User Behavior Analysis Using Alignment Based Grammatical Inference from Web Server Access Log

Ramesh Thakur, Suresh Jain, and Narendra S. Chaudhari

Abstract—**Application of data mining technique to the World Wide Web refers to as Web mining. Web based origination collects large volume of data for their operation. Analysis of such data can help the organization for better working (Marketing strategy, services, evaluation of effectiveness, promotional campaigns etc). This type of analysis require discovery of meaningful relationships from the large collection of primarily unstructured data stored in Web server access logs. We propose a new approach for automatically learning (context-free) grammar rules form server access log text (positive set) samples, based on the alignments between the sentences. Our approach works on pairs of unstructured sentences that have one or more words common.**

Index Terms—**Web usage mining, computational learning, grammatical inference, alignment profile, information extraction.**

I. Introduction

We consider the web as the largest "knowledge base" ever developed and made available to the public. It becomes increasingly necessary for the users to utilize automated tools and analyze their uses pattern. This factor gives rise to the necessity of creating server side intelligent system that can efficiently mine for Knowledge of server uses. Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web. Discovery of user access pattern form web server is known as web usage mining. The web server uses detail is generally gathered automatically by web server in web server access logs which is unstructured data sets (text file). Web uses mining has several applications [1] such as analysis of massive volume of click stream or click flow data, Personalization for a user, determining access behavior of users, etc.

Information extraction from textual data has various applications, such as semantic search [2]. If the sentences confirm to a language described by a known grammar, several technique exist to generate the syntactic structure of these sentences, parsing [3] is one of such technique that rely on knowledge of grammar. In automated grammar learning, the task is to infer grammar rules from given information about the target language. The sentences are given as examples for such learning. If the example belongs to the target language, it is called positive example otherwise it is called negative example. In fact Gold [4] shows that not all language can be inferred from positive examples only. A language that can be inferred by looking at a finite number of positive examples only said to be identifiable in the limit [4]. As per this theorem, it is not possible to identify the target language form only positive examples. One main approach for learning some subclasses of regular language is by splitting the states in the deterministic finite automata (DFA) [5]. Prefix tree acceptors are often constructed from the given sample as a starting DFA, and they are useful for modeling positive samples. Other approaches include learning by queries [6], learning by structural information [7], learning subclass of language [8], learning by genetic algorithm [9], neural networks [10], Markov approaches [11] and other related work can be found in [12]-[15].

In this paper we propose a grammar inference methodology for web server log file of unstructured data to automate the construction of context free grammar rules and facilitate the process of information extraction. We are using Grammatical Inference methodology based on alignment between texts of given collection of sentences of the web server log text file (un-structured document). This method works on pairs of unstructured sentences that have one or more common words. When two sentences are divided into two parts having equal parts (same set of words) and unequal parts (different set of words then these parts are taken as possible constituents of the grammar.

II. Web Logs

A web log file [16] records activity information of web user request on web server. The main source of raw data is web server log which is stored on web server for debugging purpose. A log files are stored at three different places (i) Server-side Log (ii) Proxy- side Logs (iii) Client-side Logs.

A. Web Log Structure

Web server logs are plain text (ASCII) files that are independent from the server platform. Generally there are four types of server logs: transfer log, Agent log, Error log and Referrer log.

A web log [17] is the file to which the web server writes information each time a user request a source from that particular site. Most of the web server uses the common log format. Following fragments are the server log file entry [18].
123.123.123.123 - - [26/Apr/2000:00:23:48 -0400] "GET /pics/wpaper.gif HTTP/1.0" 200 6248 "http://www.jafsoft.com/asctortf/" "Mozilla/4.05 (Macintosh; I; PPC)"

Manuscript received August 15, 2012; revised December 21, 2012.

Ramesh Thakur is with the International Institute of Professional Studies, Devi Ahilya University, and Indore, India (e-mail: r_thakur@rediffmail.com).

Suresh Jain is with the KCB Technical Academy, Indore, India (e-mail: suresh.jain@rediffmail.com).

Narendra S.Chaudhari is with the Indian Institute of Technology, Indore, India (e-mail: nsc183@gmail.com).

This reflects the information as follows.

1) Remote IP address or domain name: An IP address is a 32-bit host adders defined by internet protocol. A domain name is used to determine a unique IP address for any host on the internet.
2) Authuser: User name and password when the server require user authentication.
3) Entering and exiting date and time.
4) Mode of request: GET, POST, or HEAD.
5) Status: The HTTP Status Code returned to the client.
6) Bytes: the content length. etc.

