AIML Based Voice Enabled Artificial Intelligent Chatterbot

Imran Ahmed and Shikha Singh

*University Institute of Engineering and Technology,
CSJM University, Kanpur, Uttar Pradesh
imranahmed2k9@gmail.com, sshikha462@gmail.com*

Abstract

This paper shows the implementation of an artificial intelligent chatterbot with whom human can interact by speaking to it and receive a response by chatterbot using its speech synthesizer. Objective of this paper is to show application of chatterbot that can be used in various fields like education, healthcare, and route assistance. It is statistical model and chatterbot is based on AIML (Artificial Intelligent Markup Language) structure for training the model and uses Microsoft voice synthesizer for providing speech recognition system and natural language processing.

Keywords: chatterbot, speech recognition, AIML, intelligent system, natural language processing

1. Introduction

By using natural language user can interact with system very easily even if user doesn't know how to read and write so, it becomes incentive for user to use this system. The model receives response from user either in form of text or in form of speech and thus responds to user in both format text as well as sound thus facilitating simple and user friendly request and response system. Developing such system raises many issues as accent varies from person to person resulting in understanding what is meaning of user request. The performance also varies according to amount of corpus available for training the model.

Concept of chatterbot came into existence with "Alice" which is used to receive question from user and it was based on pattern recognition. Large amount of sentences were involved in training the model. The response provided by this bot was very monotonous which gives feeling of chatterbot being a machine and it's functionality were very limited as response for each input was predefined.

Chatterbot is sub field of AI. AI is artificial intelligence which is a branch of computer science whose main focus is to develop and study intelligent programs and machine which can be helpful to people in their daily work and make their life easy. Artificial intelligence is not limited to chatterbot it involves NLP (natural language processing), Social aware intelligence, knowledge representation and inferring from that knowledge.

This is bot system implemented by using AIML. AIML is artificial intelligence markup language which is another flavor of XML (extensible markup language). AIML has been used in this model such that it parsing and declaration different kinds of sentences can be easy. AIML is base of chatter bot brain.

For the purpose of natural language understanding, Microsoft speech recognition is used which helps in speech recognition and speech synthesis for speech to text and text to speech to make it more user friendly for people.

2. Related Works

The bot [1, 2] system began in early nineteenth century where first chess playing machine was built such that a system act as a player and other player an human can play against it, such that system giving a feeling of another person existence. Here is different chatter bot build for various purposes.

ALICE [7-9] was implemented with pattern matching algorithm which was as simple as string matching technique. ALICE takes the text as input and produces output as text which was like question and answer based system. Whereas another chatter bot[5][6] which was build earlier known as Elizabeth requires set of input rules, keyword patterns for identifying input and output rules to produce required response.

The recursion [3, 4] based approach was implemented in ALICE chatter bot and it is a vital feature for the model to work properly. Since it was recursion based so calling function by itself made it easy to respond appropriately but it overflow the stack. However rules in other bot may lead to language specific chatter [23-25] bot such that it sticks to particular language therefore to make it language independent one has revised all the rules to work it properly and produce output.

ALICE can combine two answers in the case of splitting during the normalization process, or by recursive process. In Elizabeth [10-12] we cannot partition the sentence in two parts and then combine the result.

Pattern-managing matching process is the vital feature of ALICE chatter bot, but it is simple and used algorithm is depth-first search which result in producing no output also. DFS [15, 17, 18] tries to find algorithm finds the longest common subsequence pattern matching the required text in the database where other one Elizabeth produce output according to the first keyword [21, 22] matched.

Both systems are memory based. Elizabeth's dynamic process, it can perform other actions while other conversation is in process, is considered a excellent feature which is not in ALICE.

If the inputs are repeated during the conversation, Elizabeth gives different answers by use of different random selection responses from the list. ALICE use random selection which can lead to duplicate responses.

Since Elizabeth is based on specific language [13, 14, 16], therefore rules written for it also language based, therefore it is difficult to other language user to interact. However language independent was ALCIE but it was string matching based which was not fruitful.

