Universität Hamburg
Department Informatik

A Deep Learning Chatbot

Seminar Paper

Neural Networks

Ilay Koksal
Matr.Nr. 7016062
7koeksal@informatik.uni-hamburg.de

12.07.2018

Abstract

This paper aims to create a chatbot with specific behavior which uses deep learning methods. The main goal of the implementation is creating natural dialogues with the chatbot. To realize the chatbot, the sequence to sequence model used. The sequence to sequence model is frequently used for natural language applications like translation or language generation and it will be explained briefly in the following sections. The created model uses Reddit comments as training data. To specify the behavior of the chatbot, only selected subreddit comments fed into the network.

Contents

1 Introduction

Over the past decade, Deep Neural Networks (DNN) became more and more common among researchers from various fields. From speech recognition [3] and image classification to natural language applications, deep learning successfully proved itself in a wide range of tasks. Recently, it started to be used in natural language applications like language modeling or paraphrase detection.

However, using simple DNNs for language tasks generates the problem of mapping sequences with dimensions not known beforehand. This is an important problem for many sequential tasks like speech recognition or conversation generation [6].

In this paper one of the widely-used deep learning models, Sequence to Sequence Model, will be used to create a chatbot with some personal traits we decided beforehand. This model was first suggested by Sutskever et al. [6] and now it is mostly used in Natural Language related tasks like translation and conversation generation. This model is also referred as Recurrent Neural Network (RNN) Encoder-Decoder. As the name suggests, the model consists of 2 Long Short Term Memory (LSTM) RNNs where one acts as an encoder and the other one as a decoder [7].

On top of Natural Language Tasks, Sequence to Sequence Model can work on any sequential data without actually understanding the information behind the data. For example, after training network with any given paired questions and answers, network would be able to answer questions without checking any knowledge base [8].

In the following sections, details and background information about the Sequence to Sequence model will be given, the dataset and the approached used for Sequence to Sequence Model for a chat bot will be explained and finally we will review the results.

2 Background

2.1 Recurrent Neural Networks

Recurrent Neural Network (RNN) is a type of Artificial Neural Network where, unlike other feed-forward neural networks, previous states affect the outcome of the current state. You can the representation of the model in Figure 1. RNNs use their memory to calculate inputs. This feature makes them functional in applications which process sequential data. Some of the successful applications of RNNs can be found in speech recognition, language processing and modeling, translation studies.

2.2 Long Short Term Memory

Even though RNNs architecture allows it to remember information from the past, there is an extension to it. In many cases, dependencies between pieces of

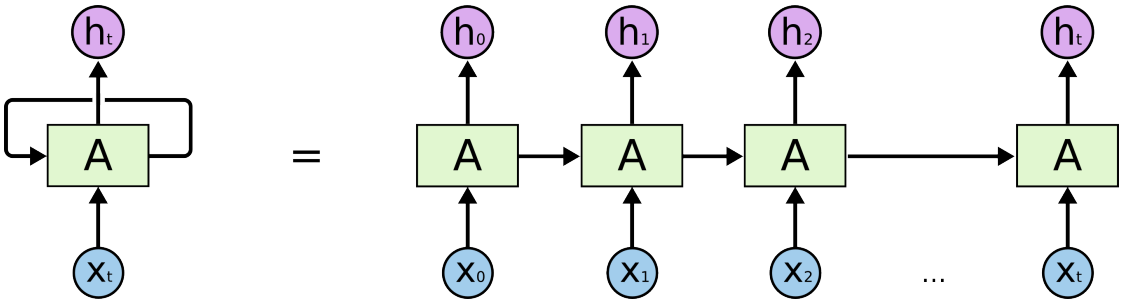
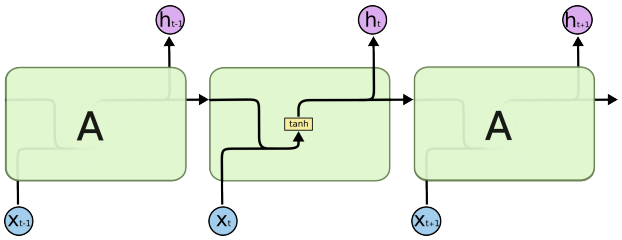
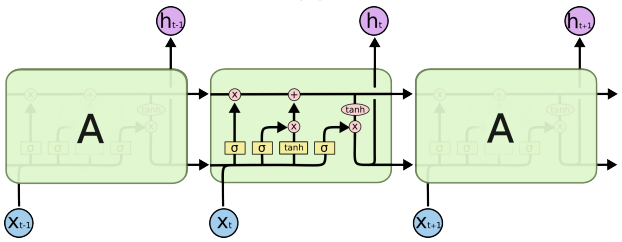


