# INTELLIGENT WEB PROGRAMMING AI SYSTEM USING AUTOMATIC SPEECH RECOGNITION TECHNIQUE

**NIRUPAN MARUTHAPPAN**
Computer Science and Engineering
*SRM UNIVERSITY* – KATTANKULATHUR
Chennai, India
m.nirupan23@gmail.com

**ABHINAV RAVICHANDRAN**
Computer Science and Engineering
*VELAMMAL ENGINEERING COLLEGE*
Chennai, India
abhinav24291@gmail.com

**Dr. E. Poovammal**
Professor and Head
Computer Science and Engineering Department
SRM UNIVERSITY - KATTANKULATHUR
Chennai, India
poovammal.e@ktr.srmuniv.ac.in

***ABSTRACT:***

**The objective is to develop an Artificially Intelligent System which helps in designing static web pages for the physically challenged people who have lost their limbs. This system detects the voice signal of the user who wants to create the webpage using Automatic Speech Recognition technique by implementing Hidden Markov Model[1][2] .It obtains the required parameters and creates the corresponding web pages using BBS system with the final decision authority rests with the user.**

**Keywords : KBS,BBS,ASR**

## I. INTRODUCTION

The Intelligent Web Programming System is an Artificially Intelligent system which takes the user's parameters and starts designing the web pages by itself. The final deciding authority lies with the user. There are two parts in this system. The first one is the voice recognition system using Hidden Markov Model (HMM) Technique[1][2] and then the voice signal to command signal conversion. The second one is the AI system which starts designing the web pages on the fly based on the given parameters. The intelligent aspect of the system is when the system predicts what the future web pages could be based on certain probability functions and then waits for the user's approval as a final call.

## II. VOICE RECOGNITION SYSTEM

Automatic Speech Recognition[3] (ASR) is a technology that allows a computer to identify the words that a person speaks into a microphone . This is a very difficult task, partially because the field is motivated by the promise of human-like performance under realistic conditions. The ultimate goal of the system is to find out the parameters which are necessary for creation of web pages by the intelligent system.

*ASR System overview*

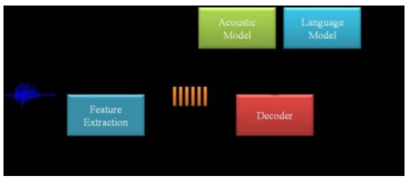


FIGURE 1 ASR ARCHITECTURE

There are several Automatic Speech Recognition Systems built before. We use the practical part of the ASR system built for the Arabic Automatic Speech Recognition System. The first module is a training module, whose function is to create the knowledge about the speech and language to be used in the system. The second subsystem module is the HMM models bank, whose function is to store and organize the system knowledge gained by the first module. The final module is the recognition module, whose function is to try to figure out what is the command given in the input speech given in the testing phase. This is done with the aid of the HMM[1][2] models.

### A) *Database*

The Command Interpretation Database contains a database of speech waves and their transcripts of the messages conveyed by the speakers. It was designed to train and test automatic speech recognition engines and to be used in identifying different types of parameters needed for creating web pages by the Intelligent Web Programming System.

TABLE 1

| Parameter List | Values |
|---|---|
| **Title** | The title goes here |
| **Header** | My new web page |
| **Bg colour** | Red |
| **Font style** | Times new roman |
| **Font size** | 12 |
| **Text** | Black |
| **Domain name** | www.pc.net |
| **Host name** | www |

As illustrated in the table 1 , the parameters of the system are shown. The experiment during during training period were an 8 KHZ sampling rate with a 16 bit sample resolution a 25 millisecond Hamming window duration with a step size of 10 milliseconds, 26 filter bank channels of which the number of MFCC coefficients and of which 0.97 are as the were the pre-emphasis coefficients.

Phoneme -based models are good at capturing phonetic details. Also context-dependent phoneme models can be used to characterize formant transition information, which is very important for recognizing the parameters of the web pages required for creation. The Hidden Markov Model Toolkit[2] (HTK) was used for designing and testing the speech recognition systems throughout all experiments.

