Sequence-to-sequence Architecture Using BERT

Claudio Scheer and **José Fernando Possebon**
Pontifical Catholic University of Rio Grande do Sul - PUCRS
claudio.scheer@edu.pucrs.br, jose.possebon@edu.pucrs.br

Abstract

Abstract.

Introduction.

Related works

Related works.

Deep Learning

In this section, we will discuss the sequence-to-sequence model using recurrent neural networks and transformers. Also, in a nutshell, we discuss how a BERT model works.

Sequence-to-sequence

The encoder-decoder architecture was initially proposed by (Cho et al. 2014). Although simple, the idea is powerful: use a recurrent neural network to encode the input data and a recurrent neural network to decode the encoded input into the desirable output. Two neural networks are trained.

(Graves 2013) - Generating sequences with LSTM

(Bahdanau, Cho, and Bengio 2015) - Proposed attention

(Vaswani et al. 2017) - Attention is all you need

BERT

(Devlin et al. 2018) - BERT

Similarly to the original sequence-to-sequence model using a recurrent neural network, the model discussed in this paper uses two BERT neural network: one neural network to encode the input and another to decode the input encoded.

Dataset

As we focused our project on automatic email reply, we used The Enron Email Dataset[1] to train our model. The dataset contains only the raw data of the emails. Therefore, we created a parser[2] to extract the email and the replies from each email.

To identify whether an email has a reply or not, we look for emails that contain the string `-----Original Message-----`. After filtering only emails with non-empty replies, we parse those emails in an input sequence (the original email) and in the target sequence (the reply email). The entire extraction was done automatically, that is, we did not manually extract or adjust any email.

We used two libraries to parse the dataset: `talon`[3], provided by Mailgun, and `email`, provided by Python. The `email` package returns the email body with the entire thread. To extract only the last reply from an email thread, we use the `talon` package.

The original dataset contains 517,401 raw emails. After parsing the raw dataset, we created a dataset with 110,205 input and target pairs.

In the parsed dataset, we have 8,368 pairs with specifics email and reply patterns. Since these pairs do not represent a large part of the dataset, we trained the dataset with this "wrong" data.

References

[Bahdanau, Cho, and Bengio 2015] Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In Bengio, Y., and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[Cho et al. 2014] Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078.

[Devlin et al. 2018] Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

[Graves 2013] Graves, A. 2013. Generating sequences with recurrent neural networks. *CoRR* abs/1308.0850.

[Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *CoRR* abs/1706.03762.

[1] https://www.kaggle.com/wcukierski/enron-email-dataset

[2] https://www.kaggle.com/claudioscheer/extract-reply-emails

[3] https://github.com/mailgun/talon