Figure 1: Loop passes the information from previous state to the next state [2].



(a)



(b)

Figure 2: a) The module in a normal (vanilla) RNN. b) LSTMs also have same feedback model, but their repeating module has a different and more complicated structure, that allows preserving dependencies for longer term. Details of the LSTM module are out of the scope of this paper [2].

information can be lost when there is a long gap between them. We will refer to these dependencies as long-term dependencies. To solve long-term dependencies problem, Long Short-Term Memory (LSTM) model developed [4] [2].

3 Model

The sequence to sequence neural network learning problem was tried to be solved by numerous models and approaches. Unlike other neural network applications like classification or problem solving, input data for natural language is a sequence of characters. Lets think about language generation applications. This problem is rather easy since input and output data has the same length. As you can see in the Figure 3, expected output is the shifted input. But in our case, input and output sequences do not have the same length because an answer might

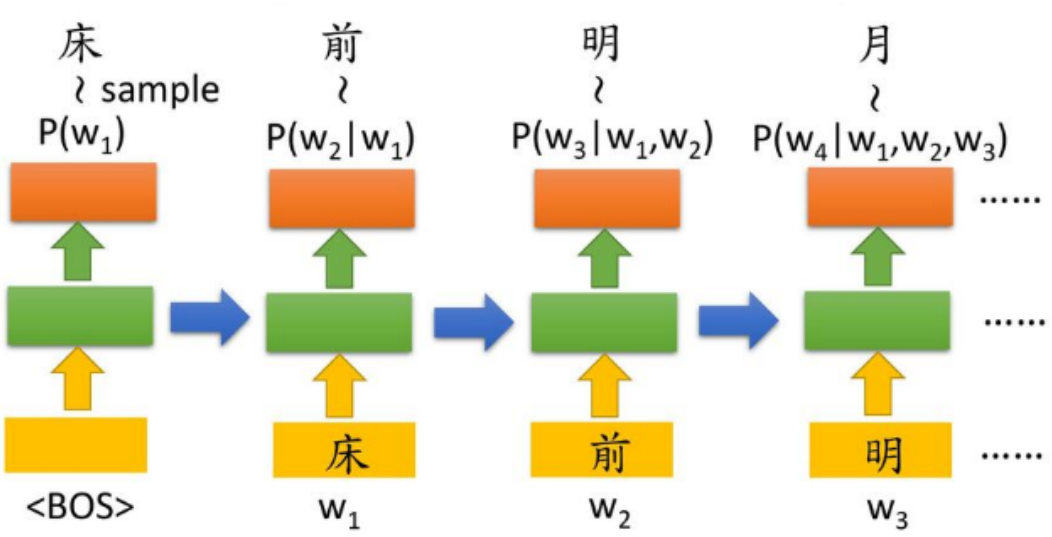


Figure 3: Unlike chatbot applications, in a language generation, input and output have the same length [5].

not have the same number of words as the question.

The solution to this problem suggested by Sutskever et al. [6]. The idea is to use two separate RNNs which will be called the encoder and decoder. As the name suggests, the encoder encodes the input sequence into an intermediate sequence for the decoder. And the decoder will use this sequence to decode output. So basically what sequence to sequence model does is mapping the input sequence into a fixed size vector using LSTM (Figure 4).

Another problem this model faces is input sequences also have different lengths. This problem is solved by using zero padding which basically means choosing longest input sequence length as fixed input length and padding others with zero.

4 Implementation

One of the main goals of the chatbot we want to create was to focus on predetermined topics and show distinct character traits. To achieve this, my approach was to feed network with data that is limited to our interest.

