Heuristic Computational Matrix Method for Marathi Grammar Checker

Nivedita S. Bhirud, R.P.Bhavsar, B.V.Pawar

Abstract: Spelling, morphology, syntax and semantics are the important areas of Natural Language (NL) sentence analysis. Syntax checking of a sentence is broadly referred as a 'grammar checking', however it also involves morphological analysis hence technically it is a multidimensional problem. Syntax of a natural language defines permissible sentence structures and constraints on constituents such as their order and unification constraints. It is a purely theoretical aspect and considered as computationally trivial rule enforcement problem. Rule formulation needs expert labour work and is costly and time consuming affair. Modern data driven language engineering approach advocates use of minimal knowledge base (linguistic information) and relies on knowledge extraction from tagged data. It is difficult to find such tagged data for non-English natural languages like Marathi (Indian Language). Considering these facts for grammar checking problem, we have come up with intuitional heuristic method for Marathi grammar checking which uses basic syntactic cues and minimal lexical information. We have modeled this heuristic method scientifically using basic matrix comparison operation. Our approach relies on syntactic cues like word ending, verb ending. We have tested our method on handcrafted Marathi sentences catering different Marathi sentence structures (one hundred and fifty three). The performance is measured using precision and recall metrics. The system has yielded 83% precision and 93% recall on sample data. This approach can be exploited for well structured text documents typically in the closed domains like legal, official, educational etc.

Keywords : Computational Linguistics, Heuristic Function, Marathi Language Grammar, Natural Language Processing, Rule based approach, Statistical approach

I. INTRODUCTION

Information retrieval, summarization, grammar checker, spell checkers, QA system, machine translation, text-speech, and speech-text conversion, etc. are some prominent applications stated under NLP domain. Grammar checking is the most used application and has become attracting research area for researchers. The objective of a grammar checker tool been observed that it require intensive lexical resources. Bhirud and et.al. [8] analyzed grammar checkers of foreign and Indian languages w.r.t. approaches, methodologies as well as features such as grammar errors, weakness and evaluation and found that there is scope to develop grammar checker for the Marathi language.

The proposed work focuses on the development of Marathi grammar checker. Marathi is a morphologically rich language and hence requires intensive lexical resources to develop Marathi grammar checker application. Along with objective of proposed system i.e. suggesting and correcting grammatical errors in Marathi sentences, one of the challenging objectives of proposed system is to reduce requirement of intensive lexical resources that can be achieved by proposed heuristic computational matrix method. Computational matrix method makes use of postpositions primarily to check syntactic and shallow semantic correctness of a sentence.

The rest of the paper is organized as follows: Section II brief at related work. Section III explains core concepts used in proposed system. Section IV and V outline proposed computational heuristic method. Section VI discusses result analysis. The summary and conclusion are listed in section VII.

II. RELATED WORK

This section explains the general algorithm and approaches for developing grammar checker application and its analysis.

A grammar checker takes input in form of a sentence and input sentence has to undergo some preprocessing stages such as sentence tokenization, morphological analysis, and parts of speech tagging [1]. Grammar checking of a preprocessed sentence involves syntactic parsing using chosen methods. Broadly rule-based, data-driven, and hybrid grammar checker methods are used for developing grammar checkers of worldwide languages. In a rule-based method, the text is checked against hand-crafted rules and it is a most common method [9]. Data-driven method has two sub methods, namely, corpus-based and probabilistic/statistical method [14]. The input text is checked against corpus, which is supposed to be a complete document of a language representing all language features under corpus-based method. In probabilistic/statistical checking method, an annotated corpus is used. If correctly occurring sequence observed then it is declared as the correct sentence and uncommon sequence lead to an error [2]. Hybrid method combine both rule-based and data-driven methods [12].

After the study of various grammar checkers for world-wide languages, Bhirud et.al.[8] analyzed some finding based on performance evaluation of grammar checkers developed using the above mentioned approaches. It has been observed that studied grammar checkers gives prominent results, however,

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author

Nivedita S. Bhirud*, Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, India. Email: nivedita.bhirud@viit.ac.in

R.P. Bhavsar, School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, India. Email: rpbhavsar@nmu.ac.in

B.V. Pawar, School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, India. Email: bvpawar@nmu.ac.in

requirement of expertise and extensive labor for rule management and availability of relevant good corpus are disadvantages of rule based and data driven method respectively [18]. Finally, reducing the requirement of such extensive lexical resources can lead to give more promising results.

III. BACKGROUND

The foundation of the proposed method is based upon karaka relation which h describes the theory behind sentence analysis. This section describes the karaka relation followed by data structures used in the system.

A. Karaka Relation

The proposed approach is inspired by Computational Paninian Grammar framework [3]. Many NLP tools of modern Indian languages have been developed using this framework and most suitable for free word order languages. Paninian framework is also known as 'karaka theory' and due to its features; it is more suitable to Marathi.

