INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1217)
Available online at: https://www.ijariit.com

# Aura – Virtual Personal Assistant

*Shubham*
*shubhamgheek1@gmail.com*
*Atria Institute of Technology, Bengaluru, Karnataka*

*Supriya Rani*
*ranisupriya34@gmail.com*
*Atria Institute of Technology, Bengaluru, Karnataka*

*Sanskar Srivastava*
*sanskarsrivastava30@outlook.com*
*Atria Institute of Technology, Bengaluru, Karnataka*

*Pallavi T. P.*
*pallavi.tp@atria.edu*
*Atria Institute of Technology, Bengaluru, Karnataka*

## ABSTRACT

***Today, with the advancement of technology, virtual personal assistant plays important role in day-to-day life from searching information from web to automating certain tasks. There are various virtual personal assistants that exists like Google Assistant, Siri, Cortana. Laika is one such text based virtual assistant which can perform certain tasks. Laika is simple with user-friendly interface and can performs simple tasks like making short notes, telling jokes, greetings and suggest certain tasks. With the increasing popularity of Virtual Personal Assistants, the interaction between humans and their machines has increased significantly. This system can interact with humans and machines without being limited by their physical appearance or capabilities. This system is mainly used in various industries such as education, medical assistance, and home automation. Due to this, the concept of Virtual Personal Assistants has gained widespread acceptance.***

***Keywords***: *Natural Language Processing (NLP), Stemming Techniques*

## 1. INTRODUCTION

Virtual Personal Assistant is a software program that enables an end user to interact with a virtual assistant in a virtual world. Virtual Personal Assistant is a tool that enables users to make various tasks happen instantly. It works seamlessly on real time. This paper aims to develop a dialogue system that can handle multiple user input modes, such as speech. It can then augment and replace human-generated customer service. React is used for creating a user interface. styled components are used to give a stylish and attractive look to the interface. Natural language processing tool helps in interaction between computers and humans. In virtual personal assistant mode, the process is used to demarcate and classify a string of input characters. Stemming process is a natural language processing technique that involves reducing a string of words to its word stem. This procedure is performed by two algorithms known as the Lancaster and Porter. NestJS is a server-side framework that enables building web apps with minimal programming experience. RXJS is an event-based library that enables creating and managing event-based programs.

Virtual Personal Assistant will process the string or command given, by using NLP engine which helps system to understand the natural language. Natural language which is used by the humans to communicate with others. Segmentation process is used for splitting the input query into several small meaningful segments it would be sentence, phases, words. Virtual personal Assistant will process that segment and provide relevant answer according to the query. It is a software which is used for doing various monotonous work without any human effort.

## 2. LITERATURE REVIEW

Natural Language Processing will help the machine to understand the human language which is commonly used by the user to communicate. The input which is received by the system from the user is in an unstructured format that cannot be understood by the system hence Natural Language Processing will take the unstructured query from the user and convert it into a structured format. NLP extracts the important word and pattern to convert the unstructured query to a structured format with the help of NLP techniques. NLP does not directly understand the meaning of user query it undergoes several sequences of process. For understanding user query NLP must understand the meaning of every word of the sentence hence for that tokenization process is used.

### 2.1 A review on word segmentation

Virtual personal Assistants split the input query into several segments which are used to get the exact meaning of the Query this process is called tokenization. Tokenization split the sentence into words and punctuation and provide it as an individual unit. The tokenizer considers dates, numbers, etc, in the decimal format else it will not consider and lose its meaning in the sentence. **Mohammed Javed et al. [1] [2015]** explained the method to implement tokenizer in his algorithm he mentioned the way to calculate the character space in the sentence. The character space will include all types of gaps

between the character. In his algorithm, all the gaps will be considered. The algorithm is based on the number of gaps between the sentences that will be calculated after the calculation of gaps an average will be taken of gaps which are also called average gaps. It is calculated to find out the mean average between the character and sentence. This average will then apply to the sentence which is to be segmented. The gaps in the sentence are always more than the average gaps this is called the point of tokenization. Usually, the gaps in the sentence are more than the average gaps present in the sentence.

**Naeun Lee et al. [2] [2017]** advance the algorithm of word segmentation using Natural Language ToolKit (NLTK) which is basically a python package that is used to provide service to Natural Language Processing (NLP). It has an inbuilt tokenizer in the python package. Users can import the package and use the tokenizer as required by calling the function. NLTK provides several tokenizers such as words, classic, standards patterns, etc. the tokenizer commonly used is the word-Punkt tokenizer which basically split the sentence into blank spaces. In terms of accuracy, speed, and efficiency NLTK is very convenient, and also NLTK not required any algorithm because the execution process of a package is processed in the backend.