Elizabeth requires large amount of analysis tables for the steps to perform in matching [19, 20] a text or string, therefore for the developer how answer question is carried out.
From the above it can be understood that none of the previous bot was perfect such that it raises a need of new chatter bot which is language independent and it does not entitle to input a text by writing it has to produce output by just voice enabled and user doesn't has to read also it should also be voice response such that it becomes user friendly for user and it ease of use should it uniqueness.

3. Proposed Work

The architecture of bot system is divided into following modules which helped in implementing this chatterbot.
   i)   AIML
   ii) Microsoft Speech recognition
   iii) Corpus
   iv) Brain
   v)   Bot engine

i). AIML

Artificial intelligence markup language is a flavor of XML, extensible markup language and it is custom form XML. The most important parts of an AIML documents are:

    a) <aiml>

    b) <category>

    c) <pattern>

    d) <template>

There are many others tags which are used to describe a scene for bot to understand and respond appropriately. It's main objective is to simplify pattern matching, request response from knowledge to be carried out and as well as to exist in the web development where information can be sent via HTTP. AIML consists of aiml objects and these aiml objects consist of topics and categories which contain either parsed or unparsed data from their information is extracted. Data which is parsed consists of characters and this character data is being parsed by the aiml parser or interpreter. An Aiml interpreter can be describe as a one who scan aiml objects and provide a response according to that characters. A parser is biggest part of application like bot program. There are some goals for design that need to be existed for betterment.

    1) It should be simple and easy to learn for people and machine.

    2) With minimal encoding and decoding should provide precise result.

    3) A simple program can be used to parse aiml documents.

    4) There should not be any dependencies for particular language.

AIML Objects

AIML Root

It is the root element from where there is initiation of aiml document. It can be compared with html document where writing begins after specifying its html file same is the case here

    <aiml>….</aiml>. So it also have opening and closing tag also. AIML
    TOPIC

It is first level optional item that specify category of elements. It has following attributes namely 'name'.

    <aiml:topic name='topic name'>…</aiml:topic>

AIML Category

It is first level if topic not present else it is second level element that contains only one pattern and exactly only one template. It doesn't contain any attribute.

    <aiml:category>

    ……Content for category…

    </aiml:category>

AIML Pattern

It is an element which contains mixture of pattern. Pattern should appear inside the category.

> <aiml:pattern>
>
> …..Pattern…..
>
> </aiml:pattern>

AIML Template

It is an element which is inside category element. Template must follow pattern-side element.

> <aiml:template>
>
> …..Template…
>
> </aiml:template>

Now after reviewing some of basic and most important AIML objects. There are some atomic template elements.

Star: Star element indicates that AIML should replace value by a particular wildcard from specified portion of match path and returning the required template.

> <aiml:star index='integer value to be stored'/>

That: It tell AIML to replace material produce by bot in previous output.

> <aiml:that index='single integer| comma separated integer value'/>

ii). Natural language Processing

Before speech recognition can work, there arises need of natural language processing. One need to understand how language should be interpreted as there is ambiguity in processing any language.
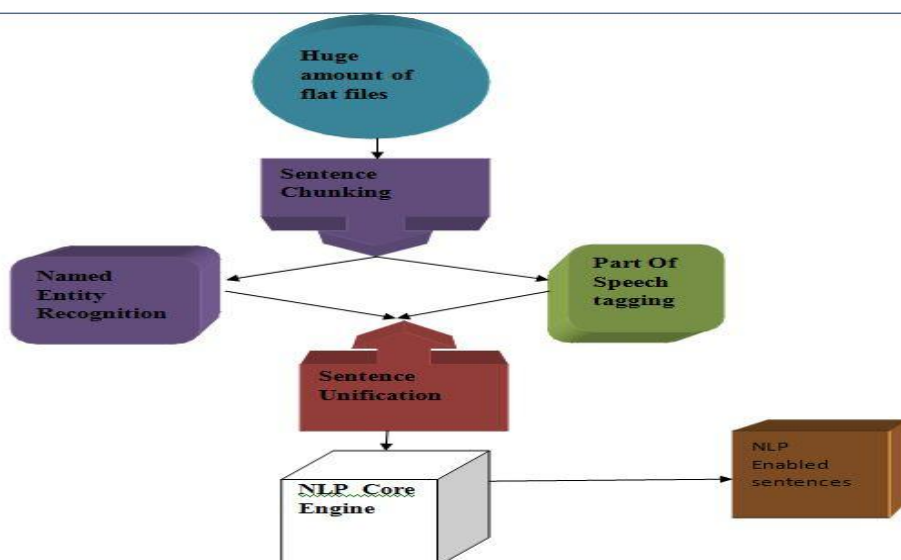

