HiALi!

An artificial intelligent chat bot

By

Abdullah Frahmand

Table of contents

Introduction

Hi Ali is the project I had in mind for a long time, although numerous projects of same goal can be found throughout the history and evolution of computer sciences, but I am determined to apply the idea I have which without further ado I will explain:

Hi Ali is intelligent software that has the ability to listen to human input sentences, break them down into words and understand the structure, tense, meaning and the goal of the sentence and responds using its specially structured stored data and the implemented logic inside the scripted code.

The use case very wide and to mention a few:

1. Interface for the functionality of a large complex system e.g. power plants, factories etc. where this robot is responsible at all time that everything is going well.
2. Personal life assistant, where the robot is responsible of all the schedules, spending, special occasions, trips, location, well-being somewhat more humanoid version of google account etc.

At this point the project's goal is to build a prototype of the software, with basic functionalities implemented to be tested in practice. That is all for Ali to know about in short, upcoming is the documentation of the HiALi prototype implementation.

Architecture

Ali is made of three major parts: UI, Controller and Back End. Since web technology has been used to implement the project, the three parts are using the following web technologies.

UI: UI as the client side of software uses HTML and CSS for layout and graphics and JavaScript and JQuery for the interface elements functionalities e.g. message input and output mechanics.

Controller: Server side script of the software has been hosted in Apache server and PHP have been used as server side script. 50% of the artificial intelligence is implemented inside the server script and is the place where sentence break down and words are sorted, grammar is implemented while at this stage the script has nothing to do with the meaning of the words.

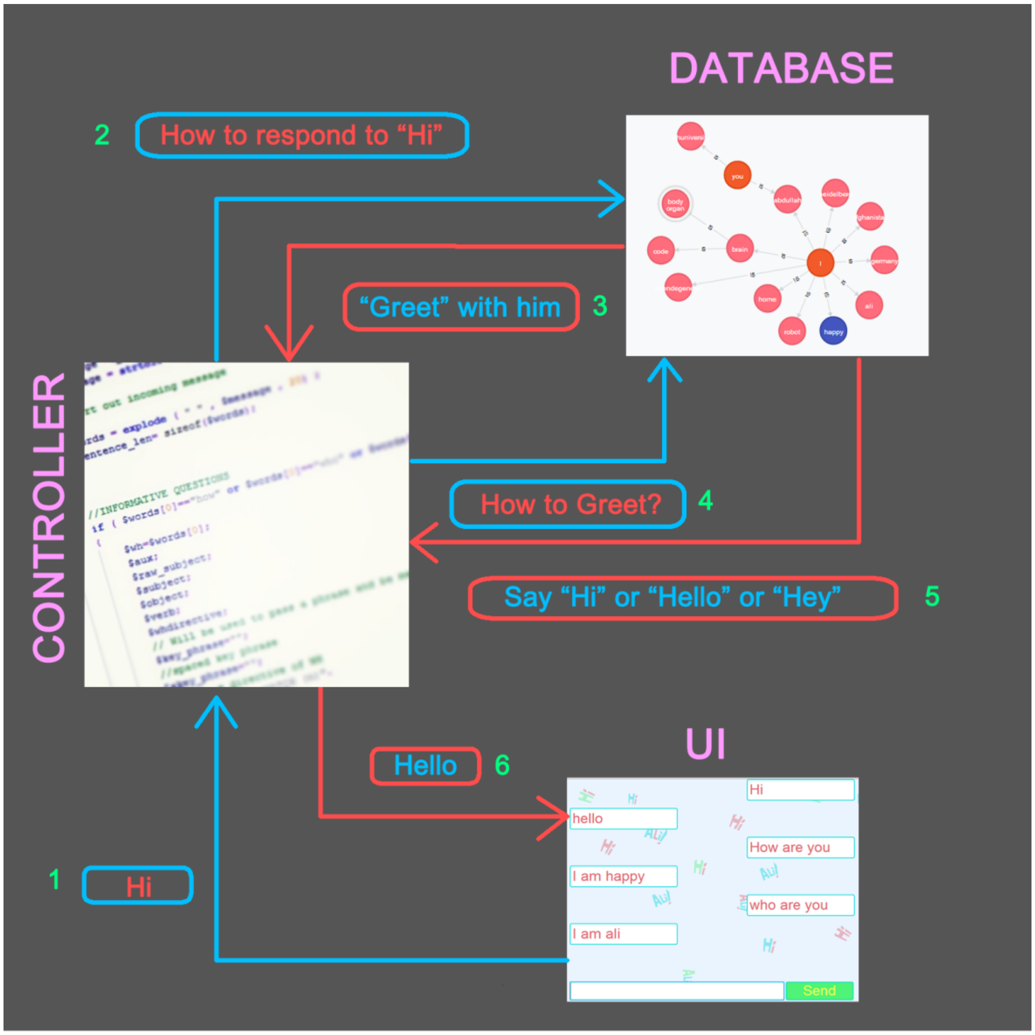
Back End: Neo4J is a graph database that has been used to implement the data storage for Ali in a way that makes the other 50% of the artificial intelligence. Since Neo4j is a white board friendly and it is very easy to imagine real world in it I have chosen it. The database is full of nodes that are people, material, emotions, names that are connected to each other with relationships for example:

