Artificial Intelligence Based Chatbot using Sequence-to-Sequence Model

Varsha.C.Bendre, Dr. Savita Choudhary
Dept. of CSE, Sir M Visvesvaraya Institute of Technology
Bangalore, India
Varsha6319@gmail.com

Abstract—**The paper presents the architecture of the Chatbot using Sequence to Sequence model. The paper depicts detailed functional blocks, associated layers and software methodology adopted during the development.**

Keywords—Chatbot; Retrieval Based Model;

I. INTRODUCTION

Chatbot is a program which aims to make a conversation between the human and the machine [1]. The bots have a repository of pre-defined responses that can be used.

As shown in fig. 1, the input message is divided into Intent and Entities. Intent is the information about the context and the Entities are the structured bits of information from the message. The classified intent and the recognized entity are directed into the algorithm. Context which represents the conversation at that point in time and the pre-defined potential responses are considered for the Response Generator. The output of the model among the multiple responses is the response with the highest score. This conversational agent is based on the retrieval based model.

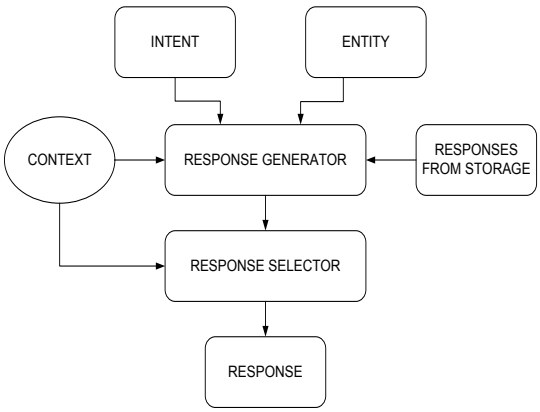

Fig.1. Functional Block Diagram

The Chatbot must be able to determine the best response to any given message that it receives [2]. The response should either the answer to the sender's question, give the sender relevant information, ask follow-up questions or continue the conversation in a realistic way [3].

The Chatbot needs to understand the intentions of the sender's message to determine the type of response message is required and follow correct grammatical and lexical rules while forming the response [4].

Section II presents the proposed model. The datasets are shown in Section III. Section IV gives details on the test methodology. Section V and VI gives turning test and performance on long sentences respectively. Conclusion is presented in section VII.

II. PROPOSED MODEL

As shown in fig. 2, the proposed model is multilayer architecture. It broadly consists of three different layers each with well classified objectives. The three layers are: Back End Systems, Response Generation and the Chat Channel.

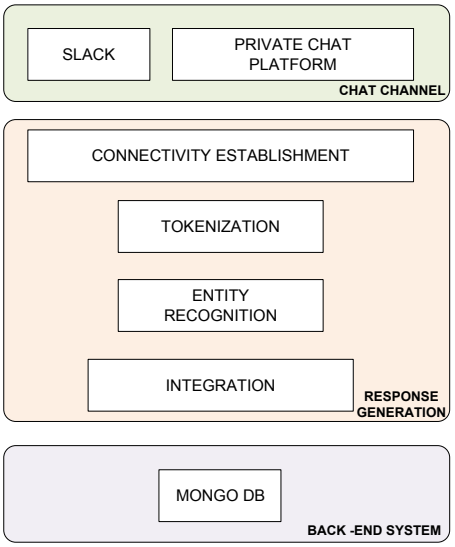

Fig.2. Proposed Model

Various functional blocks associated in the proposed model are as follows:

A. *Supervised Learning*

In the first stage of training, model is built on the prior work of predicting a generated target sequence, for a given dialogue history, using the supervised learning. That is the Seq2seq (Sequence-to-Sequence) model.

B. *Padding*

The dataset which is of variable length is converted to a fixed length sequence before feeding it to the algorithm to start the training. There are a few special features that are used in padding namely, EOS (End of Sentence), PAD (Filler), GO (Start decoding), UNK (Unknown or word not in vocabulary).

C. *Bucketing*

To avoid padding that leads to extraneous computation of the data. The group sequences of similar lengths are put into the same buckets. The Embedding layer is the first layer in the network where the word embedding is done. The embedding layer maps an index to word from vocabulary to a dense vector of given size. The embedding layer weights are trained with the other parameters in the seq2seq model.

