Saarland University

Department of Computational Linguistics

Software Project: **Neural Network**

Multi-turn Architecture for Neural Chatbot

Author:
Chirag Bhuvaneshwara
Matriculation: 2571703

Supervisor:
Prof. Dr. Dietrich Klakow

08/04/2019

Abstract

In the modern world, a lot of textual data is generated on a daily basis. All of this data can be used to train a deep learning model to respond to queries posed. While many such concepts have already been implemented to respond to the current query, the way the networks are trained is not similar to how humans converse in the real world. In this project, we have developed a new network architecture that will ensure that the network answers the current query not only by considering the current query but also by considering certain important aspects of the previous parts of the conversations.

Contents

1 Introduction

In today's world of abundant data, it is not possible for just humans to process all textual data being generated. So it becomes necessary to develop computer algorithms that can process large amounts of textual data and based on it, generate meaningful responses. To this end, several information retrieval based systems have been developed to process an input question and generate an answer. These methods use a similarity measure to match the input question to the questions present in the database and output the answer associated with the closest matching question. Several such simple methods are discussed in Reference 1.

Recent advances in deep learning have led to development of information retrieval based systems where a number of retrieved answers are combined using deep neural networks to generate a more meaningful and unique response to the question as described in Reference 2. But in order to make full use of the power of deep learning, it is better to train a neural network to generate a response for a particular query without using a database of query-response pairs. Sutskever et al show exactly this in Reference 3 where they train a neural network to map an input sentence of variable length to an output sentence of variable length by using an encoder - decoder architecture. This allows the neural network to produce more relevant and unique answers.

But the above method does not perform well when trying to process long sentences because the Long short-term memory (LSTM) or Gated recurrent units (GRUs) that form the encoder will not be able to pinpoint that certain words present in the beginning of the query sentence are important. This is solved by Bahdanau et al in Reference 4 where it is shown that a new mechanism called attention can be used to find out which time steps of the encoder outputs are important by assigning attention weights to each of them. The attention weights can be computed in several ways and one way would be to use a simple 1 layer neural network. This method was improved by Luong et al in Reference 5 which introduced the concept of Local and Global attention and introduced three new ways of applying the attention weights.

But real life conversations are not just based on the current query. In a two person real world conversation, people tend to remember what they have previously said and also what the other person has previously said. When there is a new query in the conversation, people consider all the previous aspects of the conversation and the current query to produce an answer. In this project, I have developed a way to train a neural network such that it considers not only the current query, but also aspects of the previous conversations to generate the final answer. The following sections will present the single-turn architecture used, the new multi-turn architecture to process a conversation, practical limitations faced and summarise why such a neural network is beneficial.

2 Single-turn Architecture

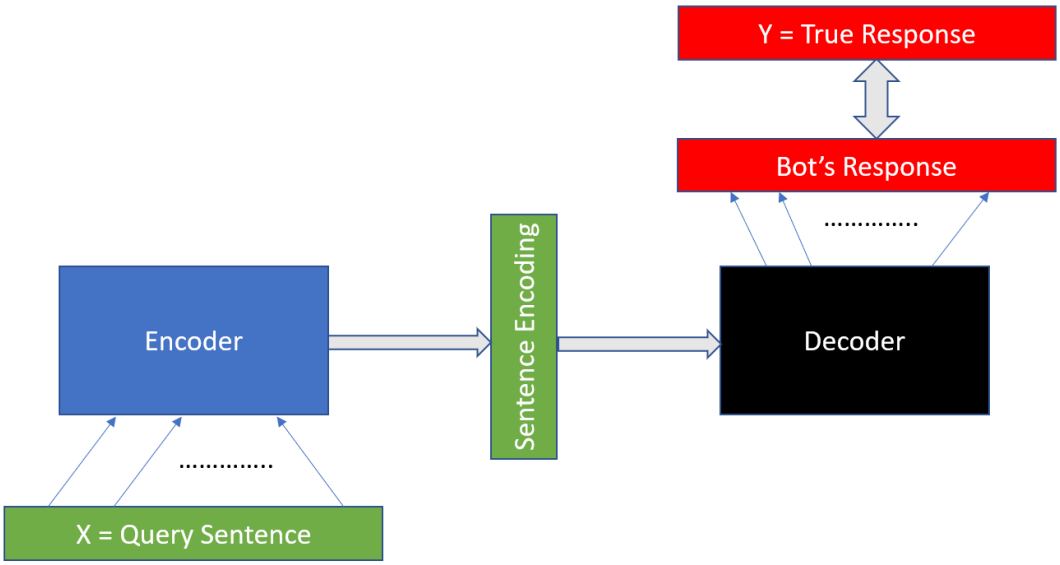

Figure 1. The Single-turn architecture.