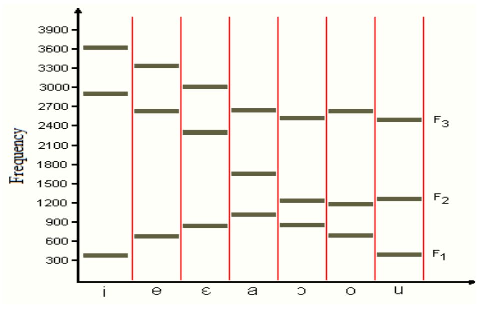
### B) *Error classes in spontaneous speech*

Common simple errors in sentence-like utterances (SUs) include filler words ("you know") as well as speaker edits consisting of a interruption points(IP) , an operational interregnum (like "I mean"), and a repair region as shown in the figure. These edit regions can be classified into 3 main groups:

1. In a repetition, the repair phrase is approximately identical to the error.
2. In a revision, the repair phase alters the error to the correct command stated.
3. In a restart fragment , the data introduces no new content.

The Johnson and Charniak (2004) approach referred to in this document, combines the noisy channel paradigm with a tree-adjoining grammar (TAG) to capture repeated elements. The TAG approach models the crossed word dependencies observed when the data incorporates the same or very similar words in roughly the same order , referred to as the rough copy.

### C) *Word Level ID Experiments*



This diagram denotes the frequency of the vowels used by the user during different phases of the experiment. F1 denotes the pre-programmed adaptation. F2 denotes the learned adaptation. F3 denotes the evolved adaptation.

### D) *Feature functions*

We aim to train our CRF model with sets of features with orthogonal analyses of the erroneous text, integrating parameters from multiple sources.

Lexical features including

-The lexical item and part of speech (POS) for tokens $t_i$ and $t_{i+1}$,

-distance from previous token to the next matching word/POS,

-whether previous token is partial word and the distance to the next word with same start

-the token's(normalized) position within the sentence.

JC04-edit: whether previous , next , or current word is identified by the JC04 system as an edit and/or a filler.

Language model features: the unigram log probability of the next word (or POS) token p(t), the token log probability conditioned on its multi-token history h $(p(t/h))^2$ , and the log ratio of the two(log (p(t/h)/p(t))) to serve as an approximation for mutual information between the token and its history as defined below,

$$I(t;h) = \Sigma_{h,t}\ p(h,t) \log [p(h,t)/p(h)p(t)]$$
$$= \Sigma_{h,t}\ p(h,t)\ [\log [p(t/h)/p(t)]]$$

This aims to capture unexpected n-grams produced by the juxtaposition of the erroneous and repair data.

## III. INTELLIGENT WEB PROGRAMMING SYSTEM

This is an Artificially Intelligent system which is built on several frameworks together. The several steps after the command interpretation are as follows.

1. Getting parameters from the user's speech from the ARS system.
2. Creating the web pages.

The first part was discussed extensively in the previous section.

Creating web pages – Types of execution

1. Pre-programmed adaptation

Here in this type , the user in his speech gives the full parameter lists he wants explaining every intricate detail of the required web page.

The system creates the required page according the process discussed later.

The following table describes the full detailed web page list the user discusses in his speech.

2. Learned adaption

Here the user in his speech just vaguely describes the required parameters he wants in his web page. The following table describes the vague description . The system process is explained in the coming section.

3. Evolved adaptation

Here the system predicts a possible pattern the AI system recognises in order to design the web page.

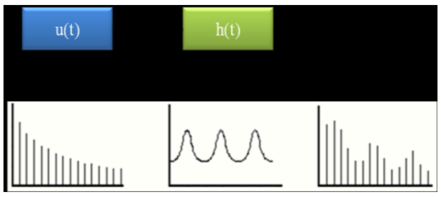


FIGURE 2

The diagrammatic and graphical representation of the initial speech signal u(t) to the digital signal h(t) and then the final conversion of the signal to the commands obtained from the database which are executed based on the function of time.

### A) *Types of AI systems*

#### 1. *Knowledge based systems*

The KBS are tailored for the simulation of the most abstract elements of thought. Thus KBS[6][8] has been very effective in simulating abstract ways of exhibiting intelligence, by successfully demonstrating theorems, solving problems, playing chess, etc. This system utilizes the information it has and compares it with the current problem to make a decision on the current premise to arrive at a solution.

#### 2. *Behavioural based System*

We can say that the goal of a BSS[7][10] is to offer the control of an autonomous agent. An agent is a system that has goals to fulfil, within an environment, which may be dynamic and complex. If the autonomous agent is able to adjust his goals in terms of what he perceives in his changing environment(his beliefs) it is also to be adaptive. If this adaptation is opportunistic, we can say that the autonomy and the adaptation themselves are of a higher order: intelligent.

### B) *System overview*

The system is of two parts both KBS[6] and BBS[10]. KBS is active during pre-programmed adaptation phase. The BBS starts its initiation during the learned adaptation phase. Then its dominance is cemented during the final phase.

