IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. VI (Jan – Feb. 2015), PP 05-18
www.iosrjournals.org

Implementation of Emotional Intelligence in a machine

Indrajit Sinha[1], Dr. Kanhaiya Lal[2]

[1](*Computer Science and Engineering, Birla Institute of Technology Patna Campus/ Birla Institute of Technology Mesra, India)*

[2](*Computer Science and Engineering, Birla Institute of Technology Patna Campus/ Birla Institute of Technology Mesra, India)*

Abstract: *This paper is an extended work of our paper "A New Concept on Thinking Machines: Cyber Personality" published in IJCS journal. Our previous work dealt with a new concept on implementing the entire personality in a machine. In this paper we have provided an implementation of introducing emotional intelligence inside a computer by providing a pseudo-code along with a partial implementation of the same. Suitable results have been included to prove our work.*

Keywords: *Artificial Intelligence, Cyber Personality, Emotional Intelligence, Psychology and Turing Machine.*

I. Introduction

This paper involves implementing emotional intelligence inside a computer based on a particular individual. It is an extended work of our previous work Indrajit Sinha, Dr. Kanhaiya Lal, "A New Concept on Thinking Machines: Cyber Personality", In IJCS, pp: 25-29, Vol 2, issue 1, January, 2015. Our previous work involved implementing the complete personality of an individual in a computer. In this paper we have provided a method to implement emotional intelligence inside a computer. However it is well known that emotions of all humans are unique and different from each other. Hence our implementation also relates to using the emotional approach of a particular human. In other words the computer is to analyze emotions based on the interpretation capabilities of a particular person. To make our work more clear we shall provide a brief introduction of certain topics that are related to our field.

A chatbot is a program designed to simulate a conversion with one or more humans. Its main aim is to fool the human into believing that it is communicating with another human.[1-2] It uses the concept of the Turing machine which is a machine intelligent to fool anyone. Although many chatbots and expert systems have been designed to get close to it none are exact. Cleverbot is a bit more sophisticated than chatbots in the fact that they have a learning system from human inputs.[3]

Some such systems include Elbot created by Fred Roberts[4], Jabberwacky created by Rollo Carpenter[5], ELIZA[6] and A.L.I.C.E. which stands for Artificial Linguistic Internet Computer Entity.[7]

However certain chatbots now involve application of linguistics which is the scientific study of languages.[8] This involves minimalist program which deals with development under generative grammar.[9] Phrase structure is used to define syntax of a language which includes principles and processes used for constructing a sentence for a particular language.[10-11] However all of this is integrated using sentence diagram which is pictorial representation of the structure of a sentence.[12] The theories of linguistics can be implemented mathematically using predicate logic and inference rules.[13-18]

What lacks in these chatbots is the capability to understand the emotions of the human with which the chat is going on. Often a word is stated in different ways depending on the current mood of the person. Using emotional intelligence requires understanding the working of our brain. Our brain contains a neocortex which is a bulb of convoluted tissues forming the top layers. This is the thought region of our brain. The limbic system surrounds the brainstem and looks like a bagel with a bite taken out at the bottom where the brainstem nestles into them. This system generates emotional feelings. The main specialist in emotions is however the amygdala which is an almond-shaped cluster of interconnected structures perched above the brainstem, near the bottom of the limbic ring. There are actually two amygdalas that nestle towards the sides of the brain.[19]

Humans have great creative thinking but cannot deal with huge amounts of data and that's where statistical and empirical methods come in for aid.[22]

As we know that Artificial Intelligence tries to understand and build intelligent systems,[20-21] it is suitable to say that our work falls under this category. Personality is an individual's unique traits and the study of emotions has revealed the physiological and psychological aspects of emotions.[23-24] Beliefs help to create attitude and hence enhance the personality of a person.[25]In order to make artificial agents capable of understanding human emotions we have to endow such agents with a suitable model of our emotions.[26]Hence it is of vital importance after so much development that emotions be implemented in a computer that will make it more advanced technologically.

II. Previous Work

In our previous work we provided a model for cyber personality. We will provide the diagram of the model here once more for better understanding of our pseudo-code that we have made in this paper.

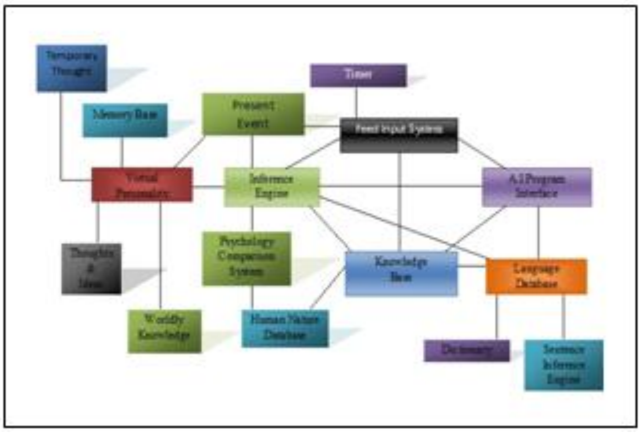


Figure 1. Design Model of the concept

Each of the blocks are separate modules used in our pseudo-code. The functions of each module have been defined in the pseudo-code. The algorithm has already been provided in the previous work.[27]

III. Pseudo-code

```
string s=i/p                                [input statement]
feedback(s)                                 [function or module]
//In feedback(s)
t=0   //initialize
t=t+1
p_event(s);                                 [function or module]
//In p_event(s),
str[]=s;                                    [store original string]
knowledge_base(s)                           [function or module]
//In knowledge_base(s)
k=lang_db(s)                                [function or module]
//In lang_db(s)l
L=s.length
create char array A[l]
for i=0 to L
if A[i].equals ' '
c=c+1                                       [c is initialized to 0 at the
                                            beginning]
L2=i                                        [L2 is different from l]
if(c==1)                                    [nested if loop]
for j=0 to L2
B[j]=A[j]                                   [end of for loop]
string s1=B[]
dict(s1)                                    [ function or module and
                                            end of nested if loop]
else                                        [nested else loop]
for j=m to L2
B[j]=A[j]
string s1=B[]
dict(s1)                                    [end of nested else loop]
m=L2                                        [end of outer if loop]
// In dict(s1)
/*send the word to a dictionary software and extract different meanings of it and store them in an array*/
D[a]={array of meanings of the word}
return D[a]
```

```
28.    //In lang_db(s)
 //store word and meanings in a 2-D array
E[i][j]={word with various meanings}
29.    for i=0 to c
30.    for j=0 to a                       [a=number of meanings for each
                                              word]
31.    E[i][j]=D[a]
32.    X_Bar(E[][]) and infer(B[],E[][])     [functions or modules]

33.    //In X_Bar(*x,*y)
       Let us take an input statement: -
          Hi, how are you?
       First using X-BAR THEORY: -
```