4.1 Data Set

To train chatbots, there are different dataset options to choose from. Most common ones are Twitter chat data, movies scripts, movie subtitles, and available reddit comments. For this chatbot, the reddit comments dataset was chosen. The main reason for this decision was that reddit data is pre-categorized with subreddits. For my implementation, I choose to use data from the following subreddits.

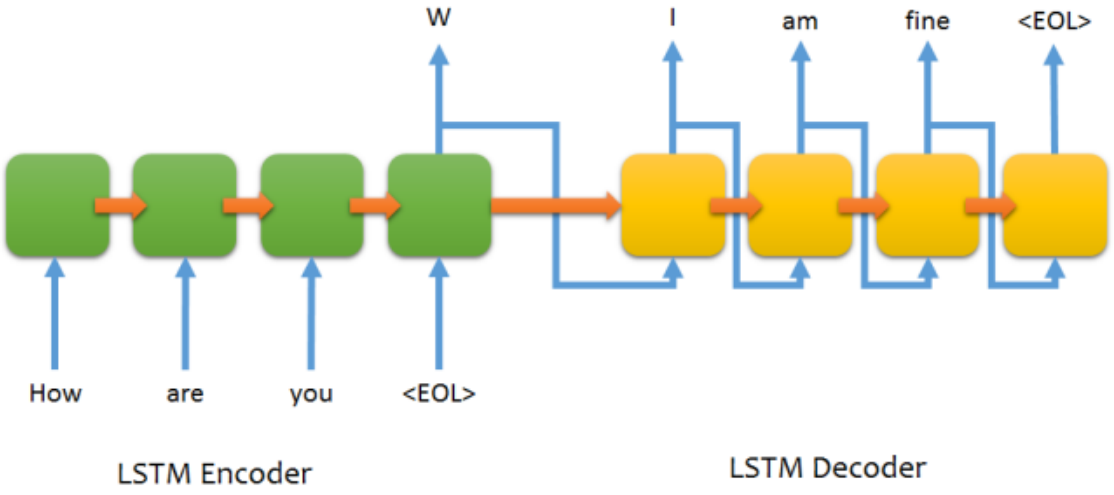


Figure 4: Basic Sequence to Sequence Model. Green parts shows the encoding of the input sequence and how the vector output from encoder fed into the decoder. Notice how the first generated token affects the rest of the sequence [1].

- depression
- offmychest
- self
- amiugly
- Emo
- suicidenotes
- ForeverAlone
- alone
- nosleep
- loneliness
- lonely
- HorriblyDepressing
- collapse
- socialanxiety
- Anxiety
- sciencefiction
- scifi
- StarWars
- AskScienceFiction
- EmpireDidNothingWrong
- wikipedia
- google
- todayilearned
- mildlyinteresting
- interestingasfuck
- Damnthatsinteresting
- DamnInteresting
- BeAmazed
- Showerthoughts
- DoesAnybodyElse
- changemyview
- CrazyIdeas
- AskScienceFiction
- EverythingScience
- Science
- AskScience
- doctorwho
- gallifrey
- totallynotrobots
- shittyrobots
- AskReddit

Here is the example data for one comment:

{

```
"author":"Dethcola",
"author_flair_css_class":"",
"author_flair_text":"Clairemont",
"body":"A quarry",
"can_gild":true,
"controversiality":0,
"created_utc":1506816000,
"distinguished":null,
"edited":false,"gilded":0,
"id":"dnqik14",
"is_submitter":false,
"link_id":"t3_73ieyz",
"parent_id":"t3_73ieyz",
"permalink":"/r/sandiego/comments/73ieyz/best_place_for_granite_counter_
"retrieved_on":1509189606,
"score":3,
"stickied":false,
"subreddit":"sandiego",
"subreddit_id":"t5_2qq2q"
}
```


To train the bot comments from March 2018 cleared and prepared. For the mentioned subreddits, total 1347370 comments collected. As mentioned before to make zero padding, choosing longest input as the fixed input size is the common way. But since our dataset has some dirty data with links and very long comments; we predetermined the maximum input length. That also reduced the number of comments used in the application. In addition to the Reddit data, we decided to add information like name and location to the chatbot to see if it will remember this information after training.