(Node)-[Relationship]->(Node)

(Robot)<-[is a]-(Ali)-[loves]->(Ellen DeGeneres)-[is a]->(Show Host)

Using this mechanic I have implemented a data graph for Ali that shapes all of his understanding of the world.

The Following flow chart shows the mechanism all three parts work together:



```
ge = strt
rt out incoming message
rds = explode ( " " , $message . 
entence_len= sizeof($words);

//INFORMATIVE QUESTIONS
if ( $words[0]=="how" or $words[0]=="who" or $words
{
    $wh=$words[0];
    $aux;
    $raw_subject;
    $subject;
    $object;
    $verb;
    $whdirective;
    // Will be used to pass a phrase and be
    $key_phrase="";
    //spaced key phrase
    $spkey_phrase="";
```

Hi

hello

How are you

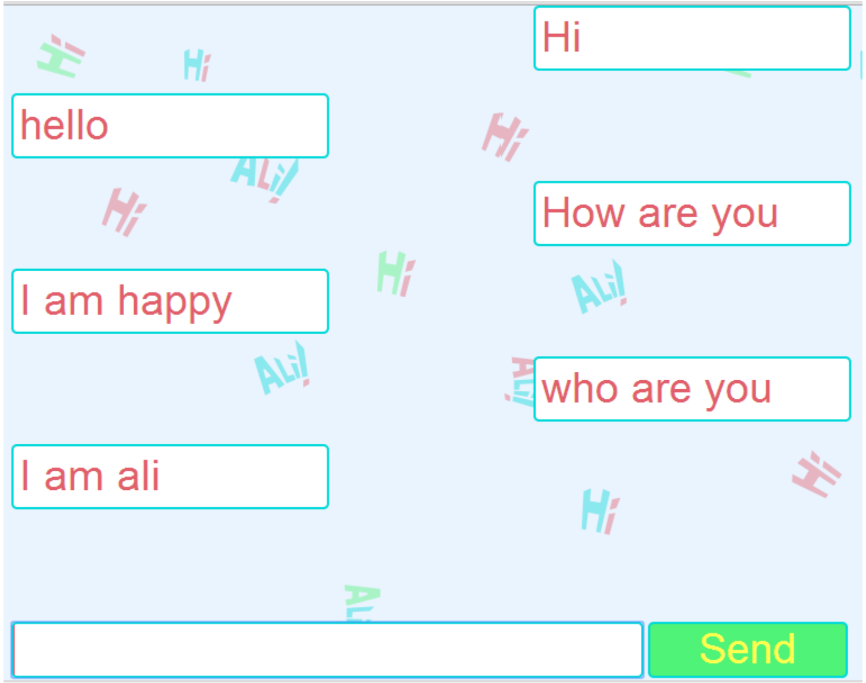
I am happy

who are you

I am ali

Implementation

UI:


Hi

hello

How are you

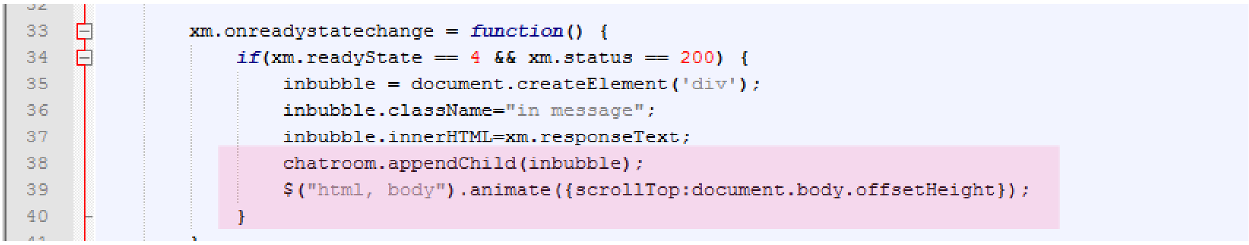
I am happy

who are you

I am ali

UI is very simple a single page with two controls: text box and the send button. The page is loaded the focus of input devices are brought on the text-box so you can directly start typing then you can send the message either by pressing enter or clicking send button.

Messages are appended in bubble chat style using JavaScript runtime HTML element building. Once the page is full the window is extended and scrolled down automatically and appearance of each bubble so the most recent messages are visible.


```
        xm.onreadystatechange = function() {
            if(xm.readyState == 4 && xm.status == 200) {
                inbubble = document.createElement('div');
                inbubble.className="in message";
                inbubble.innerHTML=xm.responseText;
                chatroom.appendChild(inbubble);
                $("html, body").animate({scrollTop:document.body.offsetHeight});
            }
```

Controller:

Controller is basically responsible to break down the sentence, identify to-be verbs, auxiliaries, subjects, objects in the meantime ask data-base for direction and send out queries and structure the response message. It is made of three major parts: Informative Questions, Yes/No Questions and Statements.

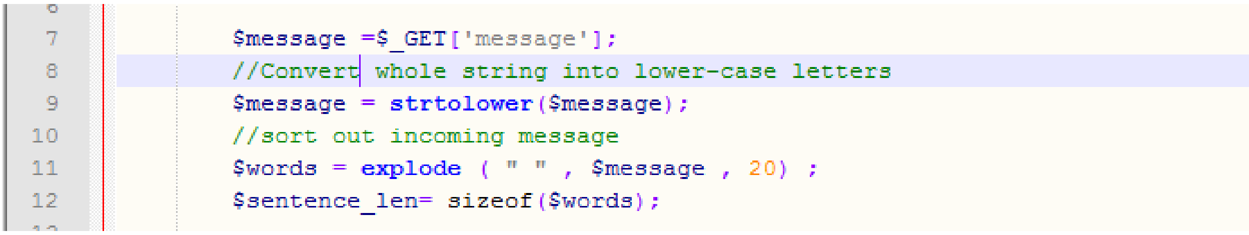
Note: The code has been commented very in detail.

Informative Questions looks for a WH word in the beginning of the sentence including “how”, and cycles through the words of sentence looking for auxiliary. The word after auxiliary is considered as Subject and the rest of sentence as a key-phrase.

Yes/No Questions is rather simpler that the WH section. It detects wether the first word is an auxiliary and then admits the following word as subject and the last word as object and uses the rest of sentence a key-phrase and looks up in the database whether there is a relationship from subject to object considering the key-phrase.

Statement part is where the user enters information which Ali might store in its brain. For example he will re-learn user’s name and over write and refers to that name as “you”.

PHP have been used for the server script because it is a very powerful tool for general software development with variety of functions especially with string.



```php
$message =$_GET['message'];
//Convert whole string into lower-case letters
$message = strtolower($message);
//sort out incoming message
$words = explode ( " " , $message , 20) ;
$sentence_len= sizeof($words);
```

For the communication between the server and client XMLHttp Reuqest object have been used which is available in JavaScript that provides background data transfer between server and client also known as AJAX (Asynchronous JavaScript and XML).