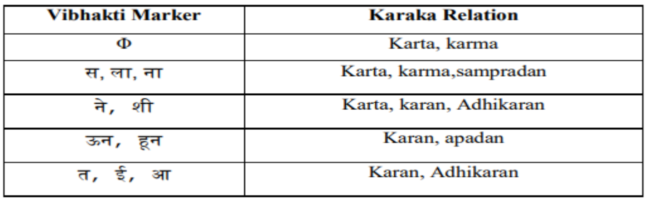
A sentence is composed of words to which parts of speech is assigned. In Marathi, there are 8 types of parts of speech [5] viz. noun (नाम), pronoun (सर्वनाम), adjective (विशेषण), verb (क्रियापद), adverb (क्रियाविशेषण), conjunction (उभयान्वयी अव्यय), postposition (शब्दयोगी अव्यय) and interjection (केवलप्रयोगी अव्यय), play vital role in valid sentence construction at core level.

Words have semantic relations with each other in a sentence, and such semantic relations are called as 'karaka' relation. These karaka relations can be identified from syntactic cues provided by postposition markers and these postposition markers are 'vibhakti pratyaya' (विभक्ति प्रत्यय). In Marathi, generally vibhakti pratyayas are attached to nouns or pronouns [7] whereas postpositions attached to verbs are called as TAM (Tense, Aspect, Mood) label [4]. Vibhakti pratyaya have one to many relations with karaka i.e., one vibhakti pratyaya can imply more than one karaka which provide syntactico-semantic information.

In Marathi, there are 6 karaka relations namely: karta(कर्ता), karma(कर्म), karan(करण), sampradan(संप्रदान), apadan(अपादान), adhikaran(अधिकरण). Table I shows a couple of examples of mapping between vibhakti pratyaya and karaka relations (relation w.r.t. verbs).

Illustration: In the sentence, 'रामने आंबा खाल्ला', word रामने has 'ने' vibhakti marker, and according to table I, word रामने is assigned with karta, karan and adhikaran karaka relations w.r.t. verb. However, 'karta' karaka relation is more appropriate w.r.t. verb खाल्ला. Whereas in the sentence, 'राम चाकूने फळ कापतो', word चाकूने has 'karan' karaka relation w.r.t. verb कापतो. Though same vibhakti marker 'ने' is attached to a different word in both sentences, a relation of that word with verb i.e. semantic role changes.

Table I: Mapping between vibhakti markers and karaka relation



Vibhakti Marker	Karaka Relation
Φ	Karta, karma
स, ला, ना	Karta, karma,sampradan
ने, शी	Karta, karan, Adhikaran
ऊन, हून	Karan, apadan
त, ई, आ	Karan, Adhikaran

B. Data Structure

Words with postpositions and suffixes are stored into a data structure called 'open set' whereas other remaining words are considered into 'closed set'.

Let U is a universal set of all the words under study, A is closed set of words and B is an open set of words which can be called as a complement of A .

Mathematically it can be represented as:

$$B = U \setminus A; \{x \in U \,|x \in A\}/$$

Open set contains infinite words as any word with postpositions can be member of it and closed set is finite as it is set of stored words.

IV. PROPOSED APPROACH

This section will describe the proposed method to check grammaticality of Marathi sentences, where minimal lexical resources are required. Initially, details of dataset explaining types of sentences considered for testing of the system is given followed by explanation of pre-processing steps such as sentence extraction, tokenization, morphological analysis, and parts of speech tagging. Further, word group formation and its validation are explained. Along with the validation of words within a group, there is a need to check the validation of inter-group words, which is explained in section 4.D. Proposed computational matrix method which checks grammaticality at the sentence level is described with the illustration of the system.

A. Dataset

Simple handcrafted sentences of Marathi are considered as the dataset. We have used handcrafted simple sentences to cover all structures of Marathi sentences which make sentence grammatically fit. A simple sentence consists of a single clause, where only a single subject and predicate is involved.

Simple sentences are broadly categorized into copular, declarative and modal sentences. In copular sentences, copular verbs are involved in sentence construction, declarative sentence states a fact and modal auxiliary verbs are used in modal sentences. मुलगा हुशार आहे, आपण काम करु are an example of copular and modal sentences respectively.

Declarative sentences further can be categorized into:

Transitive: transitive verbs are involved such as खा, पी, धू

Intransitive: intransitive verbs are involved such as झोप, पळ, नाच

Ditransitive: ditransitive verbs are involved such as दे, शिकव, सांग

Casual: transformation from intransitive to transitive e.g. हसवले

Impersonal: involves verb that do not require a subject e.g. उजाडले, सांजावले, ढगाळले

Dative: involves verb which show physical or psychological notion such as आवड, दिस, पट

Passive: verb agrees with an object rather a subject.
For experimental purpose, we have considered sentences as given in table II. While considering these sentences, we also considered different categories of verbs stated in table III. Verb inflects for grammatical feature such as gender, number and person of subject or direct object or sometimes verb remain in their unmarked form. While inflection, the verb ending plays vital role as inflectional form depends on verb ending whether consonant ending or vowel ending.