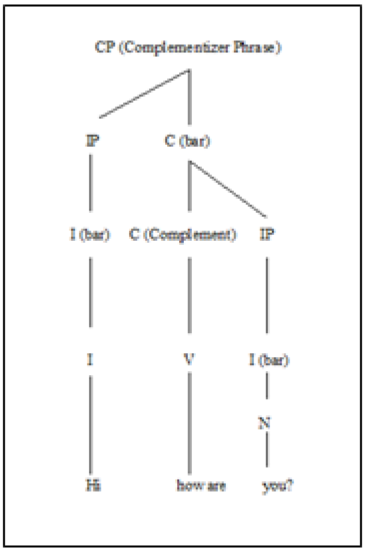


Figure 2. Input sample using X-BAR Theory

After forming the X-BAR Tree, the meaning of constraints is carried out using E[i][j] to realize that the theme is “how” and it’s a question due to the presence of ’?’ at the end.

```
34.    Mark F[i][j]                      [position of each effective meaning as
                                              1 rest 0]
35.    We take P as “Hi”, Q as “how are” and R as “you”.
36.    Now using Equivalence Laws: -
       PV(Q&R) ->  (PVQ)&(PVR)         (Distributivity Law)
       Let (PVQ) = S and (PVR) = D
       Therefore,
       PV(Q&R)->S&D
       Using Inference Rules: -
       PV(Q&R)->S          (Since, P&Q->P ,Simplification Law)
       Thus Resultant is => S
       Which means => PVQ?
       This means resultant has “Hi” and “how are”.
37.    for i=0 to c
38.    for j=0 to a
39.    if(F[i][j]==1)                      [F[][] is a 2D array containing
                                                 flag value 1 being correct
                                                 meaning applied here for
                                                 each word]
40.    A1[i][j]=E[i][j]                     [end of if and both for loops]
41.    return A1[][]
42.    //In lang_db(s)
   knowledge_base(A1[][])               [function call]
43.    //In knowledge_base(A1[][])
```

```
  for i=0 to c
44.                 for j=0 to a
45.                 if(A1[i][j]==H[i][j])                     [compare with human nature
                                                                      database]
46.                 g=i;
47.                 h=j;
48.                 hit=1;                                    [hit initialized to 0]
49.                 break;                                    [end of if loop]
50.                 if(hit==1) go to step 164
51.                 else
52.                 Boolean x=false
53.                 for i=0 to c
54.                 Y[i]=0                                    [end of for loop]
55.                 infer(Y[],A1[][])                         [function call]
56.                  feedback(x)                              [function call
                                                   and end of else loop]
57.                 //In infer(X[],Z[][])
   if(X[]==1)
58.                 a1=1
59.                 psych(a1,Z[][])                           [end of if loop]
60.                 else
61.                 a1=0
62.                 psych(a1,Z[][])                           [end of else loop]
63.                 //In psych(a1,Z1[][])
   if(a1==1)

64.                 Z1[g][h]=H[g][h]                          [here fixed values of g and h are
                                                                      used]
65.                 return Z1[g][h]                           [end of if loop]
66.                 else
67.                 i=j=0;
68.                 count=0;                                  [initialization]
69.                 while(H[i][j]!=NULL)
70.                 i++;
71.                 j++;
72.                 count++;                                  [end of while loop]
73.                 H[i+1][j+1]=Z1[g][h]
74.                 count++;                                  [end of else loop]
75.                 return Z1[g][h];
76.                 //In infer(Y[],A1[][])
  persona(Z1[g][h])                          [function or module]
77.                 //In persona(Z1[g][h])
   memo(Z1[g][h])                            [function or module]
78.                 think(Z1[g][h])                          [another function or module]
79.                 wisdom(Z1[g][h])                         [a total of three functions or
                                                     modules are called]
80.                 //In memo(Z1[g][h])
  i=j=0;
81.                 while(M[i][j]!=NULL)
82.                 if(Z1[g][h]==M[i][j])
83.                 hit=1;
84.                 g=i;
85.                 h=j;                                     [end of if loop]
86.                 i++;
87.                 j++;                                     [end of while]
88.                 return M[g][h];
89.                 //In think(Z1[g][h])
  i=j=0;
90.                 while(T[i][j]!=NULL)
```

```
if(Z1[g][h]==T[i][j])
hit=1;
g=i;
h=j;                                        [end of if loop]
i++;
j++;                                        [end of while loop]
return T[g][h];
//In wisdom(Z1[g][h])
i=j=0;
while(W[i][j]!=NULL)
if(Z1[g][h]==W[i][j])
hit=1;
g=i;
h=j;                                        [end of if loop]
i++;
j++;                                        [end of while loop]
return W[g][h];
/*In persona(Z2[g][h]) where returned array values from memo(Z1[g][h]), think(Z1[g][h])
and wisdom(Z1[g][h]) are received as Z2[g][h] because any one of the three will take a hit.*/
for i=0 to count
for j=0 to count
if(Z2[g][h]==P[i][j])
hit=1;
break;                                      [end of if loop]
else
hit=0;                                      [end of else and both for loops]
if(hit==1)
i1=i;
j1=j;
return P[i1][j1];                           [end of if  loop]
else
P[i+1][j+1]=Z2[g][h]
return P[i+1][j+1]                          [end of else loop]
/*In infer(Y[],A1[][]), Z1[g][h] is received from psych(Z1[g][h]) and P[i1][j1] is received
from persona(Z1[g][h]). */
for i=0 to n                          [predefined total length of fuzzy
                                       table 1 content in infer]
for j=0 to n
if(Z1[g][h]==F1[i][j])
k1=i;
k2=j;
break;                                      [end of  inner if and outer for
                                                          loops]
for i=0 to m                                [m is predefined total length of
                                      fuzzy table 2 content in infer]
for j=0 to m
if(P[i1][j1]==F2[i][j])
k3=i;
k4=j;
break;                                      [end of if and outer for loops]
for i=0 to n1                               [n1 is length of 2D elements of
                                      fuzzy table 3 content in infer]
for j=0 to n1
for i1=0 to n1
for j1=0 to n1
if(Big[k1][k2][k3][k4]==Fuz[i][j][i1][j1])
p1=i;
p2=j;
p3=i1;
```

```
141.    p4=j1;
142.    /*Take corresponding values of column 1 and column 2 of fuzzy table
Fuz[p1][p2][p3][p4]. */
  new1=col1                          [col1 is variable for column 1]
143.    new2=col2                    [col2 is variable for column 2]
144.    F3[new1][new2]               [end of inner if and all for loops]
145.    if(capture==1)
146.    trigger=1                    [trigger is flag variable initialized to 0]
147.    p_event(trigger)
148.    persona(F3[n1][n2])          [end of if]
149.    else
150.    trigger=0                    [end of else]
151.    //In persona(F3[n1][n2])
   W1[0][0]=F3[n1][n2]               [W1 has a predefined size of 1X1]
152.    //In p_event(trigger)
    persona(A1[][])
153.    trigger=0;
154.    //In persona(A1[][])
   W2[0][0]=A1[][]                   [W2 has a predefined size of 1X1]
155.    mood(W1[0][0],W2[0][0])      [function or module]
156.    //In mood(W1[0][0],W2[0][0])
  /*Using Fuzzy Table 3 values from both W1[0][0] and W2[0][0] take the corresponding P1(i) values and
check Fuzzy Table 4 for taking results of comparison of the two P1(i) values.*/
157.    lang_db(F3[new1][new2])      [function call]
158.    //In lang_db(F3[x][y])
  X_Bar(F3[x][y])
159.    Since R was "you", that means its conditional state was asked.
        Therefore R is changed to "I am"
        P becomes "Hi" and Q becomes "fine" (a sample reply).
160.    C[i]={array of words formed}
161.    dict(C[])                    [function call]
162.    //In dict(C[])
  D[a]={array of meanings of words}
163.    return D[a];
164.    //In X_Bar(F3[x][y])
  /*store words and meanings in a 2D array*/
  F4[x][y]={new words and their various meanings}
165.    for i=0 to c
166.    for j=0 to a
167.    F4[x][y]=D[a];               [end of both for loops]
168.    Now the expression PVQ becomes P&Q as both are to be sent compulsorily
   together.
        Thus we have -> P&Q
```

