

# **Algorithmic Aspects of Speech Recognition**

**Kunal Mittal**  
(<http://www.kunalmittal.com>)

**Beloit College, Beloit, WI, USA**

# Table of Contents

<b>1. ABSTRACT .....</b>	<b>3</b>
<b>2. INTRODUCTION .....</b>	<b>4</b>
2.1. WORK ON SPEECH RECOGNITION BY NON-COMPUTER SCIENTISTS: .....	5
2.2. COMPUTER SCIENCE AND SPEECH RECOGNITION: .....	5
2.2.1. <i>Finite Automata:</i> .....	5
2.2.2. <i>Hidden Markov Model (HMM):</i> .....	6
2.2.3. <i>Example HMM:</i> .....	7
2.2.4. <i>Markov Chain and Markov Sources (MS)<sup>[9]</sup></i> .....	9
<b>3. HOW IS THE HIDDEN MARKOV MODEL USED? .....</b>	<b>10</b>
3.1. WHAT IS SPEECH RECOGNITION? .....	10
3.1.1. <i>Isolated Word Recognition (IWR):</i> .....	10
3.1.2. <i>Continuous Speech Recognition (CSR):</i> .....	11
3.2. PROBABILITY EVALUATION: .....	11
3.3. FORWARD PROCEDURE <sup>[7]</sup> .....	12
3.4. THE VITERBI ALGORITHM <sup>[3]</sup> .....	13
<b>4. A* SEARCH ALGORITHM:.....</b>	<b>13</b>
4.1. INPUTS / GIVENS TO A* .....	13
4.2. EXAMPLE – HMM USED TO RECOGNIZE SYLLABLES FROM PHONES: .....	15
<b>5. REFERENCES:.....</b>	<b>17</b>

## **1. Abstract**

After motivating how speech recognition became of interest and presenting a short history of how various challenges have been addressed, this paper will describe how various fields, other than computer science, have been applied to one or more speech recognition problems. It will continue on to describe the theory and implementation of the Hidden Markov Model, and present an example of how it is used. The paper will then show how the A\* algorithm can be used to solve this problem.

## 2. Introduction

Computer speech recognition has been a topic of interest to many people for about four decades. In 1956, a paper by Fry and Denes<sup>[1]</sup> outlines a system that can recognize a few distinct words. It can be inferred that this paper was among the first works in the subject, as it does not cite any previous works. The only other work that seems to predate the paper by Fry and Denes is a book by Potter, Kopp and Kopp on *Visible Speech* in 1947<sup>[2]</sup>. In the early seventies the folks at the artificial intelligence laboratories supported by the Advanced Research Projects Agency (ARPA) announced their interests in the problem given the progress in other areas of artificial intelligence and some papers on single, isolated word recognition by a single talker. They published a report in 1971 called *A Tutorial on Speech Understanding Systems*<sup>[2]</sup>.

*“It is hard to gauge the success of an attempt at speech recognition even when statistics are given. In general, it appears that speech recognition around 95% correct can be achieved for clearly pronounced words, isolated words from a chosen small vocabulary (the digits, for instance) spoken by a few chosen talkers. Better results have been attained for one talker. Performance has gone down drastically as the vocabulary was expanded and appreciably as the number and variety of talkers were increased. It is not easy to see a practical, economically sound application for speech recognition with this capability.”*<sup>[2]</sup>

## **2.1. Work on Speech Recognition by non-Computer Scientists:**

Speech recognition has intrigued many scientists and researchers from various different fields, other than computer science. Thus this problem is very clearly, interdisciplinary in nature. Some examples are:

1. Physics – a branch of acoustics that is concerned with the physical connection between the sound waves emitted by the “noise” source and how the human hearing mechanism can perceive the noise and distinguish the various elements.
2. Linguistics – the study of the relationship between the sounds, grammar and syntax of the language, semantics, pronunciations and other such aspects of a spoken signal.
3. Signal Processing – being able to separate and distinguish “meaningful” patterns within the sound signals. It is evident that there are always going to be some extraneous sound waves, distortion and other abnormalities that can affect the speech recognition procedure.

## **2.2. Computer Science and Speech Recognition:**

In order to understand various models and algorithms in computer science that provide reasonable solutions to some aspects of speech recognition, it is important to present a background in finite automata theory. These provide the basis to the Hidden Markov Model on which most of the Speech Recognition algorithms are based.