III. Alignment Based Learning

Alignment Based Learning (ABL) is based on alignment information [19]. In ABL Pairwise alignment for each pair of the input sentences is done by finding equal and unequal parts. Pairwise alignment is an arrangement of two sequences, which shows where the two sequence are similar and where they differ. A good alignment shows the most significant similarities, and least differences. A score is assigned to an alignment called alignment score, to measure the goodness of an alignment. Scoring scheme is usually defined on the pairing of different constituents and gap penalty for shifts in the alignments. An example of alignment for the following two sentences:

Boy likes fresh red, green apple
All Boy likes to eat red, green apple
are
[Boy likes] fresh [red, green apple]
All [boy likes] to eat [red, green apple]

Words that are located above each other and that are equal in alignment are called match. The shifts caused by insertion or deletion are called gaps. In alignment based system, more gaps means less similarity. Words that are located above each other and that are equal in the alignment are called substitutions. In an alignment, if there is a substitution, then two sub-sentences are said to be aligned in the same slot. Here the slot denotes that the sub-sentences are located in the alignment. For example "fresh" and "to eat" are aligned in the same slot, which are shown in the brackets.

Boy likes {fresh} red, green apple
All Boy likes {to eat} red, green apple

In the alignment phase, the matched parts of sentences are considered as possible constituents. Non-terminals are assigned as they possibly generate the constituents. Such assignments are called hypotheses.

[Ram] See the [large, green] orange
[My mother] See the [yellow] orange

The above hypothesis is used to create the grammar rules by assigning new symbols representing the sub-sentences, which are also called constituents that are in square bracket pairs.

S→A See the B orange A→Ram
A→My mother B→large green
B→yellow

The sentence is of unknown language then it is very hard if not impossible to say anything about their language only we can conclude it is a sentences. However, if two sentences are available, it is possible to find parts of the sentences that are the same in both and parts that are not (provided that some words are same and some words are different in both sentences). The comparison of two sentences falls into one of the three different categories.

1) The sentences are completely different.
2) All words in the two sentences are the same
3) Some of the words in the sentences are same in both and some are different.

A. Alignment Based Grammar Inference

The sentences belonging to third case, having two possibilities for selecting constituents for grammar (CFG) rule extraction i.e. (1) Select equal parts as constituents (2) select unequal parts as constituents. Without the loss of generality consider the following simple example.

1) Mother eats biscuit.
2) Mother eats cake.
3) Mother eats biscuit.
4) Mother eats cake.

In case one unequal part (underlined) parts are selected as constituents. In case two the equal parts (underlined) parts are select as the constituents. The resulting grammars are shown bellow. (Table I)

TABLE I: Constituents Inference Comparison

Method	Structure	Grammar
Equal Parts	$\frac{Mother\ eats}{X}$ biscuit $\frac{Mothereats}{X}$ cake.	S→X biscuit S→X Cake X→Mother eats
Unequal parts	Mother eats $\frac{Biscuit}{X}$ Mother eats $\frac{Cake}{X}$	S→Mother eats X X→ Biscut X→Cake

When unequal parts of sentences are taken to be constituents, these results in more compact grammars rather than when equal parts of sentences are taken to be constituents. In other words, the grammar is more compressed.

B. Overlapping Constituents

While extracting context-free grammar form unstructured web access log texts overlaps should never occur within tree structure. We use a proper data structure for selecting constituents when ever candidate constituents are selected for grammar generation, first it is checked in the constituents data structure if it already exist then it return same rule otherwise a new grammar rule is returned.

IV. Finding Pattern form Web Access Log Unstructured Text

A. Context-Free Grammar

We define an alphabet Σ as finite set of symbols. A string over an alphabet Σ is a finite ordered sequence of symbols from Σ. The length of the string α is the number of symbols in the string, with repetition and denoted by $|\alpha|$ (e.g. $|$aabbcc$|$=6). The empty string is denoted by ε, is the string of length zero. A CFG (context free Grammar) 'G' is a four tuple (N, Σ, P, S) where N is the set of non-terminals, Σ is the set of alphabets (also called as non terminals symbols P, P is set of production rules and $S \in N$ is the start symbol Conventionally A, B, …. Denotes non-terminals, a, b, Denote terminals, and $\alpha, \beta, \ldots$ represents strings in $(N \cup \Sigma)^*$

The production in CFG Si of the form $A \rightarrow \alpha$ A is called left-hand side (LHS), and α is the right hand side (RHS). We define A production to be a production with LHS as the non-terminal A. Given the production $A \rightarrow \alpha$, we say that $\beta\ \alpha\ \gamma$ is derived from $\beta A \gamma$ in one step and we denote it by $\beta A \gamma \Rightarrow \beta\ \alpha\ \gamma$. If δ is derived from γ in zero or more steps, the derivation is denoted as $\gamma \Rightarrow^* \delta$. The language of G, which is denoted by L (G), is the set of all terminal strings that can be derived from the start symbol S. Formally, $L(G) = \{X \in \sum^* | S \Rightarrow^* X\}$.