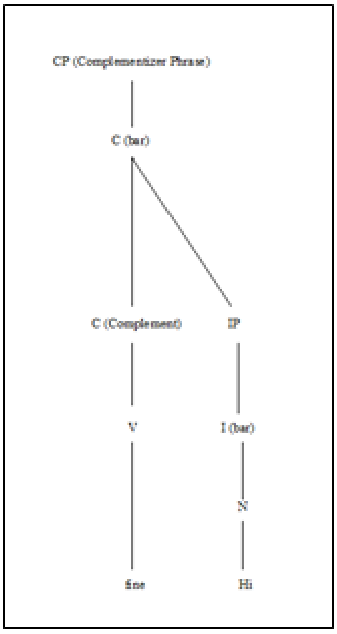


Figure 3. Output sample using X-BAR Theory

```
                        Here the “I am” or R is missing.
169.                    Thus using Inference Rules: -
                        P&Q
->(P&R)&Q (P becomes P&R Simplification rule used in reverse to bring in the missing link R)
170.                    Using Equivalence Laws: -
                        ->P&(R&Q)     (Associativity Law)
                        Now on replacing variables with constraints we get: -
                        Hi, I am fine.
(Note: ‘,’ is given after observing user’s format and “.” Is given in place of “?”)
                        Hence the user gets a suitable reply.
171.                    for i=0 to c
172.                    for j=0 to a
173.                    if(G[i][j]==1)
174.                    G1[i][j]=G[i][j]                        [end of if and both for loops]
175.                    infer(C[],G1[][])                       [function call]
176.                    //In infer(C[],G1[][])
  feedback(C[])                         [function call]
177.                    //In feedback(C[])
  string s2=C[]                         [convert array to string]
178.                    while(t>=t1)                            [t1 is a predefined fixed value]
179.                    send s2 as o/p
180.                    t++;                                    [end of while loop]
181.                    //In knowledge_base(A1[][])
  /*in case of hit in step 45. */
  boolean x=true;
182.                    feedback(x)                             [function call]
183.                    p_event(A1[][])                         [function call]
184.                    //In p_event(A1[][]),
    HMT[][]=A1[][]                      [store information]
185.                    infer(str[],A1[][])                     [function call]
186.                    //In infer(Y[],A1[][])
    Z[][]=A1[][];                       [send prestored nature]
187.                    capture=1;                              [capture is initialized to 0]
188.                    //In knowledge_base(A1[][])
     for i=0 to c
189.                    Y[i]=0;                                 [end of if loop]
190.                    infer(Y[],A1[][])                       [function call]
```

191. Repeat steps 57 to 180. /*except steps where new query or array is created as there will be a hit in each search case. */

IV. Fuzzy Tables

Fuzzy **Table 1**

Emotion(i/j)	Value Assigned
Excited	0
Tender	1
Scared	2
Angry	3
Sad	4
Happy	5

Fuzzy **Table 2**

Human Nature (i1/j1)	True	False	Reverse (False Case)
Extraverted	0	1	Introverted
Sensing	2	3	Intuitive
Thinking	4	5	Feeling
Judging	6	7	Perceiving

Fuzzy **Table 3**

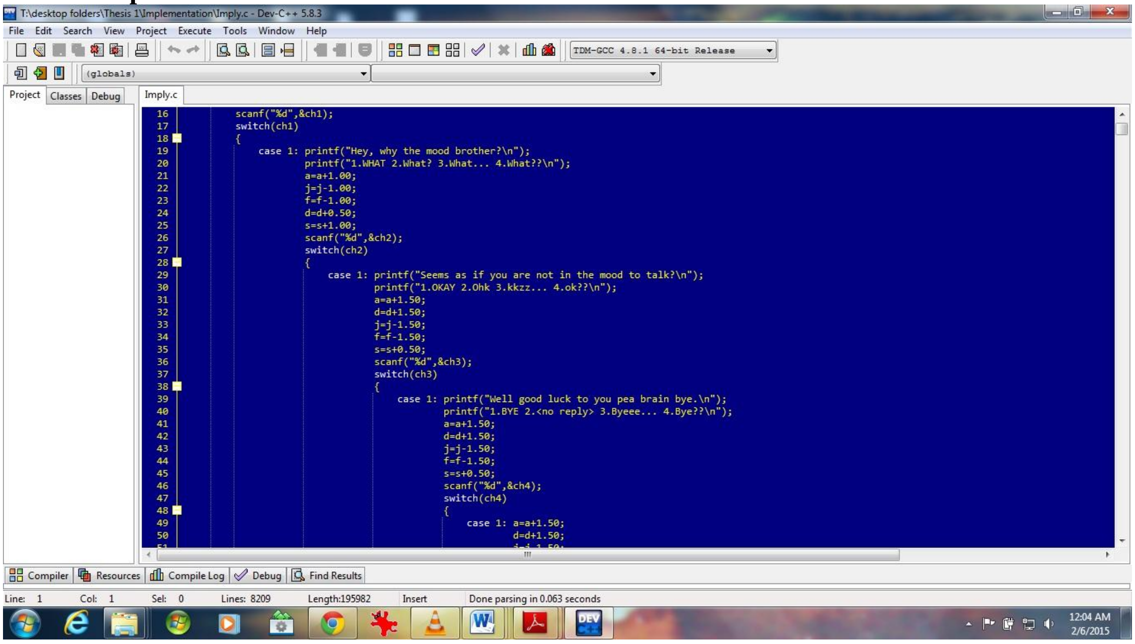
Serial No.	P1 (i)	P2 (j)	P3 (i1)	P4 (j1)	Col1	Col2
1	0	1	0	0	0	0
2	1	1	0	0	1	1
3	2	1	0	0	2	2
4	3	1	0	0	3	3
5	4	1	0	0	4	4
6	5	1	0	0	5	5
7	0	1	0	1	6	6
8	1	1	0	1	7	7
9	2	1	0	1	8	8
10	3	1	0	1	9	9
11	4	1	0	1	10	10
12	5	1	0	1	11	11
13	0	1	1	0	12	12
14	1	1	1	0	13	13
15	2	1	1	0	14	14
16	3	1	1	0	15	15
17	4	1	1	0	16	16
18	5	1	1	0	17	17
19	0	1	1	1	18	18
20	1	1	1	1	19	19
21	2	1	1	1	20	20
22	3	1	1	1	21	21
23	4	1	1	1	22	22
24	5	1	1	1	23	23
25	0	1	2	0	24	24
26	1	1	2	0	25	25
27	2	1	2	0	26	26
28	3	1	2	0	27	27
29	4	1	2	0	28	28
30	5	1	2	0	29	29
31	0	1	2	1	30	30
32	1	1	2	1	31	31
33	2	1	2	1	32	32
34	3	1	2	1	33	33
35	4	1	2	1	34	34
36	5	1	2	1	35	35
37	0	1	3	0	36	36
38	1	1	3	0	37	37
39	2	1	3	0	38	38
40	3	1	3	0	39	39
41	4	1	3	0	40	40
42	5	1	3	0	41	41
43	0	1	3	1	42	42
44	1	1	3	1	43	43
45	2	1	3	1	44	44
46	3	1	3	1	45	45
47	4	1	3	1	46	46
48	5	1	3	1	47	47