**Tao Jaing [3] [2011]** proposed an algorithm for word segmentation. This algorithm works to identify the spaces between the characters, this algorithm is known as Conditional Random Fields. With the help of this algorithm, the system can identify the gaps between the test query system provide a threshold to maintain a gap between the sentences. If there is any gap that didn't follow the specific threshold provided by the system, then it gets a splitter at that specific point. CRF needs to be trained with a high amount of dataset which causes time consumption. All the method mentioned above the most reliable and convenient method is NLTK compared to the other two. As NLTK will not require any implementation of the algorithm all aspects will take care of by the packets present in it. NLTK is better in accuracy, speed, and multiplicity provided by the packet is more convenient than the other two algorithms.

### 2.2 A review on POS Tagging

This procedure involves marking individual words with part-of-speech tags. The tags are used to denote the importance of the word in a sentence. The latent analogy for POS Tagging is a method that combines the concept of latent semantic mapping with the concept of implementing a corpus-based approach for tagging. This method involves training the corpus on the given sentences and then analysing them to find the exact same intent matter.

POS Tagging is used for providing or assigning grammatical interpretation to every word present in the sentence and that interpretation also considers Parts-Of-Speech Tags. These interpretations will help to understand the grammatical importance of words present in the sentence based on the province of words in the sentence. Some of the common tags include noun, pronoun, verb below-

**Jerome R. Bellegarda [4] [2010]** The concept of latent analogy for POS tagging is proposed. This method uses the latent semantic mapping technique to identify the sentences that are closest to the test sentences. The LSM maintains the tagged trained corpus. After that, a new query is given to LSM for tagging. After that analysis is performed on the given query to review that which training data is closest to the given set of a query. This procedure is known as the Sentence neighbourhood. When the matching query is found from the trained data, then POS tagging will get attached to the query and then mapped the test query.

**Liner Yang et al. [5] [2018]** explains the concept of implementing the POS Tagger is simple and can be done using Neural Networks. The algorithm consists of several hidden layers. Each layer is used to determine the appropriate word count for the given sentence. The next layer should provide the same tags as the previous layer. This method works by keeping the previous layer's tags the same. The NLTK tagger has a set of predefined tags and a trained data base. It tests the sentence and then applies an appropriate tag to it using its own neural network algorithm.

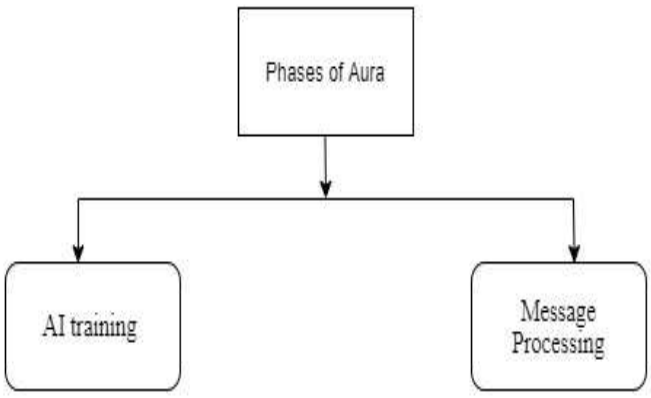
### 2.3 A review on Synonym and Pattern Recognition

The user query which we take as an input is not exactly matched with the query present in the database but in a database, there will be a sentence having the same intent. After getting the user intent system will search in the database for the specific pattern to retrieve the relevant solution for that specific query asked by the user. The user query may consist of a synonym of that particular word that is the same sentence can be written in several ways by using alternative words, but the intent is the same. This makes it necessary to understand the synonym by the system which helps the system to give accurate data from the database. There are several algorithms which is used for identifying synonym some of them are below –

**LinHua Gao et al. [6] [2018]** explains the conventional lexicon method for determining the synonym. In this method, a collection of a dataset which consist of synonyms for important keyword necessary for that domain. Query sent by the user will undergo with several process where query checked if it matches with the synonyms or not. If the query detected with synonym, then checked in dataset for the same instant. All feasible synonym of that word will be checked in the main directory. The query closest to the user query is extracted. This method is slow hence consume more time then usual and required more storage and have more complexity.