A sentence S of length $|S|$=n is a non-empty list of words $\{w_1, w_2, \ldots., w_n\}$. The words are considered elementary. A word w in sentences S is written as $S[i] = w_i$. Our algorithm learns the Grammar (CFG) from the set of sentences. These sentences are stored in a list called corpus. Note in our case the corpus is web access log file.

B. Corpus

A corpus of size $|S|=n$ is list of sentences $[S_1, S_2, \ldots, S_n]$.

C. Constituent

A constituent A constituent in sentences S is a tuple $C_S=\{b, e, n\}$ where $0 \le b \le e \le n$, b and e are indices in S denoting respectively the beginning and end of constituent, n is the non-terminal of constituent and is taken from the set of non-terminals. S may be replaced when its value is clear from the context.

D. Sub-Sentence or Word Group

A sub-sentence or word group of sentence S is a list of words $u^s_{i=j}$ such that $S = u + v^s_{i=j} + w$ (the + is defined to be the concatenation operator on lists), where u and w are lists of words and $u^s_{i=j}$ j with $i \le j$ is a list of j-i elements where for each k with $1 \le k \le j$-i : $v^s_{i=j}[k] = s[i+k]$ A sub-sentence may be empty (when $i=j$) or it may span the entire sentence (when $i=0$ and $j=|S|$). S may be omitted if its meaning is clear from the context.

E. Substitutability

A sub-sentence Sub-sentences u and v are substitutable for each other if

1) The sentences $S_1 = t + u + w$ and $S2 = t + v + w$ (with t and w sub-sentences) are both valid, and
2) For each k with $1 \le k \le |u|$ it holds that $u[k] \notin v$ and for each l with $1 \le l \le |v|$ it holds that $v[l] \notin u$.

Note that this definition of substitutability allows for the substitution of empty sub-sentences. We assume that for two sub-sentences to be substitutable, at least one of the two sub-sentences needs to be non-empty. For example consider following case

3) Mother eats biscuit.
   Mother eats cake.

In above case, the word biscuit and cake are the unequal parts of the sentences. These words are the only words that are substitutable according to definition. The word groups eats biscuit and cake are not substitutable, since the first condition in the definition does not hold (t = Mother in 3a and t = Mother eats in 3b) On the other hand, the word groups eats biscuit and eats cake are not substitutable, since these clash with the second condition. The word eats is present in both word groups. The advantage of this notion of substitutability is that the substitutable word groups can be found easily by searching for unequal parts of sentences.

V. Web Access Log Mining Algorithm

We split the problem of web access mining in to following phases:-

1) Data Cleaning: Before we can apply the algorithm we need to eliminate the irrelevant items form the server access log file so that the file contains a set of string that have only useful data for mining. Elimination of irrelevant items can be accomplished by checking the suffix of the URL name. For instance, all log entries with filename suffixes such as gif, jpeg, GIF, JPEG and map can be removed. For our analysis the username password are also removed. Data cleaning phase may be used to translate the data with context of information extraction.
2) Finding Constituent: The sentences are scanned and based on their alignment information among the sentences, the constituents (i.e. equal unequal part) are identified.
3) The resulting constituents are checked for overlapping and if no overlapping exists then a new rule is added in the result.
4) Multiple Production alternatives: If the occurrence of s is in such a position that multiple production alternatives are possible ($X \rightarrow$us$|$ws) then new production is $Y \rightarrow$ u$|$w and $X \rightarrow Y$'s.

A. Algorithm

Input: A corpus of flat sentences of access log (strings)
Output: Set of CFG rules R

Begin
Initialize rule set R=
While $\forall\ \alpha \in C\ |\alpha| > 1$ **do**
For $\forall\ \beta \in C$ and $\beta\ \alpha$ **do**
⟨⟨**{D}, {S}**⟩⟩ FindsutableSubsentences(α, β_i)
//⟨⟨**{D}, {S}**⟩⟩ are the set representing identical and distinct sub-sentences
For $\forall\ \gamma \in$ {D} **do**

//assign non-terminal to constituents that is distinct part of α and β.

