Customer Support Chatbot using Seq2seq model

In place of direct interaction with a live human agent, a chatbot is a software programme that conducts an online chat discussion using text or text-to-speech. Chatbot systems are usually designed to successfully imitate the way a person would act as a conversational partner, although many in production are still unable to speak sufficiently or pass the industry standard Turing test. In this project my attempt is to build a simple chatbot that has customer support for various organisations. I plan to achieve this using python along with different libraries and frameworks such as TensorFlow. I have used different models like LSTM, RNN and Seq2Seq. Out of all the models I found Seq2Seq to be produce results with a minimum loss (0.67).

Contents

Keywords

TensorFlow: - TensorFlow is a machine learning and artificial intelligence software library that is free and open-source. It can be used for a variety of tasks, but it focuses on deep neural network training and inference.

Keras: - Keras is an open-source software library for artificial neural networks that includes a Python interface. Keras serves as a user interface for TensorFlow.

RNN: - A recurrent neural network (RNN) is a type of artificial neural network in which nodes' connections construct a directed or undirected graph over time. This enables it to behave in a temporally dynamic manner. RNNs, which are derived from feedforward neural networks, can handle variable length sequences of inputs by using their internal state (memory). As a result, activities like unsegmented, linked handwriting recognition or speech recognition are possible.

LSTM: - In the realm of deep learning, long short-term memory is an artificial recurrent neural network design. LSTM features feedback connections, unlike normal feedforward neural networks. It can handle whole data sequences as well as single data points.

Seq2Seq: - Sequence to Sequence (seq2seq) models are a subset of Recurrent Neural Network architectures that are commonly used (but not exclusively) to handle complicated language issues such as Machine Translation, Question Answering, Chatbot creation, Text Summarization, and so on.

Turing Test: - The Turing Test is a method of artificial intelligence (AI) research that determines if a machine can think like a human being.

Chatbot: - In place of direct communication with a live human agent, a chatbot or chatterbot is a software programme that conducts an online chat discussion using text or text-to-speech.

Introduction

A chatbot is a software programme that conducts an online chat discussion utilising text or text-to-speech in place of direct interaction with a real human agent. Although many in production are currently unable to talk effectively or pass the industry standard Turing test, chatbot systems are typically built to successfully simulate the way a person would act as a conversational partner. In this project, I'm attempting to create a simple chatbot that can provide customer service for a variety of businesses. I want to accomplish this by combining Python with several libraries and frameworks, such as TensorFlow. LSTM, RNN, and Seq2Seq are some of the models I've employed. Seq2Seq, out of all the models, produced the best results with the least amount of loss (0.67).

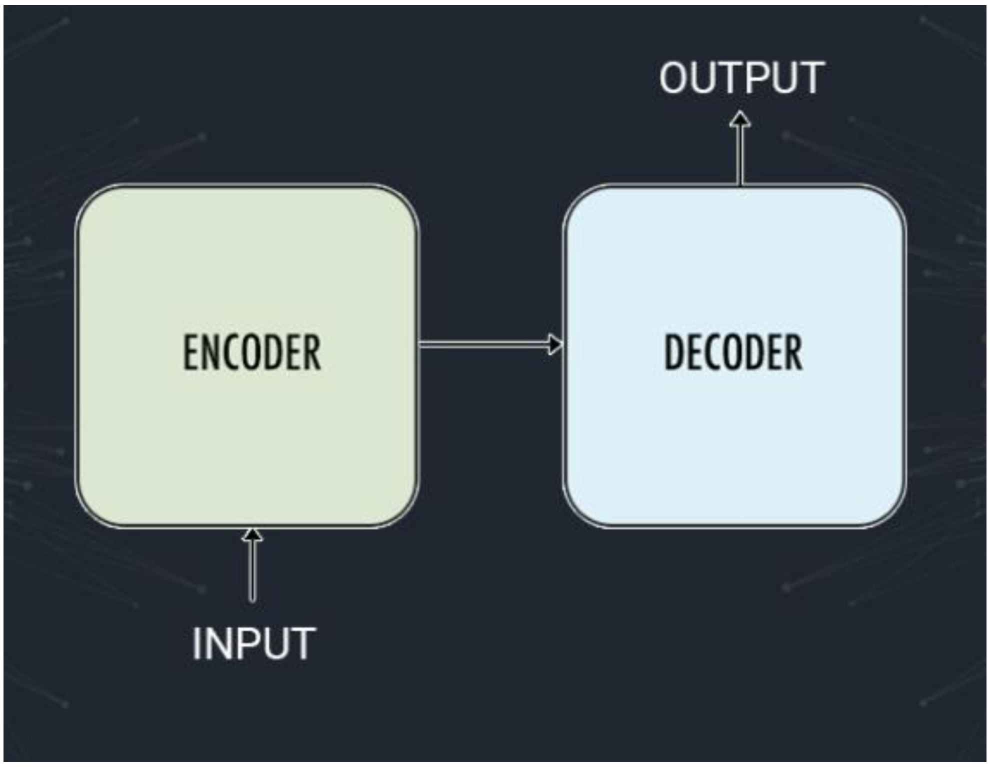
The dataset has been obtained from Kaggle which contains tweets, both inbound and outbound. The dataset has been cleaned thoroughly and then it made ready for training. Seq2seq, LSTM and RNN were tried on the dataset one by one each with 10 epochs.