### C) *Process*

The system consists of the following

-database
-web designing platforms
-HTML ,CSS styles library files.
-JavaScript ,PHP code generator.

The system reads the parameters and looks for the specific code patterns in the database. It is predefined database which contains the specific files locations of the pre-programmed codes written by default by the database developer.

### D) *Complex Web Programmed Development Kit*

This is a software which was designed to work as a common platform where several different web programming languages can be compiled and executed. Our AI system uses this web programming tool kit to compile the programming code it had obtained from the database based on the user's instructions.

### E) *Obtaining Source code file*

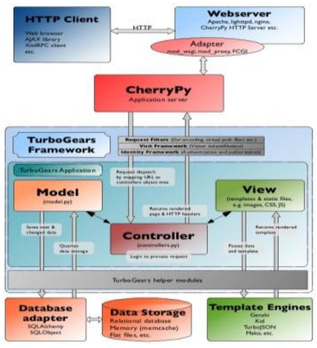
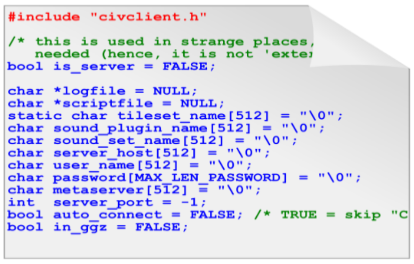


FIGURE 3

Execution of source code.



```
#include "civclient.h"

/* this is used in strange places,
   needed (hence, it is not 'exte
bool is_server = FALSE;

char *logfile = NULL;
char *scriptfile = NULL;
static char tileset_name[512] = "\0";
char sound_plugin_name[512] = "\0";
char sound_set_name[512] = "\0";
char server_host[512] = "\0";
char user_name[512] = "\0";
char password[MAX_LEN_PASSWORD] = "\0";
char metaserver[512] = "\0";
int  server_port = -1;
bool auto_connect = FALSE; /* TRUE = skip "C
bool in_ggz = FALSE;
```

FIGURE 4

Preview of web page

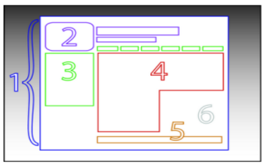


FIGURE 5

Finally the BBS system which involves in decision making for the creation of web pages based on past history.

## IV. DECISION MAKING:

Decision making can be regarded as the mental processes (cognitive process) resulting in the selection of a course of action among several alternative scenarios. Every decision making process produces a final choice. The output can be an action or an opinion of choice.

### A) *Steps:*

Each step in the decision making process includes social, cognitive and cultural obstacles to successfully negotiating dilemmas. There are seven steps:

1. Outline your goal and outcome.
2. Gather data.
3. Brainstorm to develop alternatives.
4. List pros and cons of each alternative.
5. Make the decision.
6. Take the action.
7. Learn from the decision taken.

*1) Outline your goal and outcome:*

The number of web pages to be created are determined. An idea about the required outcome is generated.

*2) Gather data:*

Inferences from the past are obtained. If required, interaction with the user takes place to get necessary data.

*3) Brainstorm to develop alternatives:*

All possible options concerning the parameters are identified. The corresponding child nodes are also found.

*4) List pros and cons of each alternative:*

All the choices are evaluated. The option that most resembles the user's wish is to be chosen.

*5) Make the decision:*

The attributes of the webpage to be generated are found.

*6) Take action:*

The webpage is generated with all the corresponding attributes and parameters set.

*7) Learn from the decision taken:*

The parameters of the webpage generated are stored in the knowledgebase for future reference.

*B) Decision trees:*

A decision tree is a map of the reasoning process. It can be used to explain why a question is being asked. Decision trees are powerful and popular tools for classification and prediction. The attractiveness of decision trees is due to the fact that, in contrast to neural networks, decision trees represent ***rules.*** Rules can readily be expressed so that humans can understand them or even directly used in a database access language like SQL so that records falling into a particular category may be retrieved.