A typical sequence to sequence architecture as described by Suteskever et al in Reference 3 is the basis for the Single-turn architecture used in this project. To make this architecture work, the vocabulary from the corpus is converted into word embeddings and fed into the model. To encourage the neural network to learn to process and produce variable length sentences, we make use of the PAD token. Also, every sentence is marked with a specific Start of Sentence (SOS) token and an End of Sentence (EOS) token.

The network architecture shown in Figure 1 consists of an encoder and a decoder. The encoder is an LSTM or GRU. At each time step of the encoder, one word embedding is fed in and upon encountering the EOS token, it'll be considered as the last time step. Thus, the EOS token is included in the training of the network and the trained network can produce EOS tokens. At each time step, the hidden state is passed along to the next time step. At the end of the input query sentence, the last time step of the encoder produces a final hidden state which is unique for each query sentence. This final hidden state is the Sentecne Encoding.

The problem with using just LSTM or GRU is that certain important words in the initial part of the query sentence might not be contributing much to the final Sentence Encoding. Due to this, we apply attention mechanism and figure out which words in the sentence are actually important. This is done by using Luong Global Attention mechanism as described in Reference 5 where attention weights are applied to each time step of the encoder outputs and the Sentence Encoding is improved.

The Sentence Encoding is passed as the initial hidden state to the Decoder network which

is also an LSTM or GRU network. With the attention mechanism in place, the decoder generates a response by generating one output word at each time step. The decoder ends its output when an EOS token is generated.

In our case, the Encoder and Decoder are 2 hidden layers each and Single-turn architecture described above is trained by feeding in the query sentence from the dataset as the input and is expected to learn the response sentence from the dataset as the output. The dataset used is the Cornell Movie dataset which was processed to the required format by our teammate Hui-Syuan Yeh.

3 Multi-turn Architecture

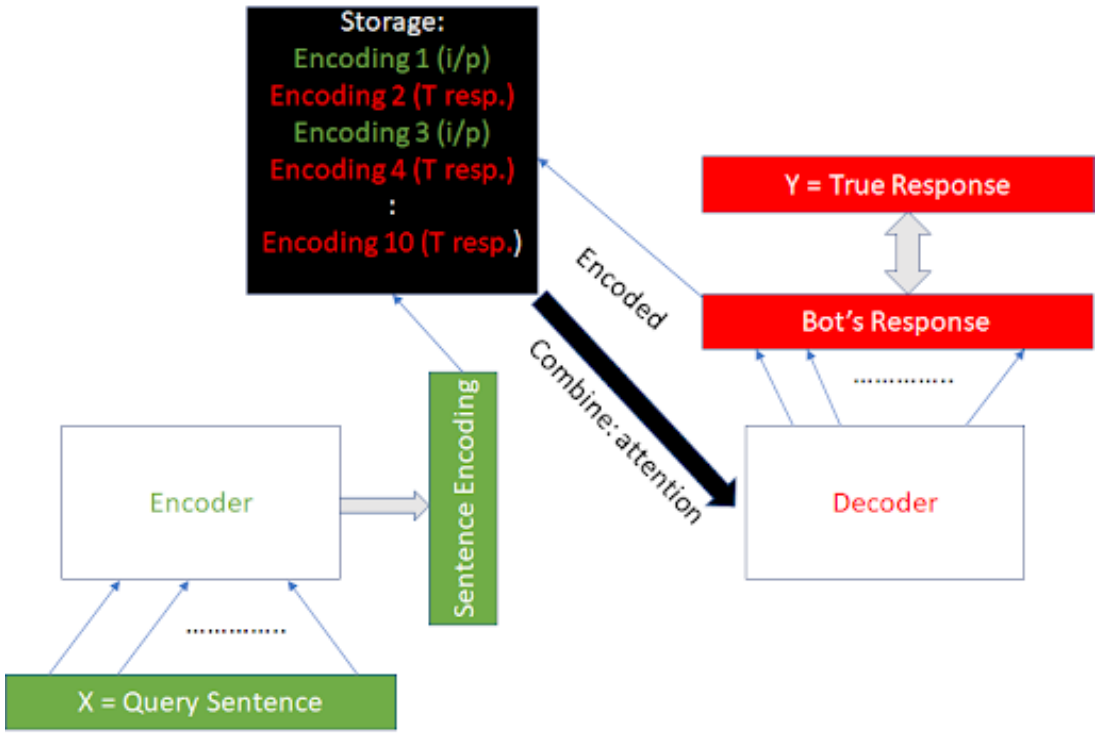

Figure 2. The Multi-turn architecture.

The drawback of using just the Single-turn architecture is that it is not modelled in the way that a normal human conversation takes place. In a normal human conversation, multiple aspects of the already concluded conversation as well as the current query are considered to respond to the current query. In order to train our neural network to consider both of the above, we would need our neural network to process the data conversation by conversation instead of sentence by sentence in the Single-turn architecture.

In order to process the data conversation by conversation, the input data had to be restructured. This was done by Hui-Syuan Yeh, who structured the data in terms of

conversations of varying lengths. This also introduced a new token called the End of Conversation (EOC) token to mark the end of a conversation. This input data is then fed into the new Multi-turn architecture shown in Figure 2.