If $|\alpha| > 1$ **then**

N= select next non- terminal for if no overlapping exist otherwise it will return same non-terminal present in *R*

$R=R\cup \{N\rightarrow\gamma\}$

// apply replacement rule for each string in the corpus

Update(C, $N\rightarrow\gamma$)

End if

End for

End for

End while

End

B. Constituents Selection

The alignment learning phase may generate unwanted overlapping constituents. Since we assume the underlying grammar for corpus is context-free grammar so we want to know the most appropriate disambiguated structure of the sentences of the corpus. We use two different approaches for the selection of constituents.

1) Assume the constituent learned first is correct. This means that when new constituent overlaps with older ones, they are ignored.
2) Constituents are selected based on their Support Factor. The algorithm compute support factor of constituents by counting the number of times the sequence of words in the constituent occurs. Normalize by total number of constituent in corpus C.

$$SF_{\gamma} = \frac{Count\ of\ \gamma\ in\ sentences \times length\ of\ \gamma}{Length\ of\ sentence}$$

where *N*= number of sentences in corpus and γ is the selected constituent.

123.456.78.9-- [25/Apr/2011:03:94:41-0580] "GET/A.html HTTP/1.0" 200 3290 "Mozilla/4.05 (Macintosh;)"
123.456.78.9-- [25/Apr2011:03:05:34 -0500]] "GET/B.html HTTP/1.0" 200 2050 A.html "Mozilla/4.05 (Macintosh;)"
123.456.78.9 -- [25/Apr:2011:03:05:39-0500] "GET/C.html HTTP/1.0" 200 4130 - "Mozilla/4.05 (Macintosh;)"
123.456.78.9-- [25/Apr2011:03:06:02 -0500] "GET/D.html HTTP/1.0" 200 5096 B.html "Mozilla/4.05 (Macintosh;)"
123.456.78.9 -- [25/Apr2011:03:10:45..0500] "GET/G.html HTTP/1.0" 200 9430 - "Mozilla/4.05 (Macintosh;)"
123.456.78.9 -- [25/Apr2011:03:12:23-0500] "GET/D.html HTTP/1.0" 200 7220 - "Mozilla/4.05 (Macintosh;)"
123.156.78.9 -- [25/Apr2011:03:07:55 -0500] "GET/R.html HTTP/1.0" 200 8140 L.html "Mozilla/4.05 (Macintosh;)"
123.156.78.9 -- [25/Apr2011:03:09:50 -0500] "GET/C.html HTTP/1.0" 200 1820 A.html "Mozilla/4.05 (Macintosh;)"
123.156.78.9 -- [25/Apr2011:3:10:02..0500] "GET/A.html HTTP/1.0" 200 2270 - "Mozilla/4.05 (Macintosh;)"
209.458.78.2 -- [25/,Apr2011:05:05:22-0500] "GET/A.html HTTP/1.0" 200 3290 - " Mozilla/4.05 (Macintosh;)"
209.458.78.3 -- [225/Apr2011:05:06:03- 0500] "GET/A.html HTTP/1.0" 200 1680 - "Mozilla/4.05 (Macintosh;)"

Fig. 1. Sample web server access log file

VI. Experimental Results

In this paper we have extracted context free grammar using alignment based learning approach of seven days web access log text file of http://www.dauniv.ac.in available in the formats of unstructured data source. Various analyses have been carried out to identify user behavior. For simplicity the time and size are ignored. A sample web access log file is shown as fallows.

A. Grammar Rule Extraction for User Identification

User identification means individual users by observing their IP address. To identify users, we propose some rules if there is new IP address then there is new user, if the IP address is the same but the operating system of browsing software are different we assume that different agent type therefore an IP address represents the different user. For each user, following grammar rules are extracted using alignment based learning the time of access has been ignored. (Table II)

1) Iterations

TABLE II: The Arrangement of Channels

Itration	Constituents	Grammar
I1	GET/A.html HTTP/1.0, GET/B.html HTTP/1.0, GET/C.html HTTP/1.0 ,……..	S→123.456.78.92[5/Apr/2011] X 200 Mozilla/4.05 Macintosh;) X→ GET/A.html HTTP/1.0 X→ GET/B.html HTTP/1.0 ……..
I2	Mozilla/4.05(Macintosh;) Mozilla/4.7[en]C-SYMPA (win95;U)	S→123.456.78.92[5/Apr/2011] X 200 Mozilla/4.05 Macintosh;) X→ GET/A.html HTTP/1.0 X→ GET/B.html HTTP/1.0 Z→ Mozilla/4.05(Macintosh;)
In	……	……..

