International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 3 No: 4 June 2018

Chatbot Powered By Deep Learning with Neural Machine Translation

Ahmad SufrilAzlan Mohamed
School of Computer Sciences 11800 UniversitiSains Malaysia, Penang, Malaysia

Abstract: The chat-bot made with deep learning is aimed at eradicating the rule-based chatbots that are extensively in use nowadays. Rule-based chatbots require a programmer to explicitly program them with thousands of rules and hard-coded responses, presuming the user's responses. This is a pretty tedious process and demands skilled manpower. The deep learning chatbot eliminates all this at the cost of a huge and relevant dataset that has real human interactions. The chatbot thus made can either be used for general or specific purposes grounded on the dataset used for training the model. This project makes use of Deep Learning to make the machine learn by itself on how humans 'speak' or 'handle' or 'behave' under the given circumstances and mimic them effectively even in the absence of the 'rules'.

1. INTRODUCTION

The project will have a great impact on data-driven businesses that need to maintain a constant relationship with their customers.

Statistically, an average application on the play store is downloaded and used only once. Hence, the catch is a super-app that does the job of all these mini apps, that too without hard-coded backends.

For example, in the future, people don't have to install apps of Swiggy™, Food Panda™, Ola™ and Uber™ to harness all of their services. People can interact with these companies via their respective Facebook pages and order food or book a ride just by interacting with a bot that makes them feel like they're talking to a real person, making processes much lighter and automated.

Currently, the organizational boundaries of the project are limited to English Speaking countries, as the datasets currently available for training are only in English. This could delay the incipience of AIs in vernacular languages to a considerable extent.

2. LITERATURE SURVEY

Chatbot Using A Knowledge in Database Human-to-Machine Conversation Modelling [1]

DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents [2]

AI based chatbot [3]

Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora [4]

3. EXISTING SYSTEM

Objectives: The major objective of the project is to eliminate the rule based systems from machine intelligence.

Systems: The existing systems include Siri, Natasha, Cortana, Google Assistant.

Method: The method the existing systems use are rule-based. This will be ruled out by introducing Deep Learning into the picture.

4. PROPOSED SYSTEM

The proposed system plans on ruling out the rule-based systems from the main screen of Machine Intelligence. This idea is precisely realized on a chat-bot. Tons of examples (5 million in this case) have to be fed into the model in order for it to learn. The proposed model uses a Recurrent Neural Network comprising of LSTM cells, which is normally the choice for time series data such as natural language.

Neural Machine Translation will be the algorithm of implementation. The example data will be passed through the model, during which the same will be trained (or) the model learns to reduce errors by improving accuracies.

During the training phase, the model identifies the patterns in the example questions and answers that have been fed into the model. It then builds a knowledge bank, called a checkpoint file, in TensorFlow terms.

A checkpoint file is a file that TensorFlow generates towards the end of a specific epoch or training step. Usually, the number of epochs for a chat-bot with 5 million examples would be 1 million epochs, on a GPU.

International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 3 No: 4 June 2018

5. ARCHITECTURE DIAGRAM

The dataset being used in this project is a dump of 5 million comments and their corresponding replies from a social media website, with which the model learns to speak like humans. As it is a real-world dataset, it is evident that it will have missing data and imperfect ones. Hence, the first step would be cleaning or pre-processing the dataset.

Post that, the data has to be split into training and testing data. Testing data is to assess the performance of the model itself.

The training data is relinquished into a recurrent neural network model (inside the NMT model).