The Fuzzy Table 4 has no specific dimensions has number of categories in which different emotions can be categorized is still under research as further developments are being found.

V. Implementation

The partial implementation was done taking the emotional responses of a particular person. We then tested the implemented responses with one of his close friends who knew him well to take results for comparison. The snapshots of our code with input and output are given below:-

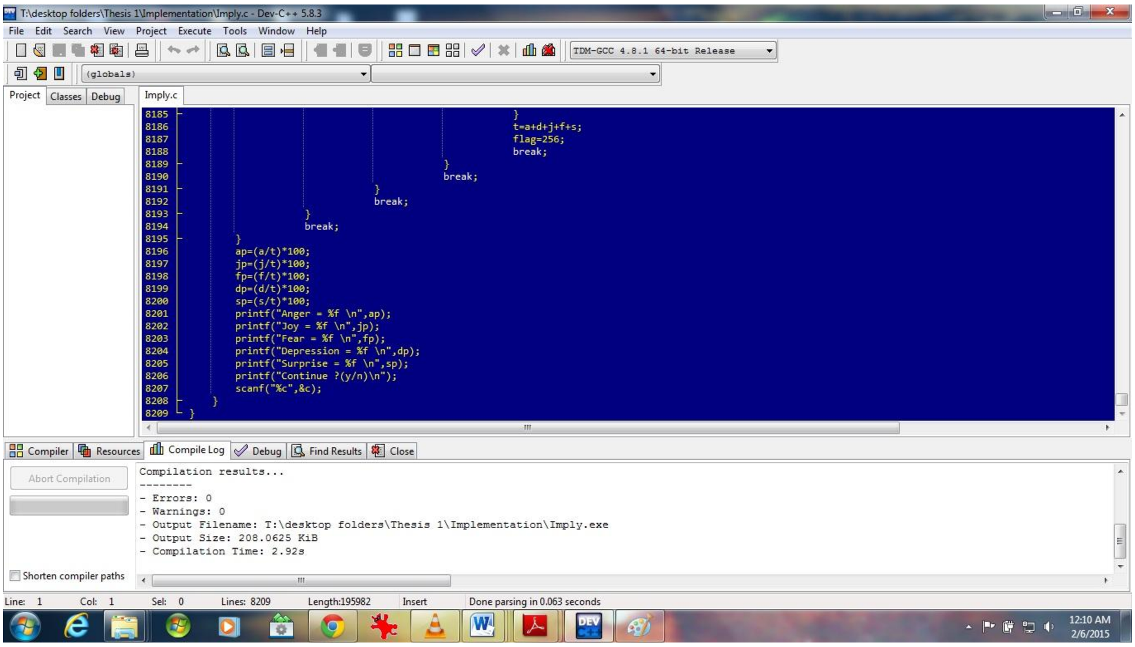
1. Code Samples :-



T:\desktop folders\Thesis 1\Implementation\Imply.c - Dev-C++ 5.8.3
File Edit Search View Project Execute Tools Window Help
TDM-GCC 4.8.1 64-bit Release
(globals)
Project Classes Debug
Imply.c
Compiler Resources Compile Log Debug Find Results
Line: 1 Col: 1 Sel: 0 Lines: 8209 Length:195982 Insert Done parsing in 0.063 seconds
12:04 AM
2/6/2015

```
        scanf("%d",&ch1);
        switch(ch1)
        {
            case 1: printf("Hey, why the mood brother?\n");
                    printf("1.WHAT 2.What? 3.What... 4.What??\n");
                    a=a+1.00;
                    j=j-1.00;
                    f=f-1.00;
                    d=d+0.50;
                    s=s+1.00;
                    scanf("%d",&ch2);
                    switch(ch2)
                    {
                        case 1: printf("Seems as if you are not in the mood to talk?\n");
                                printf("1.OKAY 2.Ohk 3.kkzz... 4.ok??\n");
                                a=a+1.50;
                                d=d+1.50;
                                j=j-1.50;
                                f=f-1.50;
                                s=s+0.50;
                                scanf("%d",&ch3);
                                switch(ch3)
                                {
                                    case 1: printf("Well good luck to you pea brain bye.\n");
                                            printf("1.BYE 2.<no reply> 3.Byeee... 4.Bye??\n");
                                            a=a+1.50;
                                            d=d+1.50;
                                            j=j-1.50;
                                            f=f-1.50;
                                            s=s+0.50;
                                            scanf("%d",&ch4);
                                            switch(ch4)
                                            {
                                                case 1: a=a+1.50;
                                                        d=d+1.50;
```