B. Output

After applying the proposed algorithm the following set of grammar results:

S→123.156.78.9[25/Apr/2011]*XYZ*
S→209.158.78.2[25/Apr/2011]*XYZ*
S→209.158.78.3[25/Apr/2011]*XYZ*
X→GET/A.html HTTP/1.0, *X*→GET/B.html HTTP/1.0,.
Y→200, *Y*→305, …
Z→ Mozilla/4.05(Macintosh;), …

TABLE III: User Profile

Day	No.of Entries	No. of IP address	No of unique user	Failures
1	63567	7587	576	2931
2	61264	7632	613	2463
3	87565	19103	1766	1738
4	64536	8340	868	795
5	75535	12638	1039	1421
6	67342	22706	2033	2304
7	88233	32657	1765	2897

We interpret this as fallows: the start symbol represents individual user complete text. We can see that a text begins with fixed preamble; followed by a variable number of occurrences of *XYZ* (Page access) each represents a listing in server log file. Here the non-terminal *XYZ* represents to reference to the web page. The data fields for each listing can be extracted by mapping the text symbols to their actual

content. Then the domain specific heuristics can be used to identify the semantic meaning of different fields. In this web server analysis domain knowledge is not used in grammar generation. So this approach can be easily applied to other types of web access analysis. (Table III)

VII. Conclusion

We have demonstrated the use of alignment based grammatical inference to infer grammar rules (CFG) for web server access log file i.e. identification of grammatical rules from a given symbolic sample in language sentences. Our algorithm uses distinctions between sentences to find possible constituents during the alignment learning phase and selects the most probable constituents. For our experiment the dauniv.ac.in server log data sheet have been used. Our approach employs alignment similarities among the sentences to formulate grammar of the data sheet. The resulting grammar has been analyzed to identify the user behaviors on web server. The result of analysis is of great use for system administrator, web designer etc for their marketing planning, web personalization, etc.

References

[1] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web," in *Proc. IEEE Computer Society*, 1997, pp. 558.

[2] P. Palaga, L. Nguyen, U. Leser, and J. Hakenberg, "High-performance information extraction with alibaba," in *EDBT*, pp. 1140-1143, 2009.

[3] J. Allen, *Natural Language Understanding*, The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, USA. Second Edition, 1995.

[4] E. M. Gold, "Language identification in the limit," *Inf. Cont.* vol. 10, no. 5, pp. 447–474,1967.

[5] P. Dupont, L. Miclet, and E. Vidal, "What is the search space of the regular inference," in *Proc. ICGI Lecture Notes in Artificial Intelligence*, vol. 862, Heidelberg, Berlin: Springer, 1994, pp. 25–37.

[6] D. Angluin, "Queries and concept learning," *Mach. Learn*., vol. 2, no. 4, pp. 319–342, 1988.

[7] Y. Sakakibara, "Learning context-free grammars from structural data in polynomial time," *Theor. Comput. Sci*., vol. 76, pp. 223–242, 1990.

[8] D. Angluin, "Learning k-bounded context-free grammars," Yale Univ., New Haven, CT, *Yale Tech*. Rep. RR-557, 1987.

[9] Y. Sakakibara, "Learning context-free grammars using tabular representations," *Pattern Recognit.*, vol. 38, no. 9, pp. 1372–1383, 2005.

[10] Y. Sakakibara and M. Golea, "Simple recurrent networks as generalized hidden markov models with distributed representations," in *Proc. IEEE Int. Conf. Neural Netw, EEE Comput. Soc*, 1995, pp. 979–984.

[11] K. Kersting, L. D. Raedt, and T. Raiko, "Logical hidden Markov models," *J. Artif. Intell. Res*, vol. 25, pp. 425–456, 2006.

[12] N. S. Chaudhari, and Xiangrui Wang, "Language Structure Using Fuzzy Similarity," *IEEE Trans on fuzzy system*, vol. 17, no. 5, pp. 1011- 1024, Oct 2009.

[13] R. Thakur, S. Jain, and N. S. Chaudhari, "Incremental Discovery of Sequential Pattern from Semi-structured Document Using Grammatical Inference," in *Proc ICDCIT 2012, LNCS*, vol. 7154, Springer-Verlag Berlin Heidelberg, 2012 , pp. 269.