Figure 1. Processing of NLP

Therefore, huge amount of information or corpus is required such that an accurate NLP engine can be build. Such that to NLP workout, it need following things like flat files which can be in any format such that like xml, json, or any simple text file. After this obtaining the corpus, sentence chunking is done so that each sentence can be processed so that separately and they are further divided into named entity recognition so that it can find the name that occur in text and they should not be processed as they are names. Further remaining part of sentences are tagged as part of speech to make sentence more information valuable. After processing, all NdER and POSed , sentences are ready to be combine for NLP Engine and after engine processing NLP can be done successfully.

iii). Microsoft Speech Recognition

This speech recognition consists of TTS and STT. TTS is text to speech system in which text is converted to speech or voice as one can say. STT is speech to text, it sound or voice receive from user is converted to text for processing of information by bot.
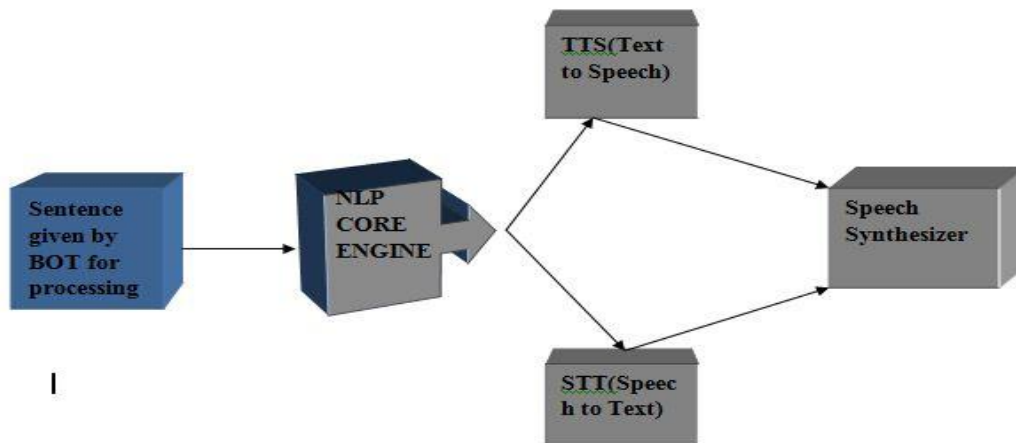
Figure 2. TTS and STT

Firstly , sentence given or receive is passed to NLP and after that NLP core engine process these sentences and depending upon the situation it is processed by STT or TTS and it is moved forward to speech synthesizer to respond.

i). Corpus

For the corpus data is being collected from various internet resources such as Alice aiml files, Howie aiml files and Professor1.0 an Professoor2.0 updates aiml files is being collected. It enables to build a huge corpus which consists of roughly around one million aiml sentences to respond accurate to question or in any situation given to the bot.

ii). Brain

The brain file size is around ninety mega bytes which is compressed form of corpus and it is build in form of tree, the information is processed in form of depth first search such that it find required information and as well as it doesn't get stuck if information is

not found for large amount of time in that case brain should respond with nearly most appropriate data or text.