Figure 4. Sample Of Code



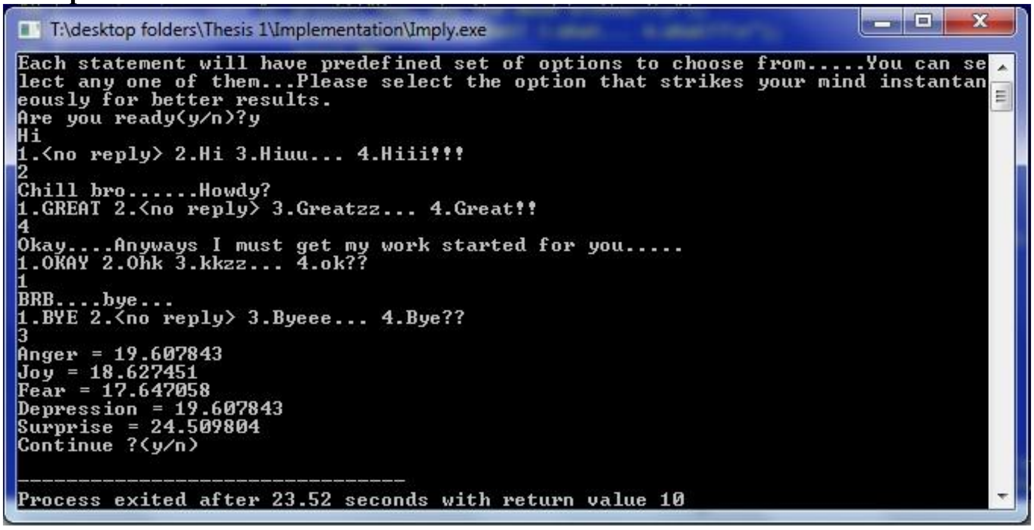
T:\desktop folders\Thesis 1\Implementation\Imply.c - Dev-C++ 5.8.3
File Edit Search View Project Execute Tools Window Help
TDM-GCC 4.8.1 64-bit Release
(globals)
Project Classes Debug
Imply.c
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths
Line: 1 Col: 1 Sel: 0 Lines: 8209 Length:195982 Insert Done parsing in 0.063 seconds
12:10 AM
2/6/2015

```
                                                        }
                                                        t=a+d+j+f+s;
                                                        flag=256;
                                                        break;
                                            }
                                            break;
                                }
                                break;
                    }
                    break;
        }
        ap=(a/t)*100;
        jp=(j/t)*100;
        fp=(f/t)*100;
        dp=(d/t)*100;
        sp=(s/t)*100;
        printf("Anger = %f \n",ap);
        printf("Joy = %f \n",jp);
        printf("Fear = %f \n",fp);
        printf("Depression = %f \n",dp);
        printf("Surprise = %f \n",sp);
        printf("Continue ?(y/n)\n");
        scanf("%c",&c);
    }
}
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: T:\desktop folders\Thesis 1\Implementation\Imply.exe
- Output Size: 208.0625 KiB
- Compilation Time: 2.92s
```

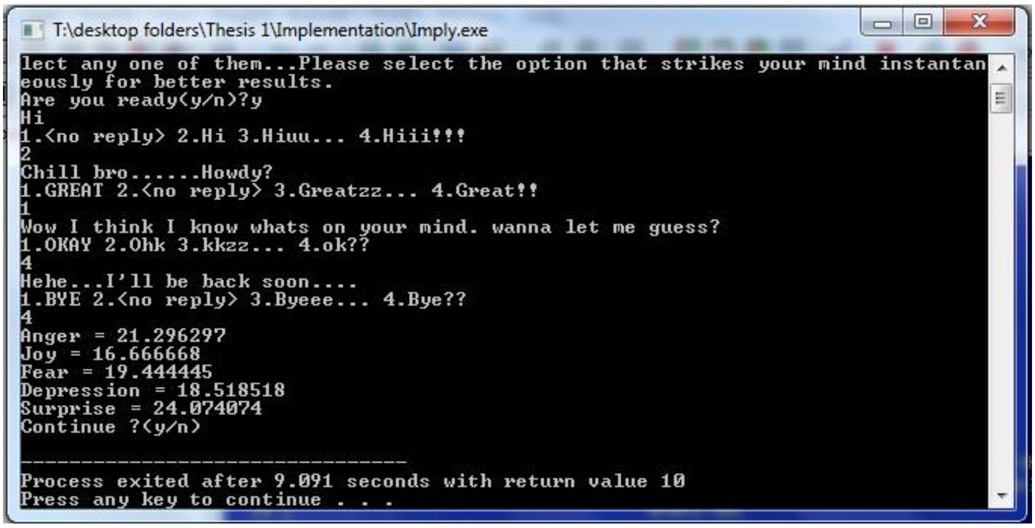
Figure 5. Another Sample Of Code

2. Test Samples :-



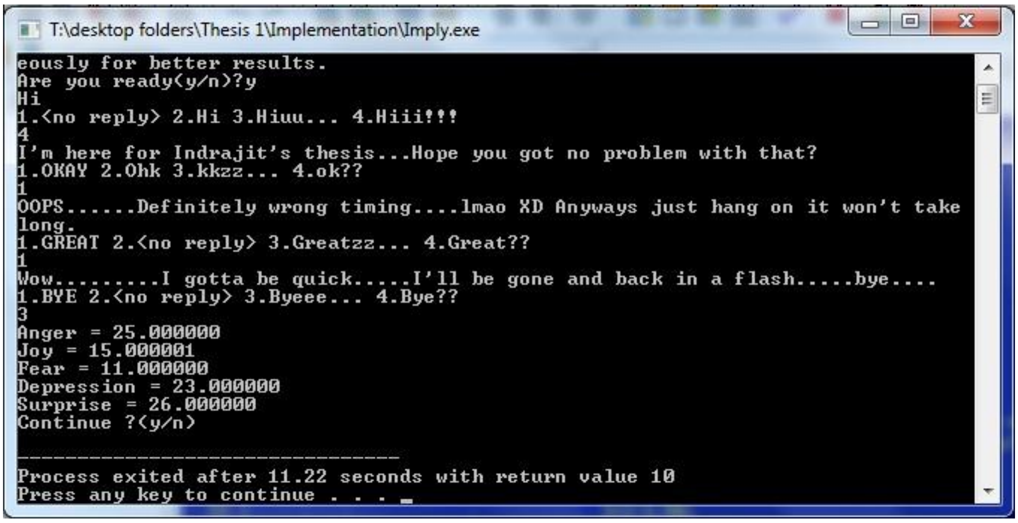
```
Each statement will have predefined set of options to choose from.....You can se
lect any one of them...Please select the option that strikes your mind instantan
eously for better results.
Are you ready(y/n)?y
Hi
1.<no reply> 2.Hi 3.Hiuu... 4.Hiii!!!
2
Chill bro......Howdy?
1.GREAT 2.<no reply> 3.Greatzz... 4.Great!!
4
Okay....Anyways I must get my work started for you.....
1.OKAY 2.Ohk 3.kkzz... 4.ok??
1
BRB....bye...
1.BYE 2.<no reply> 3.Byeee... 4.Bye??
3
Anger = 19.607843
Joy = 18.627451
Fear = 17.647058
Depression = 19.607843
Surprise = 24.509804
Continue ?(y/n)

--------------------------------
Process exited after 23.52 seconds with return value 10
```

Figure 6. Test 1



```
lect any one of them...Please select the option that strikes your mind instantan
eously for better results.
Are you ready(y/n)?y
Hi
1.<no reply> 2.Hi 3.Hiuu... 4.Hiii!!!
2
Chill bro......Howdy?
1.GREAT 2.<no reply> 3.Greatzz... 4.Great!!
1
Wow I think I know whats on your mind. wanna let me guess?
1.OKAY 2.Ohk 3.kkzz... 4.ok??
4
Hehe...I'll be back soon....
1.BYE 2.<no reply> 3.Byeee... 4.Bye??
4
Anger = 21.296297
Joy = 16.666668
Fear = 19.444445
Depression = 18.518518
Surprise = 24.074074
Continue ?(y/n)

--------------------------------
Process exited after 9.091 seconds with return value 10
Press any key to continue . . .
```