One important note for the implementation by Sutskever et al. s paper is to reverse the order of the input. They found that reversing the input boost the performance of the LSTM [6]. So after input sequences prepared, their order also reversed. To make the model less memory consuming and make things easier for the LSTM each word encoded to an id. While preparing the input sequences, the vocabulary set is also prepared. Since we have rather dirty data and very large dataset, putting a vocabulary limit will make training faster. For this implementation, we used the 20000 most frequent words from the dataset and put $< UNK >$ character for the rest. And lastly, zero padding added to the input sequences as $< ZERO >$ to make it ready for the encoder LSTM.

4.2 Implementing Network

Before feeding the data to the LSTM, the first step was to create an embedding layer. Embedding layer, by definition, creates dense vectors of fixed size from positive integer indexes. That way we do not have to vectorize the input sequences.

Embedding layer is followed by an LTSM which we called encoder with 1000 nodes in its hidden layer. For the decoder, output vector from the encoder will be repeatedly fed into the decoder with 3 layers. This repetition number is actually the maximum output length which decided by the longest output sequence we have in our dataset. And lastly, the softmax function is chosen as the activation function.

5 Results

The model trained for a week, nearly 45 epochs. At the end of the 43rd epoch, the loss was around 1.400 and accuracy was around 0.82.Even though the model got relatively high accuracy from training data, test results were not successful. Below, you can see some of the conversation logs from different epochs through the training.

HUMAN: hi
BOT: UNK
HUMAN: ok
BOT: e
HUMAN: oki
BOT: UNK
HUMAN: po
BOT: if people https but people up and to fight fight

HUMAN: a
BOT: goodbye
HUMAN: b
BOT: gas goodbye
HUMAN: c
BOT: goodbye

HUMAN: r
BOT: crazy
HUMAN: s
BOT: s
HUMAN: t
BOT: interesting theory though

HUMAN: how
BOT: the the goodbye

HUMAN: hello
BOT: UNK
HUMAN: hi
BOT: that
HUMAN: how are you
BOT: it
HUMAN: do you love me
BOT: that
HUMAN: me
BOT: that
HUMAN: ok
BOT: depends your risk risk risk you you knows down down you you risk you anything 20 gains
HUMAN: haha
BOT: it

HUMAN: what is your name
BOT: it

HUMAN: hi
BOT: that
HUMAN: ho
BOT: you

HUMAN: g
BOT: goodbye

HUMAN: a
BOT: nothing really really goodbye
HUMAN: b
BOT: then be gonna goodbye
HUMAN: d
BOT: nearly close scene if a four never ending gauntlets
HUMAN: e
BOT: re oh you the remake you hope they hold an grudge

HUMAN: tell
BOT: a death UNK d

HUMAN: name
BOT: imgur UNK UNK

HUMAN: hello
BOT: UNK the it
HUMAN: what are you doing
BOT: UNK the it

6 Conclusion

In this paper, we introduced and implemented the Sequence to Sequence model. Our aim was to build a chatbot by using this structure. Even though we had high training accuracy, our implementation did not give the results we hoped for. As much as we observed, training a sequence to sequence network requires a powerful GPU processor and memory. Since model itself consists of two deep LSTMs with the high number of hidden nodes, training it with high number of words takes time and resources. And this fact, of course, affects the performance of the network. Calculating the expected size of the network beforehand can help with the data selection, data preparation and selecting the hyperparameters. And doing this with cleaner and more precise data might give better results in the future.

7 Bibliography

References

[1] Seq2seq. `https://github.com/farizrahman4u/seq2seq`. Accessed: 2018-06-06.

[2] Understanding lstm networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Accessed: 2018-07-11.

[3] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8599–8603. IEEE, 2013.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] Hung-Yi Lee. Conditional generation by rnn and attention. `https://slideplayer.com/slide/12107737/`. Accessed: 2018-07-11.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[7] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[8] Zi Yin, Keng-hao Chang, and Ruofei Zhang. Deepprobe: Information directed sequence understanding and chatbot design via recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2131–2139. ACM, 2017.