iii). Bot Engine

The bot engine is the main architecture which consists of following components like

      a) Kernel

      b) AIML Parser

      c) Utility Manager

      d) Word Substitution

      e) Pattern Manager

Kernel

Kernel class has most of the implementation and is responsible for all processing. It requires all other classes for the processing to happen, it require AIML parser, word substitution, utility manager and pattern manager for matching. It include following modules like sessions, bot predicates, word substitute, element processor. It also include brain file , if brain file is provided kernel try to load the brain and if brain file is not their then it attempts to load all aiml files.
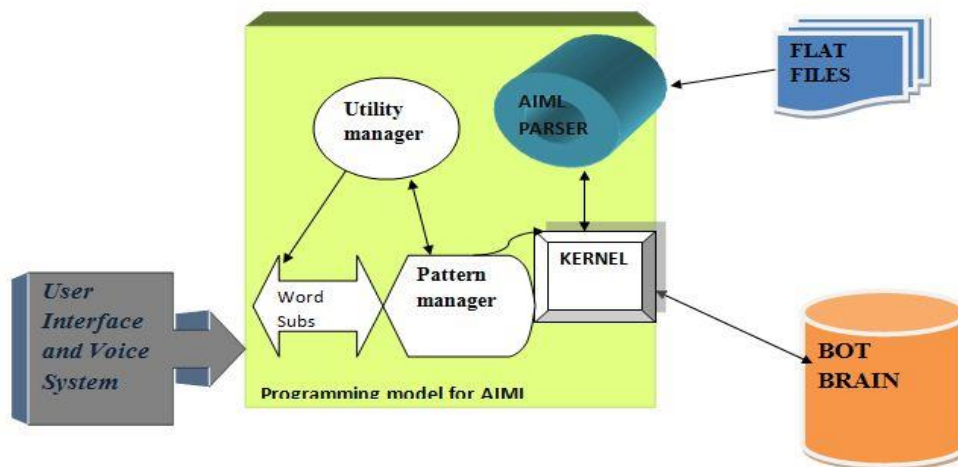

Figure 3. Architecture of Chatter Bot

AIML Parser

AIML parser is just like xml parser or say more particular it is sax parser. First need to get name of current unknown element. And set skip current category to true and also count the number of errors occurred in the aiml document to test the authenticity of document, if the number of errors occurred are greater than number of threshold value skipped that aiml file.

Utility Manager

It includes the assorted utility function which are used in PyAIML package for performing various functions. Its task is to split a given string into list of sentences. Therefore it helps in splitting string and performing all the required processing by AIML parser or that is sax parser to do work.

Word Substitution

It implements word substitution class adopted from python cookbook module for simplicity and accuracy. This class is used like a dictionary to add pairs of word anywhere in dictionary before or after. It also contain another method called as substitution method for substituting words in the string. All matching done for replacing is done case sensitive and intelligent.

Pattern Manager

This class implements pattern matching algorithm of AIML which is makes pattern matching more intelligent and accurate. The algorithm follows like this:

a) Calculate no. of templates currently stored.

b) Set the name of chatter bot for its uniqueness and name must consist of single word.

c) If any pattern found, dump it for debugging purpose.

d) If any further pattern is found store it in a file specified like a pickle file.

e) After all patterns are stored, restored all patterns for making a brain map

f) After creation of brain , it can used to interact with user.

Implementation

The following chatter bot is created with following tools and the procedure carried out is discussed later. Tools used are the following.

1) Python Interpreter

2) Brain File

3) Cherry Python Framework

4) Microsoft Speech Synthesizer

5) Visual Studio with Iron Python

These tools are used to build the chatter bot with voice enabled which can process natural language also. First of all, python interpreter is foundation of this bot which helps in carrying out process and algorithm. The brain file build for this model is used repository such it does model has not to be train again and again. Cherry python framework so as to make it available online and make it accessible everywhere whether on desktop, laptop or hand held devices. Microsoft speech synthesizer is used for text-to-speech and speech-to-text functions. Visual Studio with Iron python to make it available as windows based application, such that chatter bot is accessible to every individual.

Procedure

1.) First step is to create a brain file, it can be created by using aiml file.

2.) If brain file is not present then create a brain.

3.) If brain file is present then load the brain file in the model.

4.) After loading the brain, bot wait for the request from user or client.

5.) User can provide request either in format of voice or text.