Figure 7. Test 2



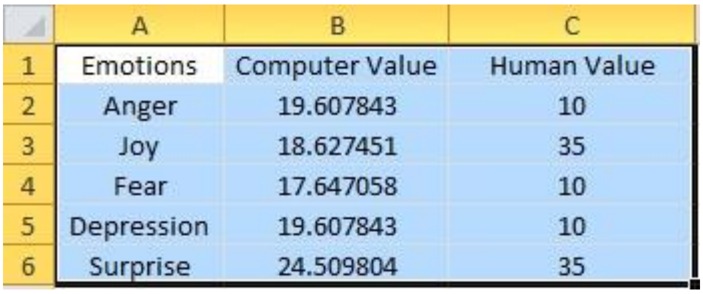
```
eously for better results.
Are you ready(y/n)?y
Hi
1.<no reply> 2.Hi 3.Hiuu... 4.Hiii!!!
4
I'm here for Indrajit's thesis...Hope you got no problem with that?
1.OKAY 2.Ohk 3.kkzz... 4.ok??
1
OOPS......Definitely wrong timing....lmao XD Anyways just hang on it won't take
long.
1.GREAT 2.<no reply> 3.Greatzz... 4.Great??
1
Wow.........I gotta be quick.....I'll be gone and back in a flash.....bye....
1.BYE 2.<no reply> 3.Byeee... 4.Bye??
3
Anger = 25.000000
Joy = 15.000001
Fear = 11.000000
Depression = 23.000000
Surprise = 26.000000
Continue ?(y/n)

--------------------------------
Process exited after 11.22 seconds with return value 10
Press any key to continue . . . _
```

Figure 8. Test 3

VI. Results

We have taken the data and compared it with actual results of that particular person.

1. Test 1:-



	A	B	C
1	Emotions	Computer Value	Human Value
2	Anger	19.607843	10
3	Joy	18.627451	35
4	Fear	17.647058	10
5	Depression	19.607843	10
6	Surprise	24.509804	35

Figure 9. Data Set 1

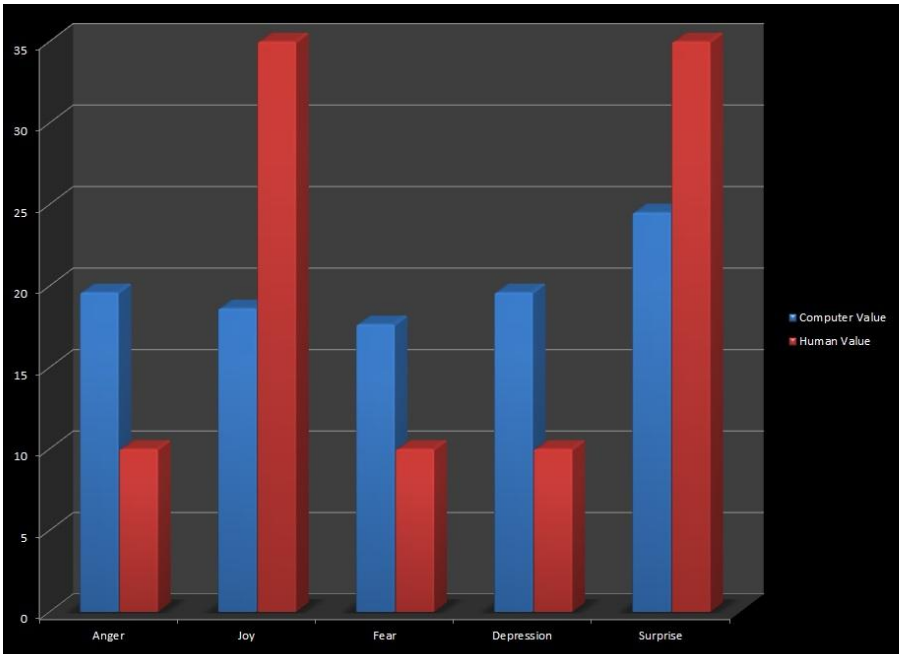


Figure 10. Column Chart 1

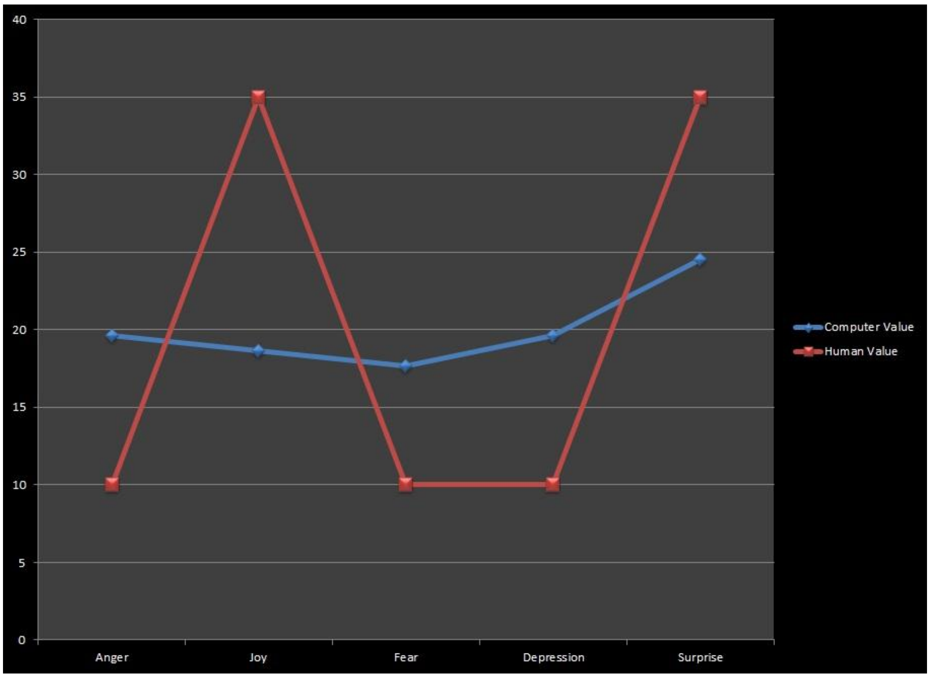
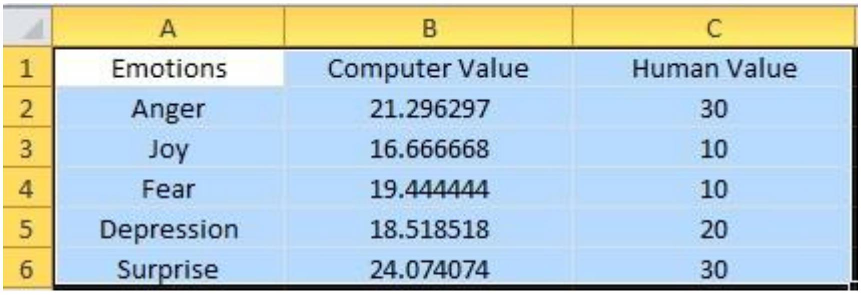