### **2.2.1. Finite Automata:**

*Definition 1:* Finite State Machine or Finite Automata (FA)

A finite automaton is a 5-tuple (A, B, C, D, E) in which

A = finite set of states (various states the problem can be in). These may or may not be acceptable states.

B = Finite set of input symbols (inputs to the problem).

C = the initial state (a state that must exist in A or to be more precise, it must exist within E which is defined below).

D = is the next state function (if in state “x”, a function that describes a method to move to another state in A).

E = set of acceptable states (a list of states that one are valid or possible to reach from some state C, using a next state function D). E is a subset of A.

By defining an abstract machine in this way, it is possible to take a problem and represent it as a FA using valid transition tables. A clear definition of a problem, and the states leading to the solution can be presented. In addition, the intermediate states, with a list of acceptable and unacceptable states, from any state in the solution of the problem can be described. This is done by using a directed, weighted graph, an output probability matrix and a initial state vector.

### **2.2.2. Hidden Markov Model (HMM):**

As said earlier, HMM's are used extensively to model and solve problems of speech recognition. The term “Hidden” comes from the fact that no one has any idea of which state the problem will be at time “t” or even at time “1”. It is defined as follows:

Definition 2: Hidden Markov Model

Let X be an alphabet of M symbols. A HMM is a quintuple

$\lambda = (A, B, C, D, E)$  where

A = the number of states.

B = the number of symbols that each state can output / recognize.

C = A\*A matrix such that  $C_{xy}$  is the probability of moving from state “x” to state “y”. It

is obvious that  $\sum_y C_{xy} = 1$ .

D = observation probability (if  $D_y$  (“ $\Pi$ ”) is the probability of recognizing the symbol

“ $\Pi$ ” when in state “y”).

E = is the initial state probability (probability of being in state “x” at time 1).

This definition seems really abstract and is difficult to comprehend at first look. Consider the following example to help make this more clear and intuitive.

### 2.2.3. Example HMM:

Consider that Mr. X is organizing a Christmas party for his grand children. He has been doing this for several years in a row. One of his grandchildren notices a pattern in his grandfather’s choice of gifts, and models it as follows:

Each year Mr. X uses a box to pack the gifts in. He always chooses randomly among a Red, Blue, Green, Orange or Yellow box. He then chooses one of four gifts, clothes, Lego sets, dolls or balls. The probabilities for both of these events, is varies based the choice he made in the previous year. This can be described as a HMM in which:

A = Gift Box Colors [ R, B, G, O, Y ]

B = Gift Choices [ C, L, D, B ]

C and D are described using a transition and output matrix shown below. E is described using a vector.

*Transition Probabilities*

	<b>R</b>	<b>B</b>	<b>G</b>	<b>O</b>	<b>Y</b>
<b>R</b>	.40	.20	.20	.10	.10
<b>B</b>	.25	.25	.20	.05	.25
<b>G</b>	.125	.10	.125	.40	.25
<b>O</b>	.20	.20	.20	0.0	.40
<b>Y</b>	.20	.20	.20	.40	0.0