D. *Seq2Seq Model*

With Seq2Seq model the UUT (Unit under Test) can build and train sequence-to-sequence neural network models in 'Keras'. Such models are useful for machine translation Chatbots. The encoder decoder network needs to be able to understand the type of responses in the decoder outputs that are expected for every query that is the encoder inputs.

III. DATA SETS

A. *Cornell Movie Dialog Corpus*

The corpus contains a large metadata. It is a rich collection of fictional conversations extracted from raw movie scripts. It has 220,579 conversational exchanges between 10,292 pairs of movie characters, involves 9,035 characters from 617 movies, in total 304,713 utterances.

Movie metadata included:

- 'Genres'
- 'Release Year'
- 'IMDB Rating' and
- 'IMDB votes'

Character metadata included:

- 'Gender' (for 3,774 characters) and
- 'Position' on movie credits (3,321 characters)

B. *Ubuntu Corpus Datasets*

Ubuntu uses IRC (Internet Relay Chat) and offers real time problem solving. It contains files which includes the data for the response classification task described in the paper.

The data is split into following sets:

- Train sets
- Validation sets and
- Test sets

Each example is a triple containing:

- Context
- Response and
- Flag

There are two sets of chat messages in the corpus namely, 'Training Set', the one which consists of the 'Unlabeled' messages and 'Testing Set', one which consists of the 'Labeled' messages [5].

IV. TEST METHODOLOGY

The fig. 3 shows the response generation in the proposed model.

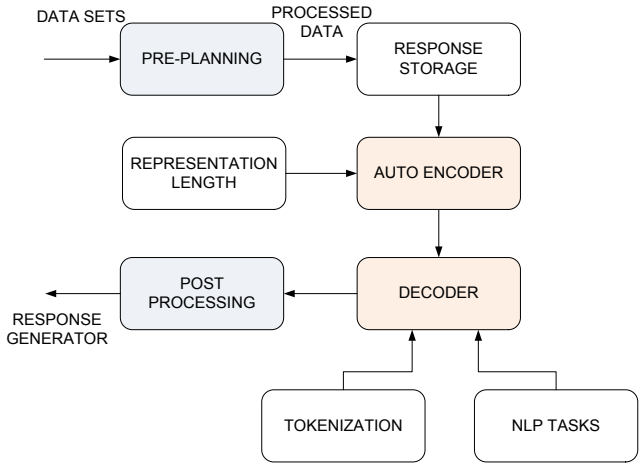

Fig.3. Response Generator

However, a bot can answer a question thrown to it in different ways. Few of them are as follows:

- No response at all
- Invalid response
- Invalid response for a valid question
- Valid response for a valid question

The desideratum of the Chatbot is to convert the given sequence of the symbols into a fixed size feature vector that will lose the unnecessary information and encodes the required information in the sequence. The visualized data flow in the encoder along the time axis, as the flow of local information from one end of the sequence to another. The two Recurrent Neural Networks (RNN) are trained jointly to maximize the 'Conditional Probability' of the target sequence given a 'Source Sequence' [6].

Every hidden state influences the next hidden state and the final hidden state can be seen as the summary of the sequence. This state is called the context and that represents the intention of the sequence that is being used. By looking into the context mentioned the decoder generates another sequence which may be a symbol at a time. At each time step, the decoder is influenced by the context and the previously generated symbols.

During each time step in the decoder, instead of using a fixed context (last hidden state of encoder), a distinct context vector 'ci' is used for generating word 'yi'.

This context vector 'ci' is basically the weighted sum of hidden states of the encoder.

$$ci = \sum j = 1n\alpha ijhj \quad .. (i)$$

where n is the length of input sequence, hj is the hidden state at time step j.

$$\alpha ij = \exp(eij)/\sum k = 1n\exp(eik) \quad .. (ii)$$

'e^{ij}' is the alignment model which is function of decoder's previous hidden state 'si−1' and the 'j^{th}' hidden state of the encoder.

Each hidden state in the encoder encodes information about the local context in that part of the sentence. As data flows from word 0 to 'n^{th}' word, this local context information gets diluted.