Dataset Decription

The Customer Support on Twitter dataset is a huge, modern corpus of tweets and replies that may be used to help with natural language processing and conversational models, as well as research into modern customer service techniques and effect. It contains around 3 million customer service tweets from a variety of firms. The dataset link is provided below:- https://www.kaggle.com/thoughtvector/customer-support-on-twitter/data. Customers' calls are received by an inbound call centre. Inbound call centres are frequently monitored by support staff because the calls tend to originate from current customers with problems or concerns. A call centre that performs outbound calls to customers is known as an outbound call centre. Outbound centres are often used by sales teams to cold contact potential clients about their products. Companies may also make outbound calls to survey customers and gather market data. Based on inbound and outbound we make train and test, and check for discrepancy if any.

Sequence2Sequence Architecture

A sequence to sequence model, initially introduced by Google in 2014, tries to map a fixed-length input with a fixed-length output when the lengths of the input and output may change.



The encoder receives the input sequence and summarises the data in internal state vectors, also known as context vectors (in case of LSTM these are called the hidden state and cell state vectors). The encoder's outputs are discarded, leaving just the internal states. To aid the decoder in making correct predictions, this context vector seeks to encompass the information for all input items.

The decoder is an LSTM whose starting states are set to the Encoder LSTM's final states, i.e. the context vector of the encoder's final cell is fed into the decoder's first cell. The decoder generates the output sequence using these beginning states, and these outputs are likewise taken into account for subsequent outputs.

As described above I have used LSTM in both the encoder and decoder.

<u>Code Explanation</u>

```
def create_model():

    shared_embedding = Embedding(

        output_dim=EMBEDDING_SIZE,

        input_dim=MAX_VOCAB_SIZE,

        input_length=MAX_MESSAGE_LEN,

        name='embedding',

    )


    # ENCODER


    encoder_input = Input(

        shape=(MAX_MESSAGE_LEN,),

        dtype='int32',

        name='encoder_input',
    )

    embedded_input = shared_embedding(encoder_input)


    # input sequence provided.
    encoder_rnn = LSTM(
        CONTEXT_SIZE,
        name='encoder',
        dropout=DROPOUT
    )

    context = RepeatVector(MAX_MESSAGE_LEN)(encoder_rnn(embedded_input)
)

    # DECODER

    last_word_input = Input(
        shape=(MAX_MESSAGE_LEN, ),
        dtype='int32',
        name='last_word_input',
    )

    embedded_last_word = shared_embedding(last_word_input)

    decoder_input = concatenate([embedded_last_word, context], axis=2)

    decoder_rnn = LSTM(
        CONTEXT_SIZE,
        name='decoder',
        return_sequences=True,
        dropout=DROPOUT
    )


    decoder_output = decoder_rnn(decoder_input)


    # TimeDistributed allows the dense layer to be applied to each deco
der output per timestep
    next_word_dense = TimeDistributed(
        Dense(int(MAX_VOCAB_SIZE / 2), activation='relu'),
        name='next_word_dense',
    )(decoder_output)


    next_word = TimeDistributed(
        Dense(MAX_VOCAB_SIZE, activation='softmax'),
        name='next_word_softmax'
    )(next_word_dense)


    return Model(inputs=[encoder_input, last_word_input], outputs=[next
_word])
```

The above code contains the function for creating the model. It follows the sequence to sequence architecture and contains an encoder and a decoder which have LSTM.

<u>**Model Comparision**</u>

Model Name	Loss
Seq2Seq	0.67

RNN	0.83
LSTM	0.75

A faulty guess results in a loss. To put it another way, loss is a statistic that indicates how inaccurate the model's forecast was for a particular case. The loss is 0 if the model's forecast is flawless; otherwise, the loss is bigger. So, from the above table we can see that we have minimum loss for Seq2Seq. The Seq2seq is superior since it predicts a word supplied by the user, and then each of the subsequent words is predicted based on the probability of that word occurring. This method will be used in the development of our Generative chatbot for text creation based on user input.

Conclusion

Through this research I came to the conclusion that Sequence to sequence model works best for creating chatbots. Also, It handles language issues very promptly. Also, it was found that the loss produced is very less as compared to its competitors. The chatbot handles a very different range of queries and hence it solves the problem by being a general purpose chatbot.