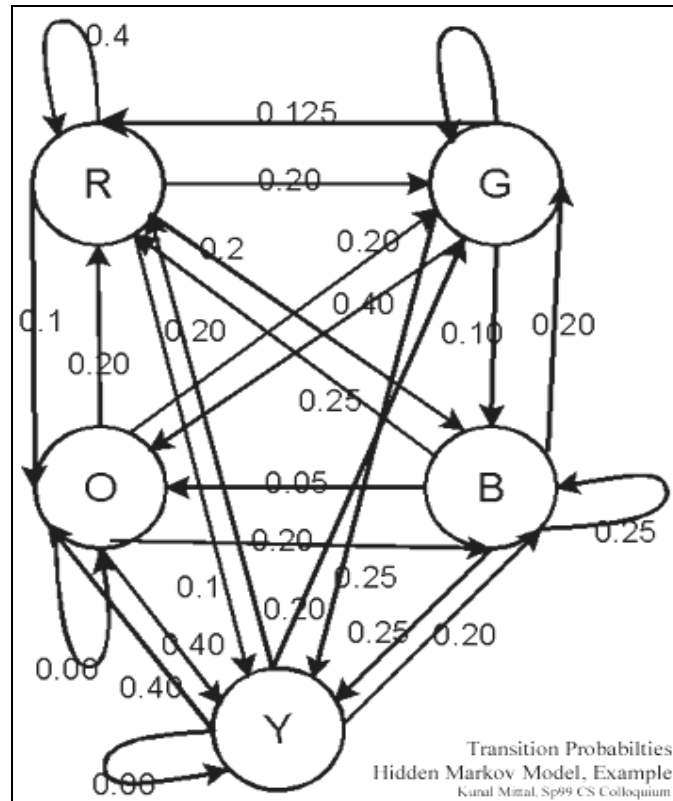
*Output Probabilities*

	<b>C</b>	<b>L</b>	<b>D</b>	<b>B</b>
<b>R</b>	.25	.25	.25	.25
<b>G</b>	.20	.40	.10	.30
<b>B</b>	.25	.20	.45	.10
<b>O</b>	.40	.30	.10	.20
<b>Y</b>	.50	.10	.20	.20

*Probability of starting Box Color (at time  $t=1$ )*

<b>R</b>	<b>B</b>	<b>G</b>	<b>O</b>	<b>Y</b>
.20	.20	.20	.20	.20

To better comprehend these matrices, they can be represented in the form of a directed graph. In the graph shown below, the colors of the boxes are nodes and it should be clear that the graph is fully connected.



A similar graph can be drawn for the output probability table and the initial start state vector.

#### 2.2.4. Markov Chain and Markov Sources (MS)<sup>[9]</sup>

In the above example, a transition probability matrix was constructed. Notice that the transition probabilities at time  $t = "x"$  depended directly and only on time  $t = "x-1"$ . Such a matrix is known as a Markov matrix. The process is known as a Markov process or a Markov chain. The matrix and a description of how it is constructed are known as a Markov source (MS).

### **3. How is the Hidden Markov Model Used?**

Before this question can be answered, it is necessary to formalize what is meant by speech recognition.

#### **3.1. What is Speech Recognition?**

Speech recognition is the process of identifying the individual words spoken by a person or machine and reconstructing the sentence. The main aspects of the process are to process the input signals, pass them through an acoustic recognizer (which is based on a phonetic representation of the sounds), then pass them through a word recognizer (which is based on a word lattice) and finally put the individual words together. In a broad sense the process of speech recognition is divided into two distinct classes.

1. Isolated word recognizer (IWR)
2. Continuous speech recognition (CSR)

##### **3.1.1. Isolated Word Recognition (IWR):**

The IWR is supplied with a lexicon (which is a dictionary of the phonetic representations for each word in the language) and a search algorithm. The acoustic models are considered to be Markov Sources over the language. The search algorithm needs to maximize the likelihood that a given observation sequence (input) matches a word in the acoustic model, taking one word at a time.

### 3.1.2. Continuous Speech Recognition (CSR):

The only addition to the IWR to make it a CSR is a language model or grammar. The CSR recognizes one word at a time, using the acoustic model and the language model as a heuristic. In this way it is more likely to get a correct output sequence, given a particular input sequence. It should be clear that this can become a very time consuming procedure, as the size of the language model and acoustic model grow at least exponentially, if not faster.

### 3.2. Probability Evaluation:

Given an observation sequence  $Y = y_1, y_2, \dots, y_n$ , and the HMM model  $K$  (i.e. the probability  $\Pr(Y|K)$ ), calculate the probability of being in any particular state at a particular time. Let  $Q (q_1 .. q_t)$  is a sequence of  $t$  states.

$$\Pr(Y | K) = \prod_{t=1}^T \sum_{i=1}^N \Pr(q_t = i) b_i(x_t) \quad [7]$$

This can be applied to the gift boxes and toys example described earlier. The likelihood of the grandfather to first pick Lego as the gift, followed by clothes and then dolls in subsequent years, can be determined.

Assume that the green box has been picked for the first year. (This is a fair assumption for purposes of calculations, due to the nature of the initial choice vector). Now looking at the output matrix, it can be inferred that there is a 40% chance that Lego will be the gift. So the probability of being in this state at time  $t=1$  is 0.08 (.20 for choosing a green gift box and then .40 of choosing Lego once the green box has been chosen). Now the calculations become slightly more complicated. Calculate from here, the probability of choosing clothes as the second gift for each of the box choices. For example we see that

the probability of choosing [R,G,Y,O,B] once in G is [.125, .1 , .125, .4, .25], and for each of these the probability of now choosing clothes as the gift is [.25, .2, .25, .4, .5]. Thus the probability of choosing clothes as the next gift is [.125\*.25 + .1\*.2 ... ] = 0.3675. Then this calculation can be carried on to get the probability of choosing dolls as the gift at time t=3. This value turns out to be 0.08 \* 0.3675 \* a. The calculation of “A” requires a complete matrix multiplication of the transition matrix, with a single column of the output matrix as there is no way of knowing which row to use from the transition matrix, at time t=2.