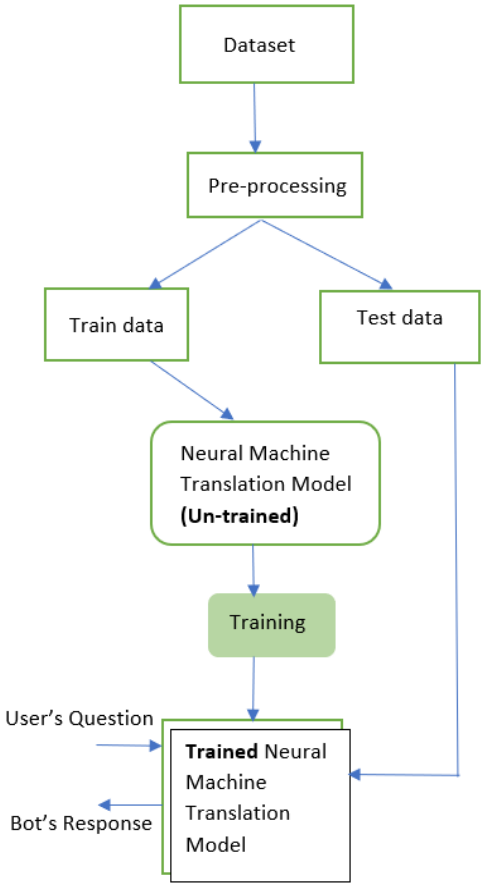

Figure 5.1. Architecture of the Chat-bot

The training data is given into the model during 'training'. Training is defined as the process of minimizing errors by learning from examples during a course of time. This results in a trained model. Users can interact with the chatbot/trained model.

6. WORKING PRINCIPLE

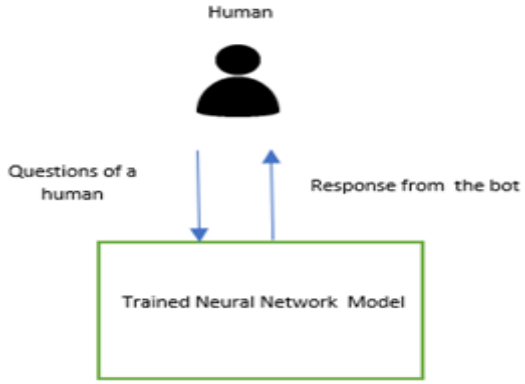

Figure 6.1. Working Principle of the Chat-bot

The user, first of all, gets to see the welcome page of the project. He, then, will be prompted to type in his message. His message will be taken in and processed by the text processers before it is sent into the neural network. Once it reaches the neural network, it will be processed and it 'calculates' the answer to the question raised by the human with a respectable amount of accuracy. The calculated answer is given out through the terminal or the UI created (if any).

7. USER MODULE

The following is a screen grab of the chatbot.

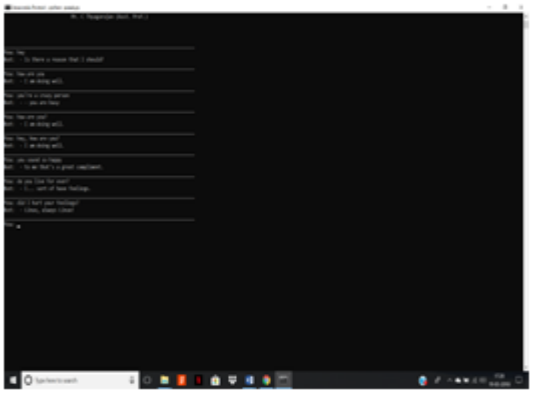

Figure 7.1. Screen Grab of the User Module

The chatbot shown above was trained on a CPU. If it were trained on a Graphical Processing Unit, the performance would improve ten folds. The User Module only has an area for the user and the bot to chat. It can be substituted with

International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 3 No: 4 June 2018

any UI that enables a two-person chat space, provided the trained neural network model remains at the back end of the user interface.

8. DATA PRE-PROCESSING

There are various steps involved in data pre-processing. It is a crucial step in deep learning projects involving real-world data.

As the dataset is crowd-sourced (in other words) as it finds its root from actual human conversations, it is obvious that it has errors associated with humans, including strong language, typos, etc. All of those have to be explicitly examined and then handled, most obviously, programmatically. The steps are as follows:

Initially, an SQL table is created and the comments and all their corresponding replies are inserted into it.

It's called a ramdisk and it's integr...	Great that it's baked into the kernel, but I'm looking for a gui...
NULL	Sorry, simple oil tech so that pic will probably come in a few ...
NULL	The last one
NULL	And Desolator.
NULL	if the art is copyright it is definitely not legal
Tar in your coffee? D:	That's just chocolate. newlinechar
NULL	Man I love Cerrone as much as the next guy but i don't think ...
Blake seems like a skyrimer	I could get her to play Wow or finalfantasy 14.

Figure 8.1. SQL Table with comments and replies

Then, the under-performing comments are discarded by some standards specific to the dataset being used.

Post this stage, various text-operations like stemming, replacement of additional stop words is performed programmatically, considering the massive size of Deep Learning datasets.

Once the pre-processing steps are done with, the near-perfect dataset (cleaned dataset) has to be split into training and testing data.

The reason for having to partition the source dataset into training and testing sets is to ensure that the trained model is performing well provided that the questions fed in are from any source.

The model, during training, sees only the data in the training set and doesn't know of the data from the testing set. Hence, the model's performance can be assessed by testing the model with the data from the testing dataset.

9. DEEP LEARNING MODEL

The model explained below deals with language translation. However, the algorithm also proves to be acceptable for translating sentences, as 'some translation' is involved. The relevancy becomes familiar with fig: 9.1.1.

The model used is called Neural Machine Translation (NMT). It was initially used to translate English to French. It was an advancement of the Sequence2Sequence model (now deprecated) of TensorFlow. This paper focuses on the adoption of this language translation model for sentence translation. The following diagram explains how an Encoder-Decoder model (NMT) works.

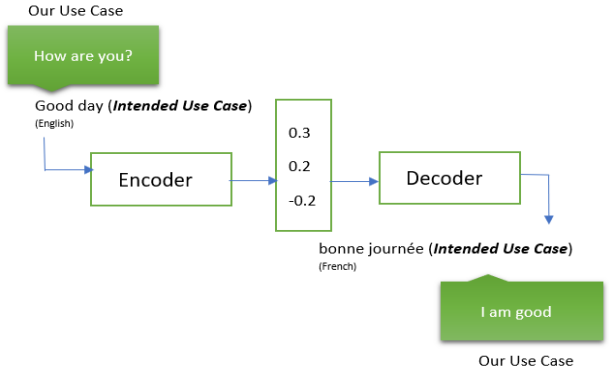

Figure 9.1.1. Encoder-Decoder Architecture

Back in the days, machines used to translate a sentence by first breaking it into a series of phrases and then translating them phrase by phrase. This approach caused numerous issues as not all languages have the same syntax.

Humans, on the other hand, translate the entire sentence of one language into an entire sentence in the other language, keeping in mind the syntax of each language involved in the translation. NMT does exactly the same. It makes use of an encoder and a decoder to accomplish this.

The encoder creates a series of vectors with a series of numbers. This represents the meaning of the sentences. The decoder then decodes the meaning of the sentence in the other language, keeping in mind the syntax of the other language. This approach is being followed in Google Translate, which is more efficient than Microsoft Translator which uses a Sequence2Sequence approach.

International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 3 No: 4 June 2018

For time-related data like natural language, the natural choice of neural network would be a Recurrent Neural Network. In this project, a Deep Recurrent Neural Network with an LSTM (Long Short-Term Memory) is used.

In a recurrent neural network, a neuron gets the natural input along with the output from the previous neurons as inputs, thus retaining what the neural network had learned from the other farther side of the neural network.

An LSTM helps the deep neural network remember longer sentences. Typically, it can remember up to 10 tokens/words, which is sufficient for a chatbot.

A vocabulary is built with random, but meaningful numbers assigned to words/tokens. Frequent words get special treatment. The model learns about the embedding weights during training. This happens at the embedding layer (Refer fig: 9.1.2)

The data is passed into the encoder which encodes, then through the decoder, which decodes. This is where the actual translation of sentences take place. These type of models perform well for larger datasets.

The following figure explains a Recurrent Neural Network.

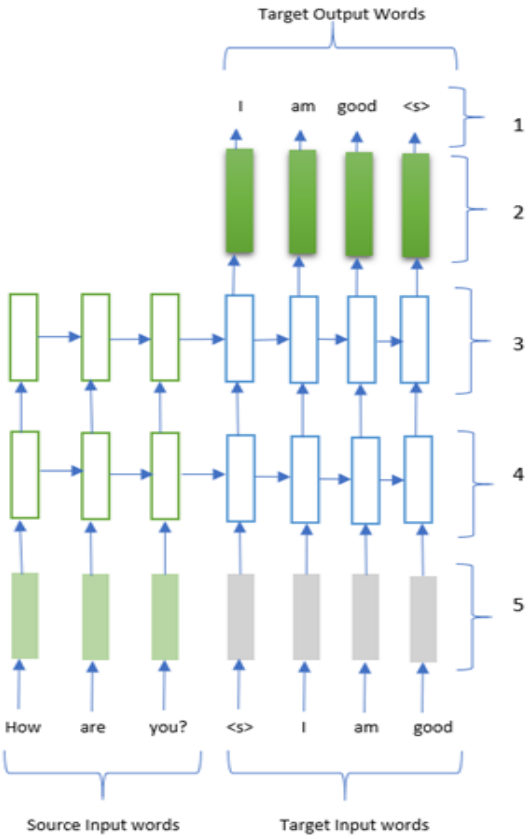

Figure. 9.1.2 Recurrent Neural Network

The numbers represent the following:

1. Loss Layer
2. Projection Layer
3. Hidden Layer 2
4. Hidden Layer 1
5. Embedding Layer

The loss layer calculates the loss in that cycle of training. The projection layer maps the word vectors in the vocabulary into a continuous vector space.

The hidden layers consist of LSTM cells that help the deep neural network remember longer sentences

10. REQUIREMENTS

Hardware Requirements:

Like any Deep Learning project, this project requires a decent NVIDIA GPU for training. The foot note is that the GPU used has to be CUDA enabled. The RAM and processor speed also have to be decent for faster and efficient training of the model with a humungous dataset.

Software Requirements:

The project is written in Python 3. TensorFlow 1.1 is used (The latest TensorFlow 1.4 (as of now) doesn't support NMT). Jupyter Notebook and Spyder IDE are used for creating the project. Anaconda is used for management of Python libraries and for the ease of working in an environment. Ubuntu 16.04 is the recommended Operating System for this project.

11. IMPLEMENTED PROJECT

A screen grab of this paper implemented is included below. It has space for the users to communicate with the chatbot/trained deep neural network. It can also be integrated to chat services like Facebook Messenger, or any chatbot in a website.

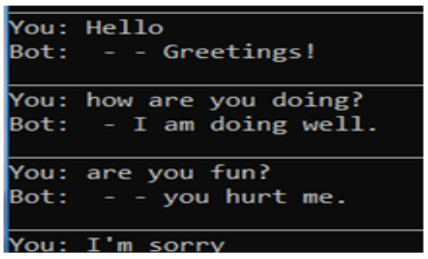

Figure11.1. Implementation of the project

12. FUTURE ENHANCEMENTS

The performance of the model could be substantially improved by training the model in a dedicated GPU. NIVIDIA P5000, for instance.

The project currently lacks filters for harsh language as a massive dataset such as the one used in this project demands greater computational power. A chatbot with filter for strong language and other abnormalities can be built on top of this chatbot as a future enhancement.

13. CONCLUSION

This project, thus successfully eliminates the rule-based chatbots. Now, chatbots can truly be called 'Artificial Intelligence'.

We firmly believe that this paper would substantially reduce the task of humans hardcoding responses of humans. These deep learning chatbots will find their permanent place in fields like hospitality businesses, etc. This will enable humans to be part of more complex errands rather than answering questions.

REFERENCE

[1] Chatbot Using A Knowledge in Database Human-to-Machine Conversation Modelling by Bayu Setiaji, Ferry Wahyu Wibowo.

[2] DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents by Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li , Jianshe Zhou.

[3] AI based chatbot by Prof.Nikita Hatwar, Ashwini Patil, Diksha Gondane.

[4] Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora by Bayan AbuShawar, Eric Atwell.