6.) Model receive request and forward to pattern manager.
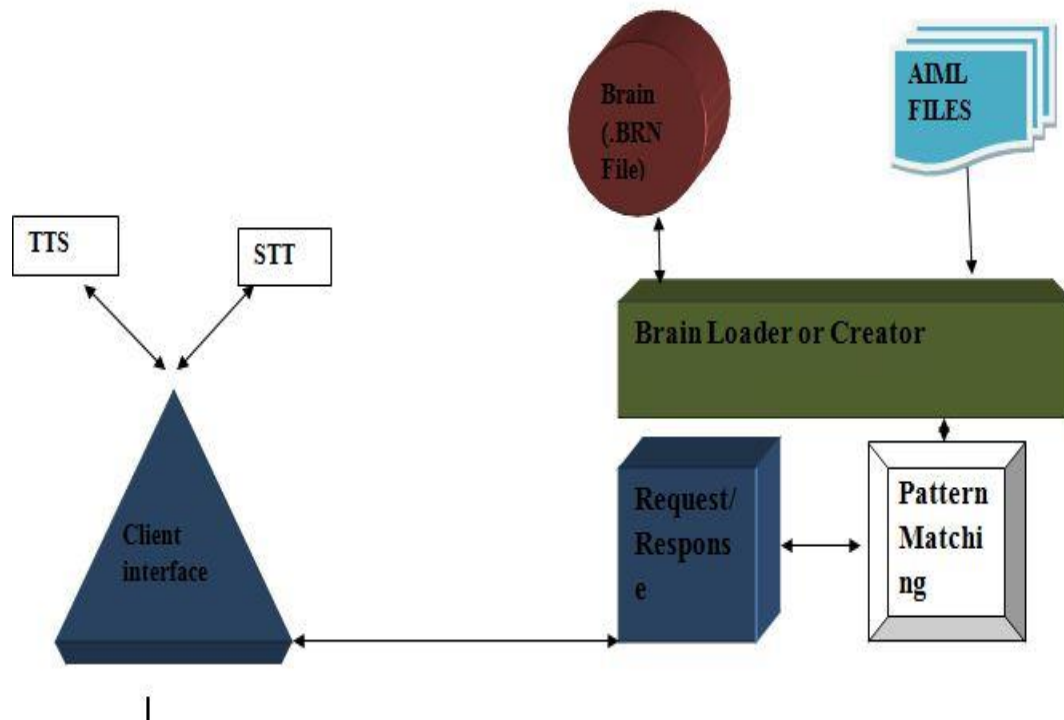
Figure 4. Flow Diagram of Request and Response

7.) Pattern matching algorithm is applied and send to the brain.

8.) The model use the brain and get appropriate response and forward to user.

Result and Analysis

The chatter bot performed up to the marked as in case of text provided, results produce were 100 percent accurate and more reliable. But in case of voice enabled input due difference in accent of user and Microsoft speech synthesizer model whose accent is based on UK or USA users. Thus result obtain were close to the appropriate answer. Thus from above it can be consider that speech synthesizer need to be model for all kinds of accent such that it process the voice input more correctly ,therefore chatter bot model can produce more accurate results.

Conclusion and Future Work

Chatter bot is proved to be a boon in any industry where one can seek assistance without need of human. Bot itself act as human mind to provide accurate information. In education, it can be better as a teacher, in healthcare industry it can provide appropriate details of medicine to be consumed for particular disease and for traveler it can be a guide assistance to provide him shortest path for travelling particular with minimum expenses.

The future work is implement accent free speech recognition system, such that bot perform more accurately to any individual, as well as it should be made language

independent such user can interact not only in English in other language also like Hindi also.

References

[1] Weizenbaum, J. ELIZA – A computer program for the study of natural language communication between man and machine. Communications of the ACM, 10(8):36–45 **(1966)**.

[2] Cliff, C. and Atwell, A. Leeds Unix knowledge expert: a domain-dependent expert system generated with domain-independent tools. BCS-SGES: British Computer Society Specialist Group on Expert Systems journal, 19:49–51 **(1987)**.

