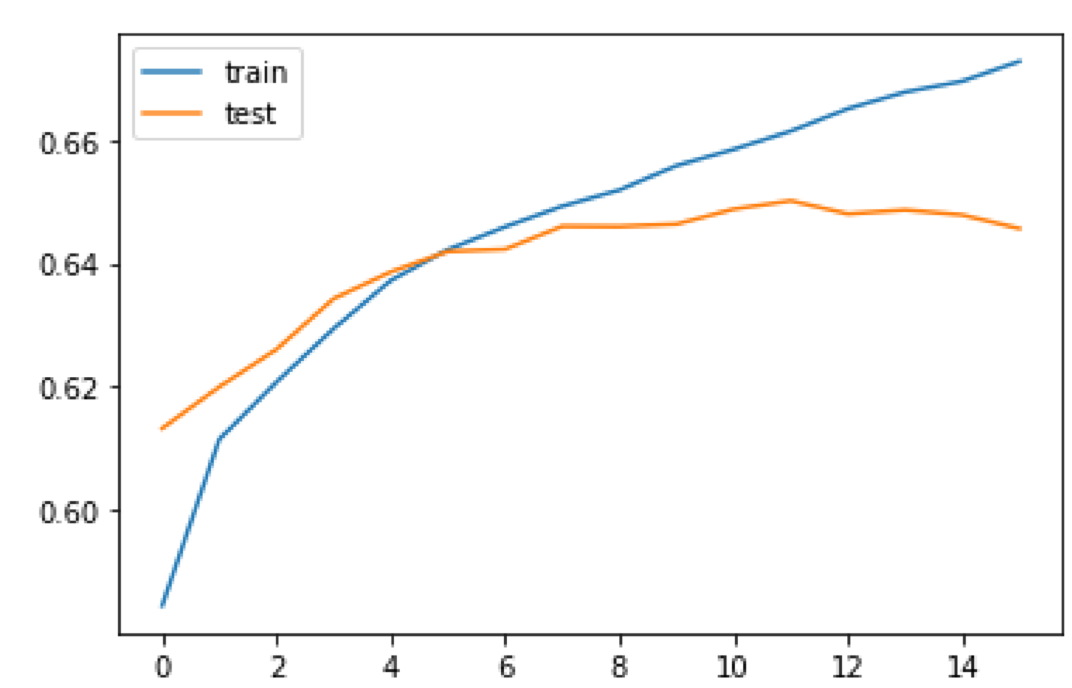

Fig.1 Algorithm accuracy Graph

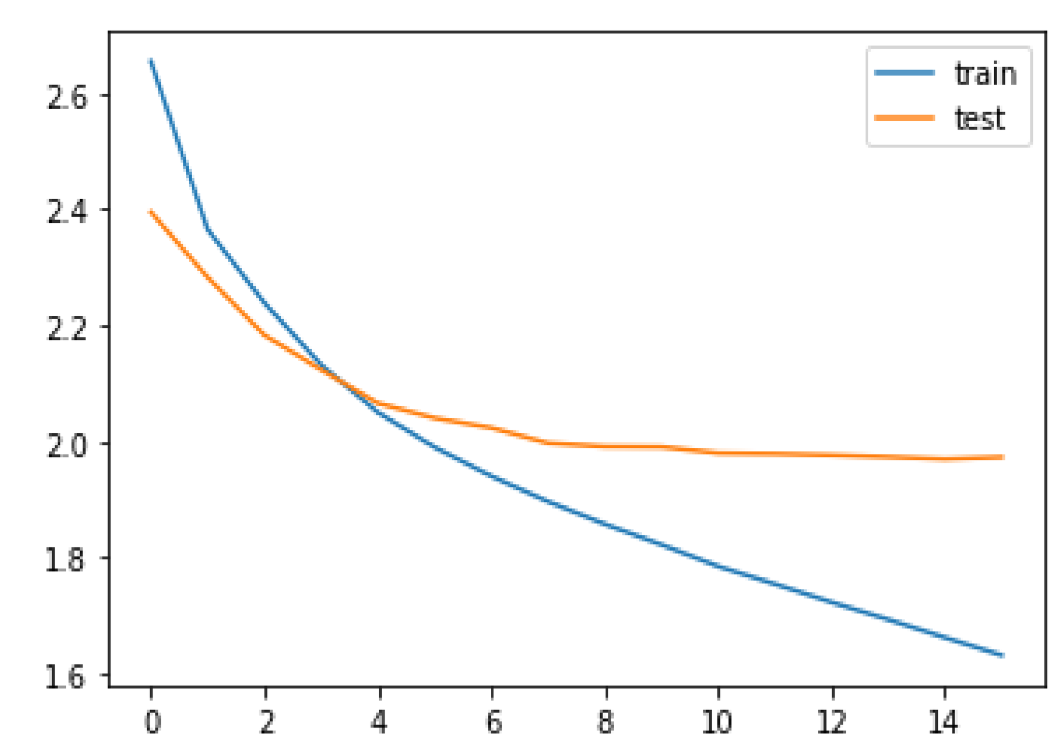

Fig.2 Algorithm loss Graph

Conclusion

Unfortunately, I couldn't get as far as my guide wanted, but now I know how important it is to give time for all the variables – as I showed above, I seem to be heading in the right direction through 80 times out of 600 with the final version, with the very small data set and a much more substantial number of epochs. The project is optimistic that if we run the final model across the epochs of this data set (this will happen shortly after this submission), a meaningful summary will work much better.

Contact Details

abhi.yalamaddi@gmail.com

Acknowledgments

It is my pleasure to express with deep sense of gratitude to Dr. D. Sumathi, Assistant Professor, SCOPE, VIT-AP, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor.