

## Chatbot: Jarvis

### System Description:

Jarvis (aka chatbot) is a bot designed to be of service, whether it be for information or a general conversation, with which can be interacted only through text. It is like one of those virtual assistants but without any voice feature. The information regarding sports is well known by Jarvis. Information such as the most trending or significant occurrences in sports. General conversation is that of normal conversing without the requirement for additional knowledge for the chatbot. Jarvis is solely based on various nltk (natural language tool kit) techniques and methods and tensor-flow. Other necessary libraries were pickle, json, numpy and random for different purposes.

NLTK is a very useful python library that can be utilized to do text processing. To process the text there were many tools such as: 1) tokenization, 2) stop words, 3) bag of words, 4) lemmatization, 5) normalization. Each had its own purpose and benefits. For consistency the normalization is done where texts are lowered. The tokenization was to create a list of individual words. Lemmatization was to reduce the tokenized words into their base form like "wording" to "word". Stop words was to remove any redundant text that had no value like ',', '?. Then both the tokenized and lemmatized words are taken to build the bag of words. With it a vector is created for every text in the list of texts. Where it gives a binary output, a 1 whenever a text exists in the system and a 0 if not. For training purposes neural network is used.

With the help of tensor-flow and its keras library was the creation of the neural network. This is done to train the data. Keras was powerful, as it is a deep learning tool to create, train and evaluate the neural network. Furthermore, the sequential model was the structure of the neural network. The structure consisted of layers that were used such as dense, drop out and activation functions. The dense layers calculate and learn the weight and bias in the process of training. The dropout prevents any overfitting making the network more robust. Activation functions bring in complex non-linearity which neural network learns. For the chatbot the input and output relation and its prediction are taken in. The activation functions used were relu (rectified input unit) and softmax. Relu performs well with gradient descent and is swift in its performance. Softmax on the other hand do well with classification of multi-class. Compiled with the stochastic gradient descent (SGD) optimizer, categorical cross-entropy loss function and metrics fitted into a training data. SGD is an algorithm to decrease the gap for loss or

error in the training process. The categorical cross-entropy loss function is good for classification of multi-class. The values for the learning rate, batch size, number of epochs play a huge role in the accuracy of the neural network. The value "x\_train[0]" refers to the first data point in training data. The value '128' refers to neurons. As for the gradient descent learning rate decides the on when to update the model weights. The momentum adds up to the previous rate to the current. Nesterov decides the update rule on when to add the momentum term. The loss function checks the difference in the predicted probabilities and the actual target labels. The gradient optimizer updates the weights. The metrics which is the accuracy to check the performance of the model. The greater the learning rate the faster convergence. But this also deviates the optimal values. Alternatively, lower learning rate has a much slower convergence and sticks to the minimal value. The values for size also have an impact on the resources and training time. Large batch size can have slower training but good convergence. For prevention of overfitting regularization and dropout rate but affect the learning negatively. All these had an impact on the performance of the model. As a result, with everything the input text data is processed, and predictions are produced. Once ready the train model is then stored to be used. The other libraries contributed for their own functionalities. The pickle for serialization and deserialization enabling data to be manipulated. Json to is also used to handle data. The random to select the data in a random manner.

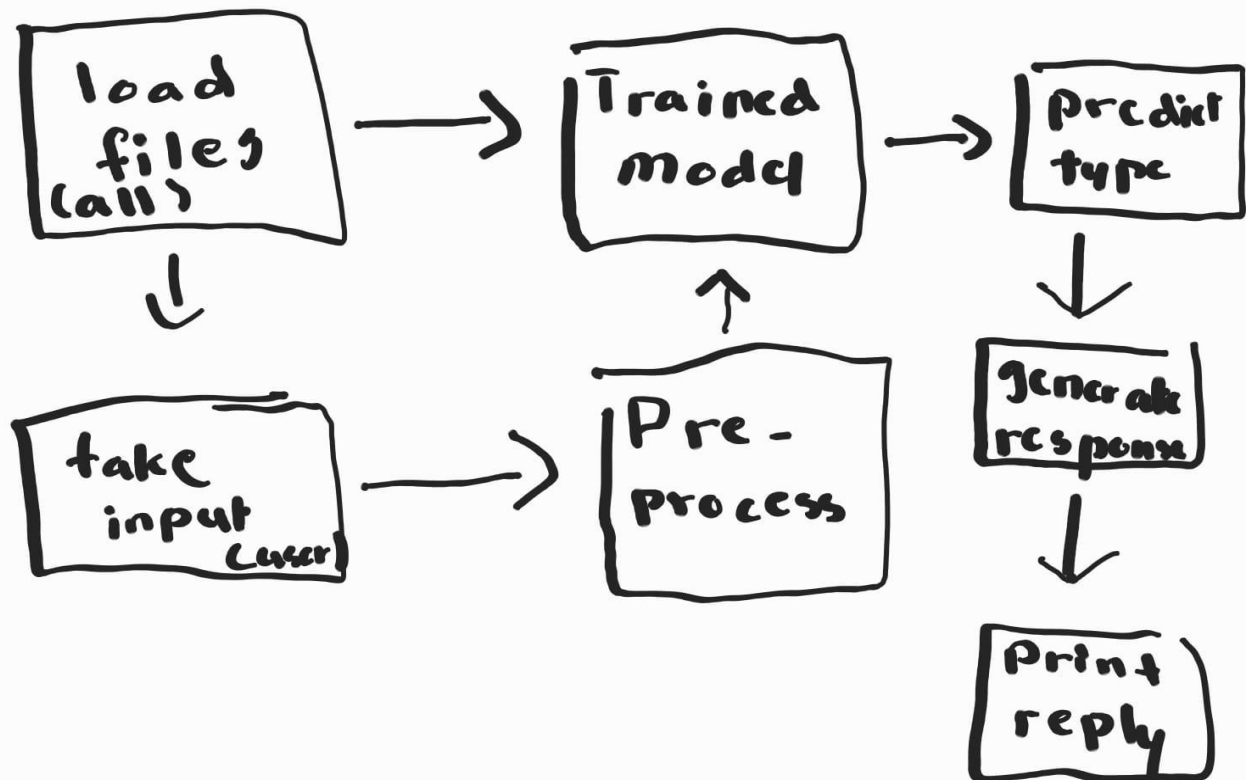
The whole process starts with loading the data file via json. A list is made for the preprocessed data for texts, types, and documents. While iterating over the targets which is associated with the system. The system is tokenized and added to the texts lists. After the tokenized list and the associated tags are added to the document lists. The text list is then lemmatized and stop words are removed. Both the "texts" and "types" list are stored to be of use. The making of bag of words was a representation of the documents for the training data. The output list contained a 0 for every tag and it updated to 1 based on the document in the "docs" list. The trained data is stored in an array to process or perhaps train the neural network. On the other hand, the model for the chatbot takes in an input, a sentence and it is then tokenized and lemmatized. The result would be returned as a list. The bag of words takes in the result and creates a representation of the sentence. Here numpy array was used to return the array. Another function for finding the types takes in the bag of words representation. This is to predict the types of sentences. The reply function decides the random response is chosen from the list. The choice is based on the best prediction type that has the highest weight. The prompt is set to an infinite loop once input, the types are checked to predict the type of