Figure 11. Line Chart 1

**2.	Test 2:-**



	A	B	C
1	Emotions	Computer Value	Human Value
2	Anger	21.296297	30
3	Joy	16.666668	10
4	Fear	19.444444	10
5	Depression	18.518518	20
6	Surprise	24.074074	30

Figure 12. Data Set 2

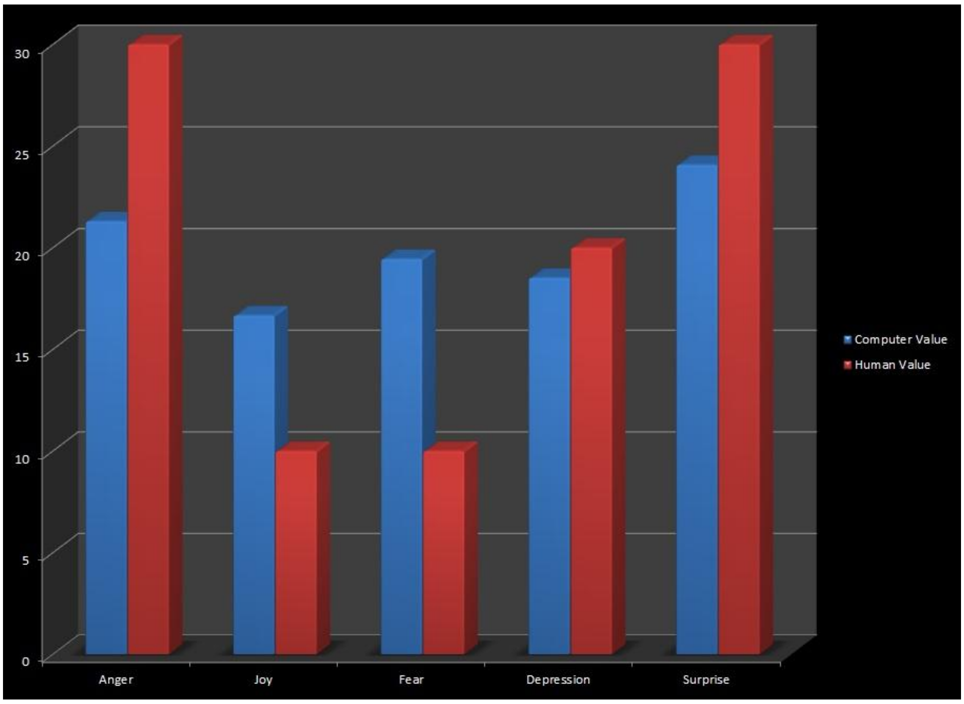


Figure 13. Column Chart 2

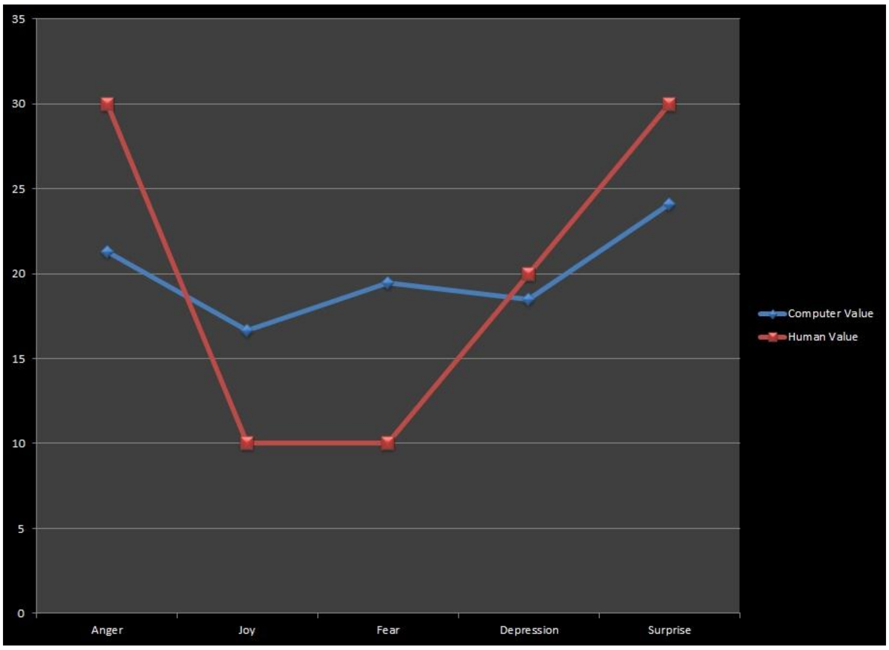
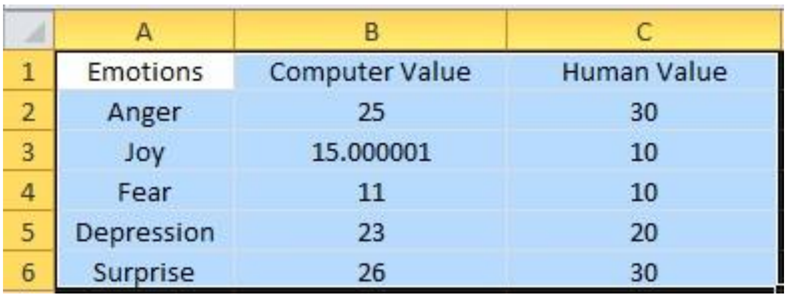


Figure 14. Line Chart 2

3. Test 3:-



	A	B	C
1	Emotions	Computer Value	Human Value
2	Anger	25	30
3	Joy	15.000001	10
4	Fear	11	10
5	Depression	23	20
6	Surprise	26	30