The difference between the Single-turn architecture and the Multi-turn architecture is that the Sentence Encoding from the Encoder is not just directly passed into the decoder. To enable multi-turn conversation, we store the sentence encodings of all the sentences that form one conversation. To do this, we consider only the conversations consisting of two people and each conversation consists of single sentences. If any of the sentences in the conversation are greater than a certain maximum length (in our case 10), the whole conversation is not considered as a part of the input data. Additionally, the conversations always consist of even number of interactions between the two subjects involved in the conversation because we need our network to produce some intelligible answer.

In order to store the Sentence Encodings in the Sotrage Variable shown in Figure 2, we had to obtain the Sentence Encoding of both the query sentence and the chatbot's response sentence. While it is possible to use the final decoder hidden state as the Sentence Encoding for the chatbot's response, I chose a different solution. The chatbot's response is first decoded and then run through the same encoder that is used on the query sentence.

The sentence encodings of all the query and responses in the conversation that are present in the storage variable are then combined into one encoding for the conversation and passed into the decoder. The following two subsections explain how they are combined. But when a conversation is initialized, the storage variable is empty and the training procedure is similar to what it was for the Single-turn architecture.

As the training procedure is in terms of conversations, the storage variable is emptied upon encountering an End of Conversation (EOC) token as the query sentence at the Encoder.

3.1 Attention Multi-turn Architecture

All the sentence encodings belonging to one conversation present in the storage variable need to be combined into one sentence encoding which ideally figures out which parts of the concluded conversation are important to answer the current query. This can be figured out by using the Global Attention mechanism described in Reference 5 by Luong et al. In my implementation, I find the attention weights between the sentence encodings in the storage variable and the current decoder output by using the dot score explained in Reference 5. We then apply the attention weights to the sentence encodings by means of a dot product. This sum is then added to the sentence encoding of the current query sentence to get the final sentence encoding for the conversation. This encoding is then

passed into the decoder as the initial hidden state.

3.2 Simple Multi-turn Architecture

Another architecture without attention was developed for the sake of comparison. In this simple architecture, all the sentence encodings are multiplied with a constant weight of 0.1 and added together. This sum is then added to the sentence encoding of the current query sentence to obtain the final encoding which is passed to the decoder as before.

4 Evaluation

The evaluation of the results obtained from the trained chatbot model are shown in Figure 3 which also consider the improvements made by our teammates Christine Schfer and Shahzain Mehboob. A detailed evaluation can be found in the evaluation report of our teammate Sourav Dutta.

	Single-turn GREEDY	Multi-turn GREEDY	Multi-turn IDF-BEAM
BLEU (1-gram)	30.32 %	50.00 %	**56.42 %**
BLEU (2-grams)	**21.82 %**	0 %	19.67 %
BLEU (> 2-grams)	0 %	0 %	**5.23 %**

Figure 3. The evaluation results in terms of BLEU scores.

5 Practical Limitations and Future Work

While implementing the multi-turn architecture, I faced some practical limitations. For the multi-turn architecture we need to feed the data conversation by conversation. For the way, I structured the multi-turn architecture, it was not possible to achieve a batch size greater than one. This is because a batch size greater than one would lead to parallel processing of the query sentences. If all the query sentences in the conversation are parallely processed then the structure of the conversation will not be followed. This would lead to the final sentence encoding obtained after combining the encodings in the storage variable to lead to a different meaning. Due to this reason, I had to set the

batch size to 1. But having such a small batch size greatly increased the time taken to process the entire dataset. Increasing the batch size and reducing the training time can be considered as grounds for future work.

6 Conclusion

In this project, I implemented a new multi-turn architecture which can ideally figure out which sentences of the already concluded conversation are important to answer the current query. The new architecture takes into account the structure of natural human conversations.The evaluation result shows that the multi-turn architecture has potential to improve the performance of neural chatbots. But the practical limitation of having set the batch size to 1 did not allow us to see the full potential of the multi-turn architecture. In a future work, this can be improved upon to show the full power of this method of training a neural chatbot.

7 References

1. Martinez, David MacKinlay, Andrew Molla Aliod, Diego Cavedon, Lawrence Verspoor, Karin. (2012). Simple similarity-based question answering strategies for biomedical text. CEUR Workshop Proceedings. 1178.

2. Dong, Li Wei, Furu Zhou, Ming Xu, Ke. (2015). Question Answering over Freebase with Multi-Column Convolutional Neural Networks. 260-269. 10.3115/v1/P15-1026.

3. Sutskever, Ilya Vinyals, Oriol V. Le, Quoc. (2014). Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems. 4.

4. Bahdanau, Dzmitry, Kyunghyun Cho and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. CoRR abs/1409.0473 (2015): n. pag.

5. Luong, Minh-Thang, Pham, Hieu and Manning, Christopher D. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015): .