the input itself. When then the reply function generates a response based on the types and the print is used to display. Thus, Jarvis was created.

Overall, the method used to create and design the chatbot is well-structured. The neural network can read in between the patterns and train itself. As a result, the predictions are produced while checking the input. There is no need to manually check and classify the texts. The chatbot is so capable and powerful as it can interpret the input and generate relevant responses. It can evolve itself and is very swift in its process.

### Diagram

The figure below shows a brief overview of the whole chatbot process where all the files are taken in the targets, texts, types, pre-trained model. Once the user inputs a text it is then preprocessed and is looked at via the neural networks training model where it is then predicted what the response should be. The reply is then displayed.



## Evaluation of the bot and analysis of its strength and weakness

Jarvis is so robust and what makes it robust is its capabilities, efficiency, and effectiveness. With the help of nltk the text is understood properly. The automation as mentioned before, the ability to make predictions and the text classification. Without the need for manual manipulation of data. Generation of responses while the other processes take place. No matter the size of the data the neural network can learn and make predictions with utmost accuracy. As a result, the chosen response has the highest accuracy. Accuracy is significant for the credibility of the model. To understand the context of text is difficult. So here the system is followed to get the gist of the text. The size of the data can be much higher in terms of volume. So, the neural network is also scalable as it can handle large amounts of data. The best part is the data can be easily changed which makes the model adaptable. Any data can be used for its knowledge. For instance, if the chatbot is chat support for an airline. The data should be about information regarding the airline and what service the chatbot can be of. This makes it so powerful and adaptable as it can be implemented anywhere and for anything.

The new data would be pre-processed and put into the training process. Though for the neural network some values must be changed based on information of the new data. Values such as updating the metrics, weights and biases that will go along with the specific data. Now this is only for the neural network and its accuracy. Neural network does have its pros and cons. Such as overfit and underfit. In this case it may lead to overfit as where the neural network only memorizes. Whereas the idea is to generalize from the data. Also, neural networks do poorly on small data. A large amount of data is a must for accuracy. Although the chatbot can be very useful. It does have its disadvantages. The most important is its cognitive abilities. Limitations in cognitive abilities meaning it only depends on the data given. Outside of the data the chatbot would respond inaccurately or just halt. But with large sets of data Jarvis can be more capable and powerful.

## Sample dialog interactions

Jarvis can take in user's name and operate accordingly. It can be asked anything within the knowledge or data given to him. This limits its cognitive abilities. As the figure below shows a normal interaction with the chatbot. Once asked how the chatbot feels it responds with the relevant answer. The data also consists of information regarding sports. When asked for specific information about a player in football it replies with the answer. The figure also shows some details of which is "1/1" telling 1 of total batches of

CS 4395.001

Motalib Rahim

Mxr170012

Dr. Karen Mazidi

Portfolio: Evaluation and report

data has been processed. With a total time to process that batch of 0 s and 58ms/step average time to process each batch. This shows how fast the processing speeds are.

```
This is Jarvis, ask away!
Hi Jarvis, I am Sam.
1/1 [=====] - 0s 58ms/step
Hello!
How are you?
1/1 [=====] - 0s 32ms/step
Excellent!
Who is Tom Brady?
1/1 [=====] - 0s 21ms/step
Tom brady is the most famous player.
I like playing piano.
1/1 [=====] - 0s 19ms/step
Me myself, I like to chat
bye!
1/1 [=====] - 0s 20ms/step
speak soon
```

## Appendix – knowledge base

The knowledge base acts as an information resource for the chatbot. The chatbot can easily educate itself. For Jarvis the knowledge base helped the model to understand the input and create responses accordingly. The knowledge base is leveraged that contributed to the relevancy as the chatbot used it as a reference. The “targets.json” file was the knowledge base for the chatbot. The reply function uses it to produce the replies while keeping track of the predicted types.

## Appendix – sample user models

When a user interacts with the chatbot a user model is created. The specific user’s name would be stored in the user model. Along with the user’s personal information collected from the conversation. The user’s interests are also taken in. Together all are saved under the user model. This in turn enables the chatbot to make the conversation more personalized. Through this the conversation is livelier for the user. The conversation would be based on the specific user.