International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

CHATBOT-A JAVA BASED INTELLIGENT CONVERSATIONAL AGENT

Mr. L S Chetan Rao, Mr. Dinesh Kini, Mr. Kanthraj S, Mr. Kiran Kumar N
Department of Computer Science and Engineering, The National Institute Of Engineering, Karnataka, India

---

Abstract - *A conversational agent, a computer program designed to stimulate intelligent conversation with users using natural language processing (NLP). A Chabot is a computer program which conducts a conversation via textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Chatbots are typically used in dialog systems for various practical purposes including customer service or information acquisition. Some Chatbots use sophisticated natural language processing systems, but many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database using ontology based approach.*

Key Words: robotics, Chabot, intelligent robots, conversational agent, artificial intelligence and ontology

1. INTRODUCTION

"Can machines think?"[1] This is the question asked by Alan Turing which has since spawned numerous, Passionate debates on the subject of artificial intelligence It has also spawned the famous Turing Test, a test Which determines if a particular machine (or algorithm?) can pass as a human. Since its inception, the Turing Test has, in fact, been passed by a few artificial intelligence algorithms. Some of these algorithms represent the latest technology in terms of artificial intelligence.

Artificial intelligence (AI) is part of a broad field called cognitive science, which is simply a study of the mind and the way it works. For the purposes of cognitive science, artificial intelligence is defined as "a codification of knowledge will finally explain intelligence". However, when it comes to software engineering, the purpose of AI is to use knowledge to solve real-world problems. One of these problems, similar to the problem of the Turing Test, is how to make an artificial device or creatures appear more human. To address this problem, a technology has been created called chatbots. These are AI algorithms that process natural language and, using the analysis that results from the processing, output an intelligent response.

Paper layout is as follows: Section 2 highlights literature survey, Section 3 explains proposed system in brief, and Section 4 explains the design on Chabot, Section 5 on implementations and Section 6 accounts for some concluding remarks.

2. LITERATURE SURVEY

It explains the design of a Chabot that is specifically tailored for providing FAQBot system for university students and with the objective of an undergraduate advisor in student information desk. The Chabot accepts natural language input from users, navigates through the Information Repository and responds with student information in natural language.

The design semantics includes AIML [2] (Artificial Intelligence Mark up Language) specification language for authoring the information repository such that chat robot design separates the information repository from the natural language interface component. Correspondingly, in the experiment, we constructed three experimental systems (a pure dialog system associated with natural language knowledge based entries, admin knowledge systems engineered with information content and a hybrid system, combining dialog and domain knowledge). Consequently, the information repository can easily be modify and focussed on particular topic without recreating the code design. Experimental parameters and outcome suggests that topic specific dialogue coupled with conversational knowledge yield the maximum dialogue session than the general conversational dialogue [4] and [7].

This work presents a prototype conversational system enabling human-computer interaction by using natural language expression. As an enhancement to well-known conversational agents like chatbots, in the proposed setting, human-machine dialogue is intended as a query/answer monotonic process aimed at minimizing semantic ambiguity within communication and delivering the required service. When user queries are ambiguous, hence semantically distant from the set of possible recognized interpretations, the system instantiates a dialogue with the user. In this case, the system provides suggestions on how to reformulate the query until a valid form is reached; this feed-back makes the dialogue-oriented interaction process resemble an ordinary chat (in the very restricted domain of system services) but with a machine interlocutor. The popularity of the chat as a synchronous communication instrument lets our proposal be suitable for a great variety of applications.

3. PROPOSED SYSTEM

In the proposed system we are creating a Chabot that can reply to the user in the most effective way. This approach is proposed to model and operate the chat bots.

It uses appropriate mapping technique to transform ontologies [3] and knowledge into relational database and then use that knowledge to derive its chats. The data that is stored in the database is used to map with the queries asked by the user. The user gives the question as a string and that string is taken as a JSON object and that object is processed using various NLP algorithms such as porter-stemming, Wordnet, TF-IDF. These algorithms will provide the keywords and those keywords will be matched to the keywords in the database to extract the required answer string.

3.1 Ontology technique

This technique is responsible on processing user input for getting the needed answer. Several functionalities should be taken into account before start searching for a match in database and then succession in getting the best answer. Such functionalities include input tokenization, stopper filtering, stemming and synonyms handling. Figure 1 illustrates the four functionalities and their sequence.