Figure 15. Data Set 3

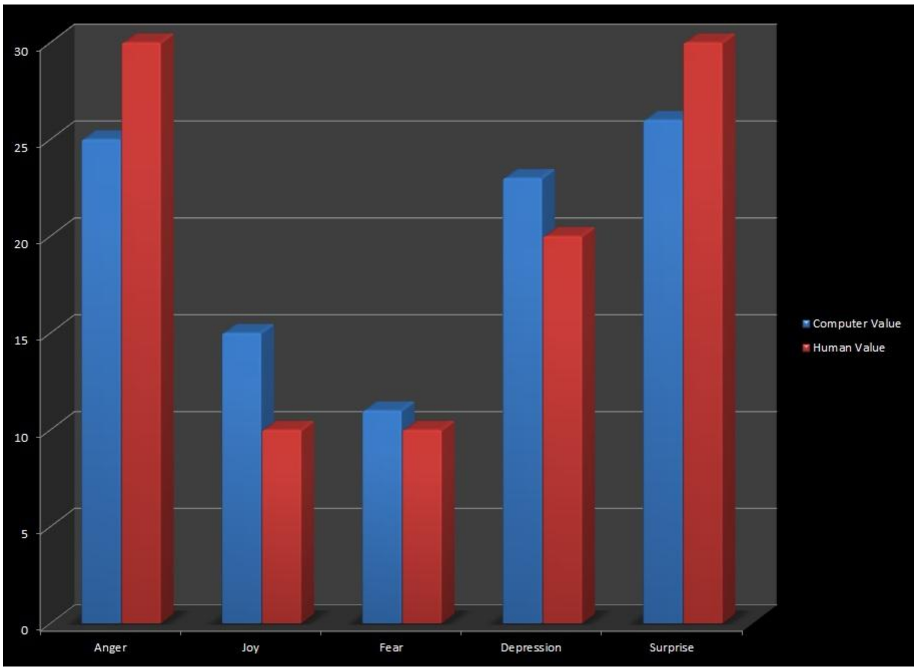


Figure 16. Column Chart 3

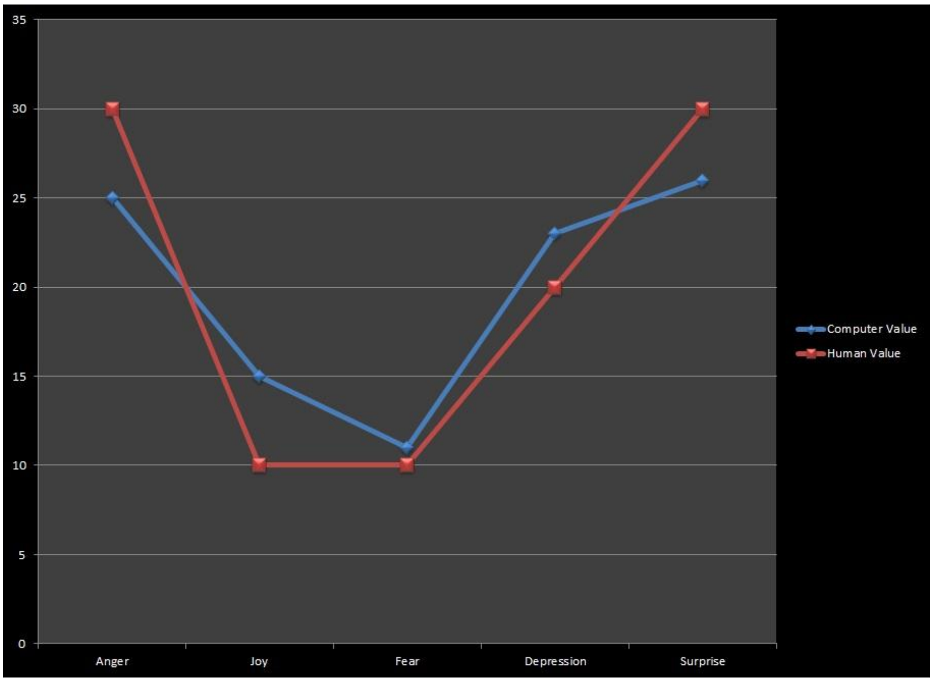


Figure 17. Line Chart 3

VII. Conclusion

We can observe that the first test could not give much expected results. However the second and third tests have shown improved results and the program results are quite close to the human results.

The advantage of this work is that this test proves that emotions can be implemented in computers and that it is also feasible.

One limitation of this implementation remains that only one word was considered for analyzing the emotions. Further research is required to use complete sentences for the purpose.

This work has huge possibilities in the near future. Computers with emotions can be used to create more efficient robots. This concept can also be used in modern warfare to create super soldiers. Machines with intelligence not only logical but also emotional can be used in security purpose, daily life task performing robots and so on. The most important thing is that simple existing tools are enough to implement this concept and this can also be made platform independent if developed further.

Acknowledgements

I would like to acknowledge my institute Birla Institute of Technology Mesra, Patna Campus for providing me the infrastructure to implement this research. I would also like to acknowledge my parents and my brother whose constant support helped me during the difficult times I faced while implementing this research.

References

Websites:

[1]. http://en.wikipedia.org/wiki/Chatbot
[2]. http://en.wikipedia.org/wiki/Chatterbot
[3]. http://en.wikipedia.org/wiki/Cleverbot
[4]. http://en.wikipedia.org/wiki/Elbot
[5]. http://en.wikipedia.org/wiki/Jabberwacky
[6]. http://en.wikipedia.org/wiki/ELIZA
[7]. http://en.wikipedia.org/wiki/Artificial_Linguistic_Internet_Computer_Entity
[8]. http://en.wikipedia.org/wiki/Linguistics
[9]. http://en.wikipedia.org/wiki/Minimalist_syntax
[10]. http://en.wikipedia.org/wiki/Phrase_structure_rules
[11]. http://en.wikipedia.org/wiki/Syntax
[12]. http://en.wikipedia.org/wiki/Sentence_diagram
[13]. http://en.wikipedia.org/wiki/Propositional_calculus#Inference_rule
[14]. http://en.wikipedia.org/wiki/Propositional_calculus
[15]. http://en.wikipedia.org/wiki/Logical_equivalence
[16]. http://en.wikipedia.org/wiki/Predicate_logic
[17]. http://en.wikipedia.org/wiki/Rule_of_inference
[18]. http://en.wikipedia.org/wiki/List_of_rules_of_inference

Books:

[19]. Daniel Goleman, Emotional Intelligence (London, Bloomsbury Publishing Plc, 1996)
[20]. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach (Pearson Education Inc., 2006)
[21]. Patrick Henry Winston, Artificial Intelligence (Pearson Education Inc., 2002)

Journal Papers:

[22]. Hai Zhuge,”Interactive semantics,” Elsevier Artificial Intelligence 174, pp. 19-204, 2010.
[23]. Stephen J. Read, Lynn C. Miller, “Virtual Personalities: A neural network model of personality,” Personality and Social Psychology Review, Volume 6, No. 4, pp. 357-369,2002.
[24]. Mandeep Kaur, Poonam Pandey,”Developing brain computer interface using fuzzy logic,” International Journal of Information Technology and Knowledge Management, Volume 2,No. 2, pp. 429-434, July-December 2010.
[25]. Emiliano Lorini, Francois Schwarzentruber,”A logic for reasoning about counterfactual emotions,” Elsevier Artificial Intelligence 175, pp. 814-847,2011.
[26]. Mark G. Orr,Roxanne Thrush,David C. Plaut,”The theory of reasoned action as parallel constraint satisfaction: Towards a dynamic computation model of health behavior,” PLOS One, e62490, Volume 8, Issue 5, May 2013.
[27]. Indrajit Sinha, Dr. Kanhaiya Lal, “A New Concept On Thinking Machines (Cyber Personality),” International Journal of Computer System, Vol 2, Issue 1, pp. 25-29, January 2015.