### 3.3. Forward Procedure <sup>[7]</sup>

The purpose of this is to calculate the maximum likelihood to be in state “y” at time t+1, given a list of states for time t=1..t. This can be done using induction.

$$\alpha_t(y) = \Pr(x_1 \dots x_t, q_t = y | K)$$

i.e. the probability at time t, that a partial sequence X has been observed and we are now in state “y”. The forward function can be described inductively as:

Step 1: Base Case:

$$\alpha_t(y) = \prod_y b_y(x_1) \quad , \quad 1 \leq y \leq N$$

Step 2: Induction

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(x_{t+1}) \quad , \quad 1 \leq t \leq T-1, 1 \leq j \leq N$$

Step 3: Result

$$\Pr(X | K) = \sum_{i=1}^N \alpha_T(i)$$

### 3.4. The Viterbi Algorithm<sup>[3]</sup>

This algorithm is used to find the optimal sequence,  $Q = (q_1, q_2 \dots q_N)$  for a particular input sequence  $X = (x_1, x_2 \dots x_N)$  through a state space  $G$  (language model represented as a HMM). It will be clear later on, how and why this is used.

To calculate this we need to define

$$\gamma_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_{t-1}, q_t = i, x_1 x_2 \dots x_{t-1} | G] \quad [3]$$

This represents the highest probability of a path, at time  $t$ , for the first  $t$  observations. This can inductively be applied until we get a complete transcript.

## 4. A\* Search Algorithm:

It is now possible to describe how all this fits together to form a simple SEARCH algorithm to solve the problem of IWR. It is intuitively obvious how it can be expanded to solve CSR. This paper will focus on the A\* algorithm, originally from the folks in artificial intelligence.

To solve the problem of ISR, the A\* algorithm is applied at various levels of abstraction, depending on how sophisticated is the software that is implementing it.

### 4.1. Inputs / Givens to A\*

Let  $P$  be the alphabet of phones. For each  $f \in P$ , where  $|P| = X$  (approximately 50 for US English), build  $HMM_f$ .  $HMM_f$  is a directed graph with exactly one source and one sink.

It is generally between four to seven states long. Now  $HMM_f$  can be thought of as the

acoustic model of a given pronunciation of phone  $f$ . It is a way of calculating, how likely is it that a given observation  $X \in P^*$  acoustically matches a given phone. This is given by  $\Pr(X | \text{HMM}_f)$ . Once a HMM has been built for each  $f \in P$ , we can obtain the acoustic model for a word  $w \in X$ , by replacing each  $f \in P$ , of a Markov source of the word ( $MS_w$ ) with  $\text{HMM}_f$ . This extends to the following two steps.

1. Construct the HMM for syllables, using the HMM of phones.
2. Construct the HMM for words, using the HMM of syllables.

This can be formalized as:

Let  $P_j$  be the alphabet at layer “ $j$ ”. The lexicon at layer “ $j$ ” is a set of directed graphs. We can obtain the HMM as follows:

1. Training procedure: Build a HMM for each unit  $P_j$  using alphabet  $F_j$ .
2. Assume  $P_{j-1}$  exists. For each graph at level “ $j$ ”, compute  $MS_j$ . Combine  $MS_j$  and  $\text{HMM}_{j-1}$ , to obtain an acoustic model of  $P_j$ .

Thus there now exists three HMMs, for phones, syllables and words. The  $A^*$  algorithm receives as input a sequence of phones  $X$ . It is able to distinguish between individual phones using some method, the description of which is beyond the scope of this paper. Each phone is made up of four to seven acoustic tokens (for US English). For each  $x_j$ , the  $A^*$  algorithm needs to find a sequence of acoustic tokens that is a “optimal” match. Thus the  $A^*$  outputs a sequence of phones  $Y$ , which is passed in as input to the algorithm, this time using the HMM for syllables and so on until finally a list of words is outputted.