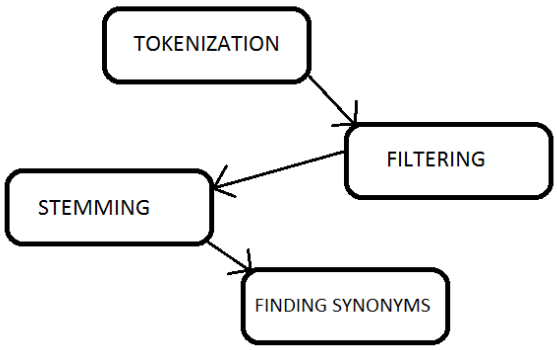


Fig -1: Ontology technique

Tokenization or splitting the input into words involves some operations that are necessary to facilitate the target matching process. A set of splitters will be used in order to break down user input into tokens. Such splitters include: space, punctuations, special symbols and others. Both user input and stored words will be tokenized.

Filtering is the phase that refers to the process of removing set of un necessary words that may exist in user input. The stop list consists of a list of common function words such as determiners (a, the, this), prepositions (in, from, to), conjunctions (after, since, as), coordination (and, or), Also those words which occur more frequently but contribute little meaning like about, them, only etc. Word stemming is an important feature especially when we talk about indexing and search systems. This project makes use Wordnet [6] database for stemming whose complete details are in next section.

There are many available stemming algorithms [5] for example, OntBot can employ one of the most effective and widely used stemming algorithms known as Porter stemming algorithm (or 'Porter stemmer').

Synonyms/alternatives of a word are obtained in order to be checked. This is required if there is no match found between user input token and a stored word one, in this case OntBot will assume that the user may use a synonym words of the stored one so it will get them all. These synonyms will be in turn considered as new entries and one by one searched against the target stored words. If again no matching found, the word will be considered as a mismatched one. Wordnet also going to be used to find words synonyms

3.2 WordNet

It is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. Wordnet is also freely and publicly available for download.Wordnet's structure makes it a useful tool for computational linguistics and natural language processing.

It superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, Wordnet interlinks not just word forms—strings of letters—but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, Wordnet labels the semantic relations among words, whereas the grouping of words in a thesaurus does not follow any explicit pattern other than meaning similarity.

4. DESIGN

The chat agent accepts to student's query and extracts the keywords (nouns and verbs from the question using a lexical parser. Then the keywords are compared with the category list database. With the category selected, the keywords are then compared with the questions under the category. The answer to the question is then obtained using most commonly occurring keyword. Figure 2 represent user and admin activities. Where user corresponds to a person who use the bot and admin add the related questions and corresponding answers to database

Usually all the answers are store directly in database but question are converted into list of keywords (words obtained after applying ontology based approach) are stored in database. Figure 3 gives ER diagram for relational database .whenever user ask any question that is taken as JSON object then converted in to list of keywords (after applying ontology based technique) map this with all list of keywords in database and calculate rank for each questions using TF-IDF finally retrieve the answer of that question which has maximum rank

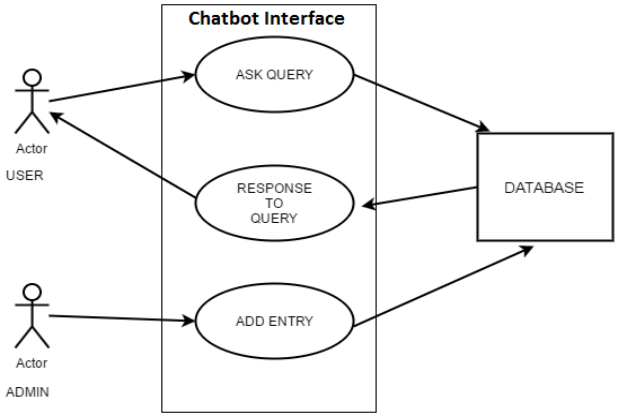


Fig -2: USE CASE representing user and admin activities

5. IMPLEMENTATION

The entire development process workflow is illustrated in the following steps:

5.1 JWNL Configuration:

A JWNL Properties file is an XML file that can be validated using the included DTD or XSD.basically the properties file allows to specify three properties:- Dictionary class: This defines the class used to interface with the dictionary. JWNL comes with three dictionary classes - MapBackedDictionary, FileBackedDictionary, and DatabaseBackedDictionary. exactly 1 dictionary tag is required in a properties file. If there is more than one, the first one will be used.