**Sijun Qin [7] [2015]** proposed a method in this method, the words having the tags as noun, verb, and adjective are marked as positive tags while the others are negative tags. The overall feature polarity is then determined by taking the word's overall feature definition and grouping it together. The main idea of this method is to classify the words having the tags as positive or negative tags. The overall feature polarity is then determined by using the POS tags.

## 3. PHASES OF AURA



### 3.1 Model Training

Training takes place once when the NestJS server starts. Each time you modify the code and save the file, the server automatically restarts (thanks to the live reload of the yarn start: dev command) and the training is re-executed.

First, we discover all the methods that have been decorated with @MessageHandler. To work, it is necessary that the class is used as a provider in some application module. For organizational reasons, they must be modules within skill module.

For each method, we call the *registerMessageHandler* function, it will take the phrases that were used as a parameter of @MessageHandler and add to the AI training using the Node-NLP package. Basically, Node-NLP needs an input and an output string. The input is the phrase that is in @MessageHandler, and for the output a string is made using the class name and the method name in the following format: "parent Class Name. Method Name". This way, Aura will later know which function to call when receiving a message.
Finally, the network is trained using train() method.

### 3.2 Message Processing

Aura firstly takes queries from the user which is in normal language through which we generally communicate. Aura processes the query with the help of natural language processing and tries to understand the intent of the user. For better understanding and the intent of the user, Aura undergoes several processes explained above. Aura starts processing the user query by tokenization process where the query segmentized into several small units. For the tokenization process, we use NLTK which is more reliable and fast with accurate results.

After the tokenization process Aura proceeds with the POS tagging process where the system tries to understand user queries in the grammatical form whether it is past, present, or future that is when we say to Aura that “please remind me after 1 hour” the query will be processed in POS tagging and it will remind after 1 hour hence the query is for future purpose. Aura processed the user query in POS tagging and then forward the query for synonym and pattern recognition.

In synonym and pattern recognition the query that the user has given us may not be an exact match to the query in the database. Instead, the query may have the same intent. Understanding the intent of the user query helps the system to interpret the query properly. The conventional lexicon method is used for determining the synonym of a word. In this method, the query is sent with a collection of synonyms for that particular word. After the query is analysed, the closest synonym of that word is extracted. The words with the tag’s "noun", "verb", and "emotion" are marked as positive tags while those with the tags "inject", "reject", and "word" are marked as negative ones. The overall feature polarity of the words is then determined by grouping the words.

## 4. CONCLUSION

Virtual personal assistants are very helpful in doing tedious repetitive tasks. Aura is one such text-based Virtual Assistant and can perform various tasks. It is very easy to install and use and also easy to add new skills as per the user's requirements. Aura offers a simple, clean, and user-friendly interface to interact with users. Aura is platform independent as its web application and it can run very easily on any platform.

Aura can be easily modified according to the user's requirements and does not require high-end hardware or software requirements. With the advancements of technologies, Virtual Personal assistants started playing more important roles in an individual's life. As of now, Aura can also fulfill most of the user's requirements and can be upgraded to performs complex tasks.

## 5. REFERENCES

[1] Mohammed Javed, P. Nagabhushan, B.B. Chaudhari, “A Direct Approach for Word and Character Segmentation in Run-Length Compressed Documents with an Application to Word Spotting”, 13th International Conference on Document Analysis and Recognition (ICDAR), 2015.
[2] Naeun Lee, Kirak Kim, Taeseon Yoon, “Implementation of Robot Journalism by Programming Custombot using Tokenization and Custom Tagging”, 2017.
[3] Tao Jiang, Hongzhi Yu, Yangkyi Jam, “Tibetan Word Segmentation Systems based on Conditional Random Fields”, 2011
[4] Jerome r. Bellagarda, “Parts-Of-Speech tagging by Latent Analogy”, IEEE Journal of Selected Topics in Signal Processing, Vol. 4, No. 6, 2010.
[5] Liner Yang, Meishan Zhang, Yang Liu, Maosong Sun, Nan Yu, Guohong Fu, “Joint POS Tagging and Dependency Parsing with Transition-based Neural Networks”, 2018
[6] LinHua gao, HePing Chen, “An Automatic Extraction Method Based on Synonym Dictionary for Web Reptile Question and Answer” 2018.
[7] ] Sijun Qin, Jia Song, Pengzhou Zang, Yue Tan, “Feature Selection for Text Classification Based on Parts-Of-Speech Filter and Synonym Merge”, 12th International Conference on Fuzzy Systems and Knowledge Discover (FSKD), 2015.
[8] ijstr.org/final-print/july2019/A-Literature-Review-On-Chatbots-In-Healthcare-Domain.pdf