*Decision tree* is a classifier in the form of a tree structure (see Figure 1), where each node is either:

- A ***leaf node*** **-** indicates the value of the target attribute (class) of examples, or
- A ***decision node*** **-** specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for

Lets consider the scenario where a colour is to be chosen out of two different each possible outcome of the test .

colours for a web page attribute. The following depicts the decision tree that can be adopted :

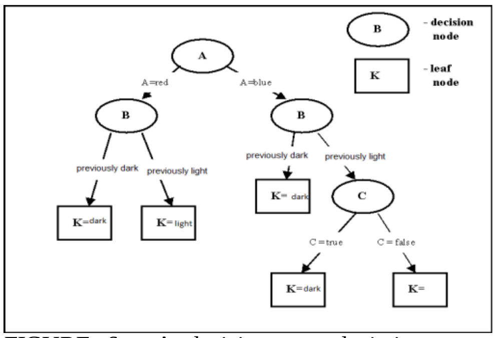


FIGURE 6 : A decision tree depicting example possibilities

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. Decision tree programs construct a decision tree *T* from a set of training cases

*.Decision tree algorithm:*

```
function ID3
Input:   (R: a set of non-target attributes,
          C: the target attribute,
          S: a training set) returns a decision tree;
begin
  If S is empty, return a single node with
    value Failure;
  If S consists of records all with the same
    value for the target attribute,
    return a single leaf node with that value;
  If R is empty, then return a single node
    with the value of the most frequent of the
    values of the target attribute that are
    found in records of S; [in that case
    there may be be errors, examples
    that will be improperly classified];
  Let A be the attribute with largest
    Gain(A,S) among attributes in R;
  Let {aj| j=1,2, .., m} be the values of
    attribute A;
  Let {Sj| j=1,2, .., m} be the subsets of
    S consisting respectively of records
    with value aj for A;
  Return a tree with root labelled A and arcs
    labelled a1, a2, .., am going respectively
    to the trees (ID3(R-{A}, C, S1), ID3(R-{A}, C, S2),
    .....,ID3(R-{A}, C, Sm);
```

Recursively apply **ID3** to subsets {Sj| j=1,2, .., m}
until they are empty
**end**

### *C) Tasks offered:*

The suggested system consists of three different possible processes. They are:

1. The requirements are specified explicitly by the user.
2. The requirements are specified by the user, but they are vague and subjective.
3. The requirements are generated by the agent using probabilities of occurrence of various elements by considering the past values.

The user may intervene in all processes.

In the decision making process, the probability of selecting an uncertain decision is very low as depicted by:

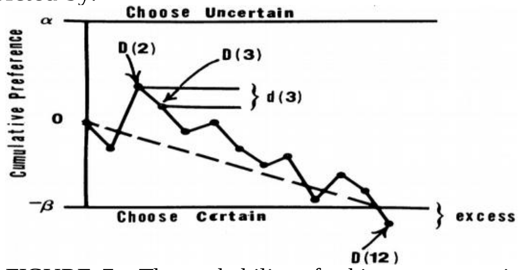


FIGURE 7 : The probability of taking an uncertain decision

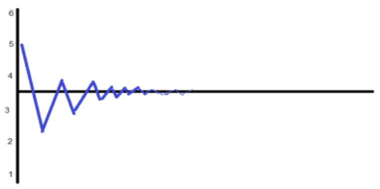


FIGURE 8 : On analysing six past outcomes that are similar it is inferred that the average converges to 3.5 or approximately to that of the third outcome.

### *D) Finding the probability of an event:*

Bayes theorem is a result in probability theory, which relates the conditional and marginal probability distributions of random variables. In some interpretations of probability, Bayes' theorem tells how to update or revise beliefs in light of new evidence: *a posterior.*

To derive the theorem, we start from the definition of conditional probability. The probability of event *A* given event *B* is

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

Likewise, the probability of event *B* given event *A* is

$$\Pr(B|A) = \frac{\Pr(A \cap B)}{\Pr(A)}.$$

Rearranging and combining these two equations, we find

$$\Pr(A|B)\,\Pr(B) = \Pr(A \cap B) = \Pr(B|A)\,\Pr(A).$$

This lemma is sometimes called the product rule for probabilities. Dividing both sides by Pr(*B*), providing that it is non-zero, we obtain Bayes theorem:

$$\Pr(A|B) = \frac{\Pr(B|A)\,\Pr(A)}{\Pr(B)}.$$

**Tournament selection** is a method of selecting an individual from a population of individuals. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population.

*Pseudo code:*

- choose k (the tournament size) individuals from the population at random
- choose the best individual from pool/tournament with probability p
- choose the second best individual with probability p*(1-p)
- choose the third best individual with probability p*((1-p)^2)

and so on...

## V. HIDDEN MARKOV MODEL FOR SPEECH RECOGNITION

There are several aspects of the model that must be defined before applying HMMs[2][3] for speech recognition.

### *A) HMM for Parameter estimation*

*Case 1:* State sequences of training sequences are known. We construct maximum likelihood estimators.

Let $A_{kl}$ be the number of transitions from state k into state l in training data (+ constant).