<dictionary class="[dictionary class name]">
..parameters
</dictionary>

- Version: Gives information on the version of Wordnet being interfaced with. Exactly 1 version tag is required in a properties file.

<version publisher="[publisher]" number="[version number]" language="[language]" country="[country]"/>

Where [language] and [country] are used to specify the locale whose language is covered by the dictionary. If these tags are not included, the default locale is assumed. See the Java documentation for java.util.Locale for information on locales.

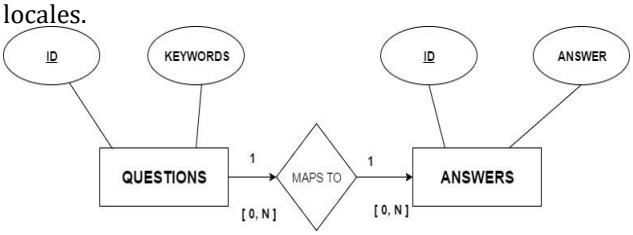


Fig -3: ER diagram for mapping answers to questions

5.2 Stemming and finding synonym:

Unfortunately, retrieving word information from Wordnet is not as fast user think so use a HashMap to store words stemmed, so stemming a word a second time will be no more expensive than a constant-time hashmap lookup. E.g. writing → write, ate →eat

Consider the prototype of the code

```
Stemmer () {
        /*
         * establishes connection to the Wordnet database
         */
JWNL.initialize (new FileInputStream
                                ("JWNLproperties.xml"));
        /*
         *JWNLproperties.xml is xml file which provide path
for Wordnet database
        */
dic = Dictionary.getInstance();
}
```

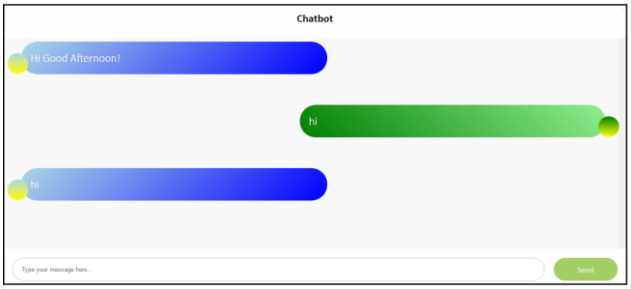
Code for close the connection to Wordnet

```
void Unload ()
        {
                dic.close();
                Dictionary.uninstall();
                JWNL.shutdown();
        }
```

Figure 4 and 5 represent the sequence of steps involved for stemming a given word using Wordnet

5.3 GUI for Chabot :

The right side text templates are user query/questions and text template that appear on rights side is the response from Chabot



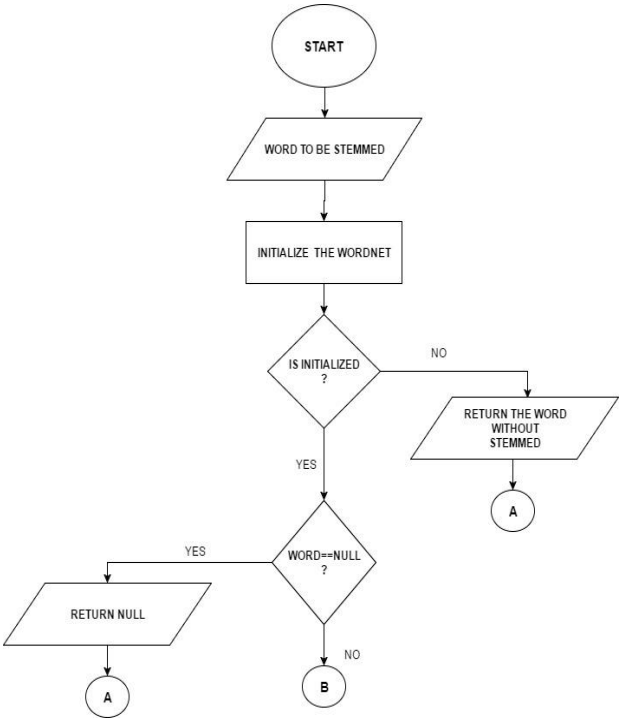


Fig -4: Flow diagram for stemming first part

5. CONCLUSIONS

This approach for the result (answer) of Chabot focuses on the case of a low computational and data budget. The results have shown that this kind of approach needs large quantity of data to be trained from scratch. In the scenario of having a small dataset (question and answer) and a task to perform, language processing can be used to find – related and appropriate answers that have been trained on large database and for other purposes.