[14] P. Adriaans and M. Vervoort, "The EMILE 4.1 grammar induction toolbox," *Lecture Notes in Computer Science*, 2002, vol. 2484, pp. 293–295.

[15] V. R. Borkar, K. Deshmukh, and S. Sarawagi, "Automatically extracting structure from free text addresses," in *Bulletin of the IEEE Computer Society Technical committee on Data Engineering, IEEE,* 2000.

[16] J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan "Web usage mining: Discovery and Applications of usage patterns from web data," in *Proc. SIGKDD Explorations,* vol.1, no.2, Jan 2000, pp. 12-33.

[17] W. W. W. Consortium the Common Log File format. [Online]. Available: http://www.w3.org/Daemon/User/Config/

[18] M. V. Zaanen, "Implementing alignment-based learning," in *Proc. ICGI (Lecture Notes in Computer Science)*, 2002, vol. 2484, pp. 312–314.

[19] A Web server log file explained [Online]. Available: http://www.jafsoft.com/searchengines/log_sample.html.

Ramesh Kumar Thakur was born in Samistipur, Bihar, India, in 1974. He received the B.E. degree in Computer Science and Engineering, the M.E. degree in computer engineering, and currently perusing Ph.D. in computer engineering, from Devi Ahilya University, Indore, and Indore. In 1998, he was appointed as Asst. Prof at SVITS, Indore India. In 2001, he joined as a Asst. Prof in Department of computer Engineering at Devi Ahilya University, Indore, India, since 2007 he is working as Associate Prof at Devi Ahilya University, Indore He is involved in coordinating graduate-level and postgraduate-level training program in computer science for the university. During 2011 he was visiting professor at Indian Institute of Technology, Indore, India. He is a member of IEEE. He has published many research papers in various national and international journals & participated in so many conferences. His area of research is Information Extraction.

Suresh Jain is a director and professor in computer engineering at Sushila Devi Bansal College of Engineering, Indore. He completed his Bachelor of Engineering from MANIT Bhopal, Master of Engineering from SGSITS Indore and Ph.D. in Computer Science from DAVV. He has experience of over 25 years in the field of academics and research. He served Devi Ahilya University over 20 years in the capacity of Lecture, Reader and Professor of Computer Engineering. He has published more than 80 research papers in reputed Journals and Conferences. He teaches courses on Artificial Intelligence, Computer Graphics, Theory of Computation, and DBMS. He is a life member of CSI, IEEE and ISTE. He is guiding research in the field of machine learning, web mining and information retrieval.

Narendra S. Chaudhari has completed his undergraduate, graduate, and doctoral studies at Indian Institute of Technology (IIT), Mumbai, India, in 1981, 1983, and 1988 respectively. Dr. Narendra S. Chaudhari has shouldered many senior level administrative positions in universities in India as well as abroad. A few notable assignments include: Dean - Faculty of Engineering Sciences, Devi Ahilya University, Indore, Member - Executive Council, Devi Ahilya University, Indore, Coordinator - International Exchange Program, Nanyang Technological University, Singapore, Deputy Director - GameLAB, Nanyang Technological University, Singapore. Currently, he is Dean - Research and Development, Indian Institute of Technology (IIT) Indore, and Member - Advisory Board, ITM University, Gwalior (M.P.).

Narendra has done significant research work on game AI, novel neural network models like binary neural nets and bidirectional nets, context free grammar parsing, and graph isomorphism problem. He has supervised more than 18 doctoral students and more than 80 Master's students. He has delivered invited talks and presented his research results in several countries like America, Australia, Canada, Germany, Hungary, Japan, United Kingdom, etc. A few institutes where he has given talks on his research includes Massachusetts Institute of Technology (MIT) USA, Nagoya Institute of Technology (NIT), Nagoya, Japan, Manchester Metropolitan University (MMU), Manchester (U.K.), Beijing Normal University (BNU), Beijing (P.R. China), etc. He has more than 240 publications in top quality international conferences and journals.He has been invited as a keynote speaker in many conferences in the areas of Soft-Computing, Game-AI, and Data Management. He has been referee and reviewer for a number of premier conferences and journals including IEEE Transactions, Neurocomputing, etc. Dr. Chaudhari is Fellow of the Institution of Engineers, India (IE- India), as well as Fellow of the Institution of Electronics and Telecommunication Engineers (IETE) (India), Senior member of Computer Society of India, Senior Member of IEEE, USA, Member of Indian Mathematical Society (IMS), Member of Cryptology Research Society of India (CRSI), and many other professional societies.