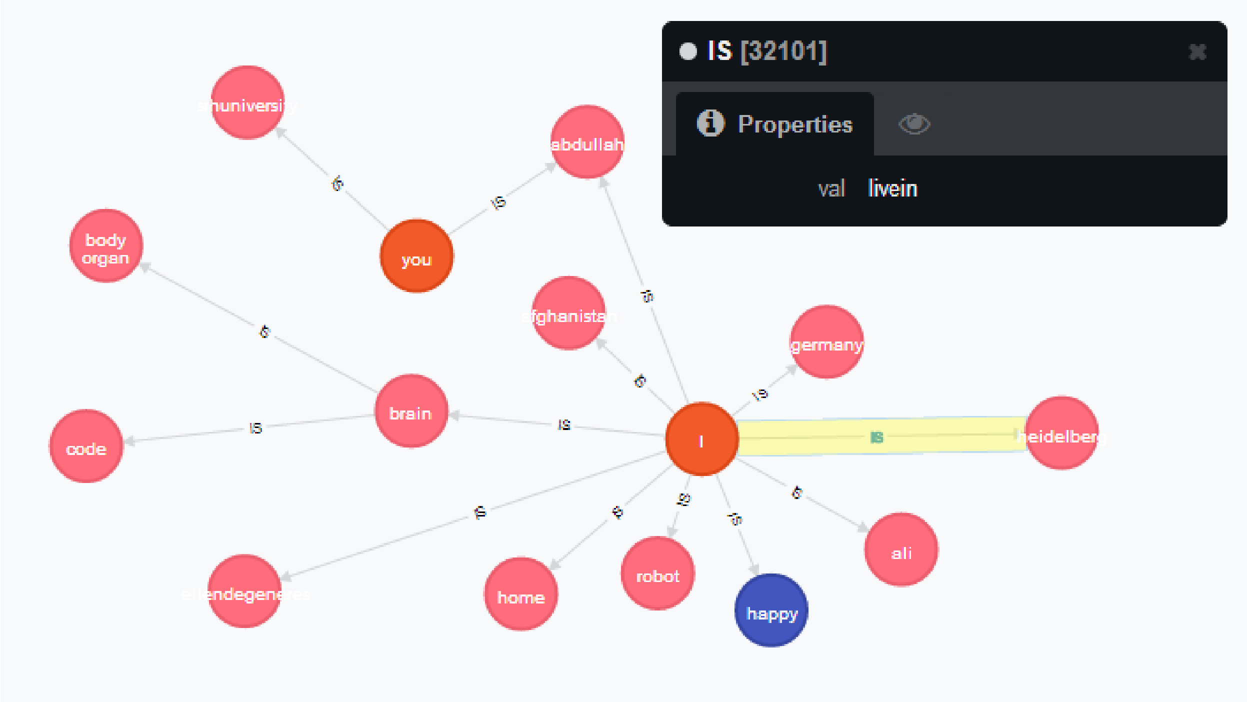
Back End:

Neo4j is a graph NOSQL database that enables to store data in nodes and connect them to each other with relationships. Both nodes and relationships can hold data inside properties which is very convenient for storing words and also directives and keywords for them to mimic meaning in a way.

In Nodes I have implemented People and Phenomenon as SUBJECT while the have properties as “pname” that stands for person and “tname” that stands for any other Phenomena and by Phenomena it is meant any visible and invisible known existence for example “Gravity”. There are nodes that group WH words or GREET or RECATION. Each node has a directive and a keyword, directive will enable script to know which word is being referred to and keyword is the way to find pointed word. The script can find its way from one word to another via relationships.

Relationships store mostly verbs mixed with prepositions and conjunctions.

Note: Neo4jPHP REST API developed by Josh Adell have been used as PHP dependency library to connect with Neo4j server.



● IS [32101]

val livein

The Exhibit shows the relationship between I(Ali) and Heidelberg which stores “livein” and means “I live in Heidelberg” to the script.

Future Objectives

My future objectives and vision for Hi Ali is as following

1. Deeper understanding of grammar and reducing the need of key-phrase
2. Wide re-factor of the code, encapsulation of functions, use object oriented approach.
3. Use of session variables to keep track of a conversation rather than request and response.
4. Refining data storage mechanism, keywords, hidden masks, space saving technics and etc.
5. In a bigger picture developing Ali into more that keyboard-screen application into other variety of input and output systems.
6. Fixing use-case for Ali

Conclusion

The module have been beneficial for me by giving me a chance to explore my abilities in term of coming up with an idea and building on it. I came across some standards and good practices and tools for Software Architecture which I have not been familiar before. Developing the prototype I have promoted my skills and learned a lot using new technologies like Neo4j.