Even though this field is underway of improvising the process a lot of work of optimization is required. As a future work some other techniques to obtain an accurate prediction of the answer (result) can be devised which takes less time for its computation

ACKNOWLEDGEMENT

We would like to thank our beloved principal Dr. G. L Shekar, NIE, Mysuru for support and encouragement. We are grateful to Dr. H.D Phaneendra, Head of the Department of Computer Science and Engineering, NIE, Mysuru for his encouragement facilitating the progress of this work. Our sincere acknowledgement to Smt. Anitha R, Associate professor, Computer Science and Engineering, NIE for her encouragement, support and guidance throughout the duration of the project.

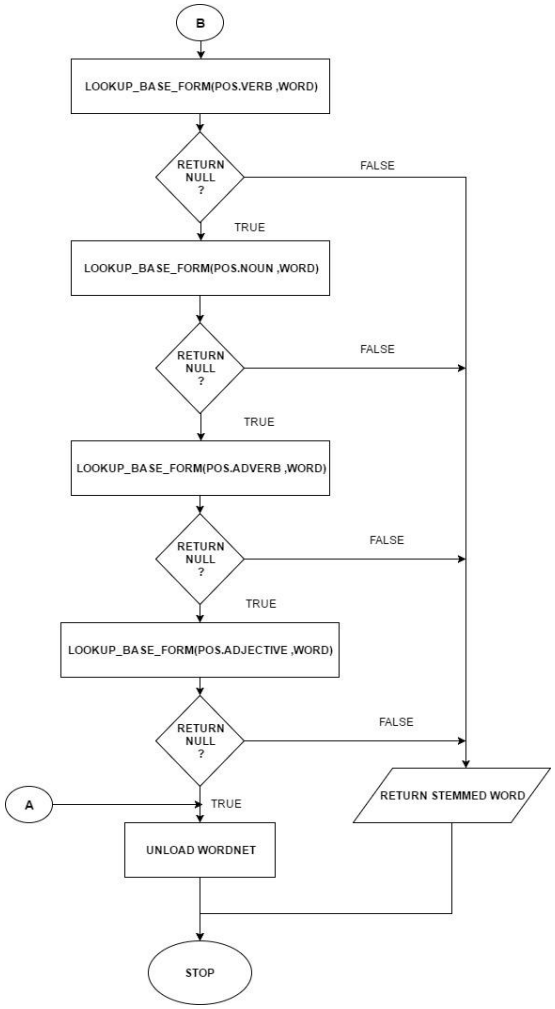


Fig- 5: Flow diagram for stemming second part

REFERENCES

[1]. Jessel Serrano, Fernando Gonzalez, Janusz Zalewski- 2015, " CleverNAO: The Intelligent Conversational Humanoid Robot ".

[2]. Comparative studies of AIML by yungang wei, Bo sun, Boying sun and Xiaoming zhu - 2016

[3]. Hadeel Al-Zubaide, and Ayman A. Issa, "Ontology based Chabot ".

[4]. Vincenzo Di Lecce*, Marco Calabrese,DomenicoSoldo and Alessandro Quarto , "Dialogue-Oriented Interface for Linguistic Human Computer Interaction".

[5]. Abran, A., Cuadrado, J., Garc´ıa-Barriocanal, E., Mendes, O., S´ anchez-Alonso, S., and Sicilia, M., (2006) , "Ontologies for Software Engineering and Software Technology". Berlin Heidelberg: Springer-Verlag.

[6]. Wordnet, (2010). Wordnet Project [online]. WN Team. Available from: http://wordnet.princeton.edu/ [Accessed 05/21/2011].

[7]. SupratipGhose and JagatJoytiBarua, "Topic specific Dialogue based Natural Language".