[3] Wallace, R., Tomabachi, H., and Aimles, D. Chatterbots go native: Considerations for an eco-system fostering the development of artificial life forms in a human world. Published **(2003)**, online: http://www.pandorabot.com/pandora/pics/chatterbotsgonative.doc.

[4] Bran, E. Chabot's in der Kundenkommunikation (Chabot's in customer communication). Springer, Berlin **(2003)**.

[5] Abu Shawer, B. and Atwell, A Using the corpus of spoken Afrikaans to generate an Afrikaans Chabot SALALS Journal: Southern African Linguistics and Applied Language Studies, 21:283–294 **(2003)**.

[6] Abu Shawer B. and Atwell A. Different measurement metrics to evaluate a Chabot system Proceedings of the NAACL'07 Workshop: Bridging the Gap: Academic and Industrial Research in Dialog Technologies. Pp.89-96, ACL**(2007)**.

[7] Abu Shawer B. and Atwal A. Machine Learning from dialogue corpora to generate chatbots. Expert Update journal, Vol. 6, No 3, pp 25-30 **(2003)**.

[8] Steven Birds, Ewan Klein and Edward Lapper. Natural Language Processing with Python by O'Reilly Media, Inc. **(2009)**.

[9] Allen Downey, Shroff Publisher "Think Python" **(2012)**.

[10] Python: http://www.pandorabot.com/Pandora / pic / wallaceaimltutorial.html"

[11] www.alice.com/aiml.html.

[12] Html, CSS, J-Query: "www.w3school.com".

[13] David Sawyer Mc Fernand, Publisher, O'Reilly "JavaScript and jQuery 2nd Edition The missing Manual" **(2011)**.

[14] IRS E-learning Glossary. **(2005)**. [Online]. Available: http://www.irs.gov/opportunity/procurement/article/0,,id=128685,00.html.

[15] A. Gracy, T. Butler, Beyond Knowledge Management: Introducing Learning Management Systems, Idea Group Publishing, **(2006)**.

[16] W. K. Harton, Designing Web-Based Training: How to Teach Anyone Anything Anywhere Anytime, John Wiley and Sons, **(2001)**.

[17] T. Murrari, "Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art", International Journal of Artificial Intelligence in Education, 10, 98-129,**(1998)**.

[18] A.L.I.C.E. Artificial Intelligence Foundation. [Online]. Available: http://www.alicebot.com/

[19] A. M M. Naves, I. Diniz, I., F.A. Barros, "Natural Communication via AIML Plus Chatterbot", V Symposium on Human Factors in Computers Systems (IHC 2002), Fortaleza – CE, 387, **(2003)**.

[20] A. M Turing, Computing machine and intelligence, **(1950)**.

[21] LOM (Learning Object Metadata). Draft Standard for Learning Objects Metadata. **(2002)** [Online]. Available: http://ltsc.ieee.org/wg14/files/LOM_1384_12_1_v1_Final_Draft.pdf.

[22] M. E. Batman, Intention, Plans, and Practical Reason, CSLI Publications, **(1999)**.

[23] Program D. Getting Started with Program D. **(2005)** [Online]. Available: http://www.alicebot.org/resource/program/readme.html.

[24] R. Asaleeson, N. T. Schulte, Foundation of AJAX, Après, **(2005)**.

[25] Fra Ling. [Online]. Available: http://grraf.epsevg.upc.es/freeling/.

Authors

Imran Ahmed, he is currently pursuing post-graduation at VIT University Vellore, Tamil Nadu in Computer Science and Engineering stream. He has done Bachelor of Technology in Computer Science and Engineering from University Institute of Engineering and Technology, CSJM University, Kanpur. His major interest work area is Artificial Intelligence, Cloud Computing and Web development.

Shikha Singh, she is currently pursuing post-graduation at VelTech University, Chennai, Tamil Nadu in Electronics and Communication Engineering stream. She has done Bachelor of Technology in Computer Science and Engineering from University Institute of Engineering and Technology, CSJM University, Kanpur. Her major interest work area is Computer Graphics, Artificial Intelligence, Web Development and Real Time Operating Systems.