The alignment model gives us a measure of how well the output at 'i^{th}' position matches with the inputs at around 'j^{th}' position. Based on which, we take a weighted sum of the input contexts 'Hidden States' to generate each word in the output sequence.

In addition to the encoder and decoder layers, a Seq2Seq model may also contain layers such as the left-stack which are stacked LSTMs (Long-Short Term Memory) on the encoder side and the right-stack which are stacked LSTMs on the decoder side, resizers that are for shape compatibility between the encoder and the decoder and dropout layers to avoid over fitting [7] [8].

V. Turning Test

'Turing Test' is a measure to determine whether a machine can demonstrate human intelligence in thoughts, words or any kind of action [9]. The modern version of proving this test is with the Chatbot, the test is marked successful if more than 30% of the judges, after five minutes of conversation, consider the computer to be human. 'Turing Test' has more than one human judge interrogating and chatting with both subjects.

VI. Performance on Long Sentence

Using the Encoder-Decoder architecture for the breaking up of the long sentences and give the relevant replies for the user. The auto encoder is used to learn a new representation length for long sentences and then the decoder network will interpret the encoded representation into the desired output. This also involves the recent encoder-decoder LSTM style networks used for the natural language translation.

One encoded character is shown at a time to the network from the input sequence. The internal representations of the relationships are developed and the relationship between the steps in the input sequence is learned through the encoding level.

There are still difficulties in learning from very long sequences, but the more sophisticated architecture may offer additional leverage or skill, especially if combined with one or more of the techniques.

The LSTM is utilized for the hidden units in the neural network, as LSTM units have special gating machinery that allows them to propagate error over many time steps, as it is reasoned that this choice is logical given our decision to model characters instead of words, leading to very long decompositions of sentences as character chains and finer-grained, more-difficult-to-capture character level language dynamics.

VII. Conclusion

The best way to evaluate a conversational bot is to measure whether or not it is fulfilling its task, for example to solve a customer support problem, in a given conversation. But such labels are expensive to obtain because they require human judgment and evaluation. Sometimes the goal is not well defined in the case of the open-domain models. This model can be improved further by training it with the other datasets available other than the movie dialogues. The presented Chatbot has an accuracy of around 66%, which can be improved through further training.

Acknowledgment

Authors are very much thankful to the Head of Department, CSE in Sir MVIT, Bangalore for giving an opportunity to publish this technical paper.

References

[1] 'Deep Learning for Chatbots', www. wildml.com.

[2] Wang, Peng, et al. "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification." Neurocomputing 174 (2016): 806-814.

[3] Shayan Sadigh, ‘Exploring Seq2Seq For Generating Human-Like Responses on Internet Forums’, University of California, 2016.
[4] Kashyap P, "Industrial Applications of Machine Learning." Machine Learning for Decision Makers. Apress, Berkeley, CA, 2017. 189-233.
[5] M. Young, ‘Artificial Intelligence Predictive Modeling and Chatbots Applications in Pharma’, The Technical Writer’s Handbook, Mill Valley, CA: University Science, 1989.
[6] Karpathy, Andrej. "The unreasonable effectiveness of recurrent neural networks, 2015." URL http://karpathy. github. io/2015/05/21/rnn-effectiveness (2016).
[7] Uthus, David C., and David W. Aha. "The Ubuntu Chat Corpus for Multiparticipant Chat Analysis." AAAI Spring Symposium: Analyzing Microtext. Vol. 13. 2013.
[8] Shevat, Amir. Designing Bots: Creating Conversational Experiences. " O'Reilly Media, Inc.", 2017.
[9] Morta, Rene, and Elmer Dadios. "Proposed system for predicting Buy, Hold and Sell recommendations for a publicly listed Philippine company using computational intelligence." TENCON 2015-2015 IEEE Region 10 Conference. IEEE, 2015.
[10] Qiu, Minghui, et al. "Alime chat: A sequence to sequence and rerank based chatbot engine." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2017.
[11] http://www.chat4all.org
[12] Sutskever, Ilya, James Martens, and Geoffrey E. Hinton. "Generating text with recurrent neural networks." Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011.