Artificial Intelligence is a branch of Science which deals with helping machines find solutions to complex problems in a more human-like fashion. This generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way [1]. AI has been a go-to solution for automating a plethora of simple as well as complex tasks. Right from room cleaning robots to missile launching drones, intelligent systems have been developed that are capable of doing the tasks just like humans or better. This automation mechanism often involves another aspect of Artificial Intelligence – Learning.

Learning or Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed [2]. Thus, a machine is able to learn from its past experiences just like a human does. It does get smarter and more intelligent as it gets more and more data by building its knowledge base. A machine is able to learn from various kinds of data like text, images, video etc. The field of machine learning which involves processing textual data, understanding and learning from it, has got Natural Language Processing, a broad area of the learning and intelligence, which the project focuses on.

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human – computer interaction. It involves two aspects – language understanding and language generation. Understanding the language is the aspect of NLP that involves processing and learning from the input text, whereas generation of natural language involves the prediction of some text based on the input. Natural language generators have applications in translation of languages, conversational models etc. This aspect is explored further in the current project.

II. DESIGN

A. Proposed System Feature

In order in to optimize the design process and also to ensure that the design is efficient and the final product designed meets the desired requirements and achieves the goals and objectives of the project, it is essential to divide the project into various components or modules whose description has to be defined clearly. The different modules needed and their descriptions are given below:

1. Input Module: This module is responsible for accepting the user input question. The input is accepted using a text box which makes use of a Tkinter Python API.

2. Processing Module: The data from the input module is to be sent to the backend for processing. A JavaScript method
call is used here. Data is sent to Python where concepts like Recurrent Neural Networks (RNN) processes the data using its input hidden and output layers.

3. Output Module: The processed data from the backend is to be displayed on the Tkinter Window using the chat box along with the input question. The output module achieves this by transferring processed data using the JavaScript method call.

B. Software design

1) Front End Design:
The Graphical User Interface is attained using a Python API called Tkinter using the following steps:

• A Tkinter Window consists of a chat box for display and a text box for accepting input question.
• A submit button is used which when pressed triggers two events.
• The first is that the question given as input is displayed in the chat box.
• The second event causes a JavaScript method call at the python backend which transfers the input to the backend for processing [9] [12].
• The output feedback from the backend is sent using the JavaScript call and displayed on the chat box [18].
• A series of different input is processed the same way as above.

2) Backend design:
A Recurrent Neural Network is a straightforward adaptation of the standard feed-forward neural network to allow it to model sequential data. At each timestamp, the RNN receives an input, updates its hidden state, and makes a prediction. The RNN’s high dimensional hidden state and nonlinear evolution endow it with great expressive power, enabling the hidden state of the RNN to integrate information over many timestamps and use it to make accurate predictions [9] [10].

3) System Design:
Figure 2 shows an RNN being unrolled (or unfolded) into a full network [2]. By unrolling we simply mean that we write out the network for the complete sequence. For example, if the sequence we care about is a sentence of 5 words, the network would be unrolled into a 5-layer neural network, one layer for each word.

The formulas that govern the computation happening in a RNN are as follows [16]:

• \( x_t \) is the input at time step \( t \). For example, \( x_t \) could be a one-hot vector corresponding to the second word of a sentence [3].
• \( z_t \) is the hidden state at time step \( t \). It’s the “memory” of the network. \( z_t \) is calculated based on the previous hidden state and the input at the current step: \( z_t = f(Uz_{t-1} + Wx_t) \). The function \( f \) usually is a nonlinearity such as tanh or ReLU. \( z_{t-1} \), which is required to calculate the first hidden state, is typically initialized to all zeros.
• \( o_t \) is the output at step \( t \). For example, if we wanted to predict the next word in a sentence it would be a vector of probabilities across our vocabulary: \( o_t = \text{softmax}(Vz_t) \).

III. IMPLEMENTATION

Our approach makes use of the sequence-to-sequence (seq2seq) framework. The model is based on a recurrent neural network which reads the input sequence one token at a time, and predicts the output sequence, also one token at a time. During training, the true output sequence is given to the model, so learning can be done by back propagation. The model is trained to maximize the cross entropy of the correct sequence given its context. During inference, given that the true output sequence is not observed, we simply feed the predicted output token as input to predict the next output. This is a “greedy” inference approach. A less greedy approach would be to use beam search, and feed several candidates at the previous step to the next step. The predicted sequence can be selected based on the probability of the sequence.