**4.2. Example – HMM used to recognize syllables from phones:**

Assume that there are only 3 possible phones (x, y, and z). A combination of these three phones produce five syllables (xyz, xzy, zxy, yyzx and zzxy) each of length three or four. We use a HMM to show how the A\* algorithm can recognize an observation sequence X of phones to be one of the five syllables. The inputs to the A\* for this level would be the following matrices that form the HMM and a cost function. A description of the cost function is beyond the scope of this paper.

<b>HMM</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
<b>X</b>	0.00	0.65	0.35
<b>Y</b>	0.25	0.25	0.50
<b>Z</b>	0.40	0.40	0.20

The above table describes the HMM associated with the phones. For example, we can see that once in state “x” there is a 65% chance that the next state would be “y” and a 35% chance that the next state would be “z”. Furthermore, an initial state vector is used to estimate the probabilities of being at one of the states at time 0.

<b>Initial State</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
<b>Vector</b>	0.40	0.20	0.40

Given these two tables, it is possible to calculate the probability tables for time t=1..3. It should be obvious that since the maximum length of a syllable is four, the A\* algorithm will be at most three levels deep. The following tables show the probabilities for once in a particular state, what is the probability of the next state being  $Q \in \{x,y,z\}$ . Note that the

tables are not complete. They omit states that have probability zero for the given language model as described above.

<b>T=1</b>	<b>Xx</b>	<b>Xy</b>	<b>Xz</b>	<b>Yx</b>	<b>Yy</b>	<b>Yz</b>	<b>Zx</b>	<b>Zy</b>	<b>Zz</b>
<b>X</b>	0	0.26	0.14	0	0	0	0	0	0
<b>Y</b>	0	0	0	0.05	0.05	0.10	0	0	0
<b>Z</b>	0	0	0	0	0	0	0.16	0.16	0.08

The above table describes the probabilities of each of the possible syllables of length two phones. For example, if the first phone was “x”, the probabilities that the next phone will be “y” is 0.26. Note that if the first phone is “x”, it is not possible to have “Yx” or some such other phone at time t=1. Thus it is a clear waste of memory to store these extra possibilities. The next time state table for time t=2 is an example of where these extra states are not stored. All combinations of the phones of length three that are not possible (i.e. have a probability zero) are not stored.

<b>T=2</b>	<b>XYz</b>	<b>XZy</b>	<b>YYz</b>	<b>ZZx</b>	<b>ZYx</b>	<b>YZx</b>	<b>ZXy</b>
	0.091	0.056	0.025	0.032	0.064	0.04	0.032

The table for t=2 shows that if the first two symbols were known, what are the probabilities of getting  $Q \in \{x,y,z\}$  as the third symbol. Similarly a table for the final iteration is drawn.

<b>T=3</b>	<b>YYZx</b>	<b>ZZXy</b>
	0.001	0.001024

## 5. References:

- [1] IEEE Symposium on Speech Recognition (1974: Carnegie-Mellon University), edited by D. Raj Reddy, *Speech recognition: invited papers presented at the 1974 IEEE symposium*, New York: Academic Press, 1975.
- [2] Newell, Allen (1971), "A Tutorial on Speech Understanding Systems" in *Speech Recognition: invited papers presented at the 1974 IEEE Symposium* pp.3 - 54
- [3] Rabiner, Lawrence R. and Biing-Hwang Juang, *Fundamentals of speech recognition*, Englewood Cliffs, N. J.: PTR Prentice Hall, c1993.
- [4] John C. Martin: *Introduction to Languages and the Theory of Computation*, McGraw-Hill, Inc., 1991.
- [5] Stephen A. Book: *Statistics, Basic Techniques for solving applied problems*, McGraw-Hill, Inc., 1977.
- [6] Richard J. Larsen and Morris L. Marx: *Introduction to Mathematical Statistics and its applications*.
- [7] Adam L. Buchsbaum and Raffele Giancarlo, *Algorithmic Aspects of Speech Recognition: An Introduction*, ACM Journal of Experimental Algorithms. Vol. 2, 1997.
- [8] 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings, edited by Sadaoki Furui, B. H. Juang and Wu Chou, 1997 IEEE Inc.
- [9] Chris Rorres and Howard Anton, *Applications of Linear Algebra*, 3<sup>rd</sup>. Edition.