Let $E_k(x)$ be the number of emissions of symbol x from state k in training data (+ constant).

Transition estimator:

$A_{kl} = A_{kl} \ / \ \Sigma_{l'} A_{kl}$'

Emission estimator

$E_k(x) = E_k(x) / \Sigma\, E_k(x)$ '

*Case 2:* state sequences of training sequences are not known.

Viterbi training: We iteratively use the Viterbi algorithm to compute the most probable paths and set the estimators(from case 1) according to this data.

*Algorithm sketch:*

*Initialization :* Pick arbitrary model parameters.

REPEAT

```
Set all $A_{kl}$ and $E_k(x)$ to their constant value
    FOR each training sequence
  Compute most probable state paths(Viterbi)
  Add contribution to $A_{kl}$ and $E_k(x)$.
  Update $A_{kl}= A_{kl} / \Sigma_{l'} A_{kl'}$ and $E_k(x) = E_k(x) / \Sigma_k(x)$ '
  UNTIL stopping criterion is reached.
```

### *B) Filtering*

The task is to compute, given the model's parameters and a sequence of observations, the distribution over hidden states at the end of the sequence, i.e. to compute

$P(x(t)| y(1),...............,y(t))$

. This problem can be handled efficiently using the forward algorithm.

### *C) Probability of an observed sequence*

The task is to compute, given the parameters of the model, the probability of a particular output sequence. This requires summation over all possible state sequences:

The probability of observing a sequence

$Y=y(0),y(1),...........,y(L-1)$

of length *L* is given by

$P(Y) = \Sigma_x \ P(Y/X)P(X),$

where the sum runs over all possible hidden-node sequences

$X= x(0),x(1),.......,x(L-1)$

Applying the principle of dynamic programming , this problem, too, can be handled efficiently using the forward algorithm.

Most likely explanation. The task is to compute, given the parameters of the model and a particular output sequence, the state sequence that is most likely to have generated that output sequence (see illustration on the right). This requires finding a maximum over all possible state sequences, but can similarly be solved efficiently by the Viterbi algorithm .Smoothing

The task is to compute, given the parameters of the model and a particular output sequence up to time *t*, the probability distribution over hidden states for a point in time in the past, i.e. to compute

$P(x(k)|y(1),.......,y(t))$ for some $k < t$. The forward-backward algorithm is an efficient method for computing the smoothed values for all hidden state variables.

Statistical significance: For some of the above problems, it may also be interesting to ask about statistical significance. What is the probability that a sequence drawn from some null distribution will have an HMM probability (in the case of the forward algorithm) or a maximum state sequence probability (in the case of the Viterbi algorithm) at least as large as that of a particular output sequence? When an HMM is used to evaluate the relevance of a hypothesis for a particular output sequence, the statistical significance indicates the false positive rate associated with accepting the hypothesis for the output sequence.

## REFERENCES

1. L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, 77(2)(1989),pg. 257-286.
2.B.Juang and L. R. Rabiner, "Hidden Markov Models for Speech Recognition",Technometrics,33(3)(1991),pg. 251-272.
3. Yousef Ajami Alotaibi, "Is Phoneme Level Better than Word Level for HMM Models in Limited Vocabulary ASR Systems?" , The International Conference on Information Technology – New Generations (ITNG2010), pg. 332-337, Las Vegas, USA, April 12-14, 2010.
4. Maes,P. (1991)(ed.). Designing Autonomous Agents: Theory and Practice From Biology to Engineering and Back, MIT Press.
5. Maes,P.(1993). Modelling Adaptive Autonomous Agents. Journal of Artificial Life,1,(1-2),MIT Press.
6. Wooldridge,M.(1997). Agent-Based Software Engineering. IEEE Proc. Software Engineering 144(1), pg. 26-37.
7. Wooldridge , M. And N. R. Jennings (1995). Intelligent agents : Theory and practice. The Knowledge Engineering Review, 10(2), pg. 115-152.
8. Randall Davis (1987) 'Knowledge-Based Systems: The View in 1986' in W.Eric L.Grimson and Ramesh S.Patil (eds) AI in the 1980s and Beyond : An M.I.T. Survey (M.I.T. Press, Cambridge,Mass.) pg.13-41.
9. Vijay Vadhwana (1989) 'A Knowledge-Based System Approach to the Synthesis of Distillation Sequences' in Nigel Shadbolt (ed.) Research and Development in Expert Systems VI (C. U. P. ,Cambridge) pg. 263-75 .
10. D. A. Waterman (1986) A Guide to Expert Systems (Addison Wesley , Reading , Mass. )