A. Open Subtitles Dataset
We also tested our model on the OpenSubtitles dataset (Tiedemann, 2009). This dataset consists of movie conversations in XML format. It contains sentences uttered by characters in movies. We applied a simple processing step...
removing XML tags and obvious non-conversational text (e.g., hyperlinks) from the dataset. As turn taking is not clearly indicated, we treated consecutive sentences assuming they were uttered by different characters. We trained our model to predict the next sentence given the previous one, and we did this for every sentence (noting that this doubles our dataset size, as each sentence is used both for context and as target). Our training and validation split has 62M sentences (923M tokens) as training examples, and the validation set has 26M sentences (395M tokens). The split is done in such a way that each sentence in a pair of sentences either appear together in the training set or test set but not both. The OpenSubtitles is quite large, and rather noisy because consecutive sentences may be uttered by the same character. Given the broad scope of movies, this is an open-domain conversation dataset, contrasting with the technical trouble shooting dataset.

B. Model Samples

In addition to the perplexity measure, our simple recurrent model does often produce plausible answers. Here are a few sample question-answer pairs that allowed us to understand the capabilities of the model when trained on the noisy but large dataset of OpenSubtitles from movies. Again, in these conversations, Machine is our Neural Conversational Model, and Human the human actor interacting with it.

Below is an example of how a Chatbot can be interactive with humans:

**Human:** hello!
**Machine:** hello!
**Human:** how are you?
**Machine:** I'm good. Thank You.
**Human:** What's your name?
**Machine:** I'm Jarvis.
**Human:** When were you born?
**Machine:** July 20th.
**Human:** What year were you born?
**Machine:** 1977.
**Human:** Why are we here?
**Machine:** I'm not sure.
**Human:** okay, bye!
**Machine:** bye.
**Human:** see you later.
**Machine:** okay, bye!

IV. RESULT

In this paper, we show that a simple language model based on the seq2seq framework can be used to train a conversational engine. Our modest results show that it can generate simple and basic conversations, and extract knowledge from a noisy but open-domain dataset. Even though the model has obvious limitations, it is surprising to us that a purely data driven approach without any rules can produce rather proper answers to many types of questions. However, the model may require substantial modifications to be able to deliver realistic conversations. Amongst the many limitations, the lack of a coherent personality makes it difficult for our system to pass the Turing test.

V. SCOPE

Automating the manual tasks has been done for a couple of years now. Automating intelligent tasks is the trending area. The current project aims at realizing the possibility of
automating various everyday tasks that need the reasoning of the humans, to operate. The project has applications in various domains and this project proves the possibility of effectively using the concepts of machine learning and natural language processing to build a conversational agent. The conversational agent that can hold conversations in English in the domains that it is already familiar with. The conversational agent built can act as a psychiatrist for people to talk to. It can be used as a help desk manager to answer phones instead of having a mechanism of users selecting appropriate numbers and then talk to the corresponding person who specializes in the particular area, once the pervious data is inputted to the agent.

VI. FUTURE WORK
The project can be extended and improvised to be used in a wide range of domains. The project can be extended to rate the quality or goodness of conversations. The project can also be extended to support multiple users. These can lead to a whole new set of applications. For instance, this can be used in interviews. Every year, a huge amount of resources is spent in interviewing and talking to candidates for job interviews. If the conversational agent is able to recognize the multiple entities involved in the conversation as well as rate the conversations, the interviewer can easily select the candidate without him/her personally interviewing the candidate. The conversational model when trained with a sufficient amount of data can completely replace ecommerce and other helpdesk services.

VII. CONCLUSION
The model that is proposed is an effective usage of concepts of Machine Learning and Natural Language Processing. We see that a seq2seq neural network based approach can be used to train a conversational model. The results obtained show that it is possible for the model to extract the relevant data from a dataset which is not domain specific. It thus proves that a data driven approach without any rules can produce rather proper answers to many types of questions. With the right kind of data available for the training purposes, it is possible for the model to replace the human intervention in a wide range of tasks. The lack of coherence in the model is responsible for it to fail the Turing test.

VIII. ACKNOWLEDGEMENT
First and foremost, my utmost gratitude to Dr. Monica R Mundada (Associate Professor), Dept. of CSE, MSRIT whose sincerity and encouragement we will never forget. She has been our inspiration as we overcame all the obstacles in the completion of this project work. Dr. K. G. Srinivasa, the Head of the Department of Computer Science and Engineering, had kind concern and consideration regarding project work and we would like to thank him for continues support. We would like to thank our beloved principal Dr. N.V.R Naidu for his support and encouragement. This work would not have been possible without the guidance and help of several individuals who in one way or another contributed their valuable assistance in preparation and completion of this study. We would like to express sincere thanks to all the teaching and non-teaching faculty of CSE Department and my dear friends who helped in all the ways while preparing the Report.

IX. REFERENCES
[8] Antonella De Angeli. “To the rescue of a lost identity Social perception in human-chatterbot interaction”. School of Informatics, the University of Manchester.