Table II: Dataset

Types		Verb Count	Sentence Count
Copular Sentence		2	30
Declarative Sentence	Intransitive	15	50
	Transitive	15	60
	Ditransitive	12	60
	Casuative	12	70
	Impersonal	15	70
	Dative	15	50
	Passive	20	60
Modal Sentence		15	50
Total		119	500

Table III. Verb Category

Category		No. of verbs
Consonant ending	अ-कारान्त	100
Vowel ending	आ-कारान्त	04
	ई–कारान्त	04
	ऊ–कारान्त	01
	ए-कारान्त	08
	ओ-कारान्त	02

B. Pre-processing

Input is in the form of a document. The first step under pre-processing is sentence extraction using the appropriate symbol (full stop) [6]. An extracted sentence is further tokenized and then tokens are morphologically analysed. The objective of morphological analysis is the detection of vibhakti pratyaya and TAM label. Root words are identified after removal of postpositions and checked against root verb database or closed set and vibhakti markers are checked against an open set. Parts of speech can be assigned to word using a result of morphological analysis. Tagged words then send to next step of word grouping.

C. Word-Grouping

In Marathi sentence, a basic unit word may belong to a noun group [16] or verb group. Each word in a group is related to each other by grammatical rules. Each group has a head which has grammatical relation with the head of other groups. E.g. (मधुच्या भावाने) (बबनला) (कोरी वही) (दिली होती), in this sentence group is indicated by brackets and head of a group is shown by underlined word. (दिली होती) is verb group and each noun group head is agreed with a verb group head by agreement rules. The rule set required for word grouping validation is inspired from [18] and [19].

D. Mapping

After preparation and checking the validity of noun group and verb group, provision of the optionality of karakas for root verb and assignment of semantic roles to noun head is done using karaka-verb mapping and karaka transformation rules respectively. Vibhakti markers and TAM labels are important elements of mapping.

- **Verb-Karaka Mapping**

Verb-Karaka mapping specifies karaka permitted for verb root. Mandatory presence of karaka is indicated by ‘1’, optional presence of karaka is indicated by ‘0’ and not permitted karaka is indicated by ‘*’. Table IV represent verb-karaka mapping where root verb ‘खा’ is transitive (karma is mandatory and is indicated by ‘1’), root verb ‘झोप’ is intransitive (karma is not permitted and hence indicated by ‘*’).

Verb classes are formed on the basis of TAM label and verb classfication and these classes are assigned to root verbs. Root verb and verb class have one to many relationship.

- **Karaka Transformation Rules**

Once an appropriate verb-karaka mapping is completed, the next task is the application of karaka transformation rules using verb class and karaka-vibhakti transformation rule along with inter-group (noun group-verb group) validation checking.

Table IV. Verb-Karaka mapping

Root verb	Verb-Karaka mapping					
	Karta	Karma	Karan	Sampradan	Apadan	Adhikaran
खा	1	1	0	0	0	0
झोप	1	*	0	*	0	0

Transformation rules give mapping for TAM label of verb class. It specifies vibhakti markers permitted for applicable karaka relation. Example: Consider verb class of TAM label ‘तो’. Vibhakti markers applicable for karaka relation of class ‘तो’ are as in table V. Noun group and verb group validation checked using grammatical features Gender, Number, Person (GNP) of noun group head with Tense Aspect and Mood (TAM) label of verb group head (syntactic cue).

V. COMPUTATIONAL MATRIX METOD

Grammatical checking at a sentence level can be completed using proposed a heuristic method, a computational matrix method. Proposed matrix has words/noun group head as rows and their karaka relation as columns. It checks syntactic as well as shallow semantic correctness of sentence.

Retrieval Number: C8581019320/2020©BEIESP
DOI: 10.35940/ijitee.C8581.019320
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Le $N = n_i\, ; i = 1,2, \ldots , n$, where '$n$' is noun groups' head and $K = k_1, k_2, k_3, k_4, k_5, k_6$ where 'k' represents karaka relation explained in section III.A. Let $\{C_{i,j}\}_{N \times K}$ – resulting computational matrix where $C_{i,j}$ is the value from verb-karaka mapping.

1. Scan all rows of $\{C_{i,j}\}_{N \times K}$, if single '1' or '0' found assign respective karaka to noun head. *//to allocate single karaka to word/head of group*
2. Scan all columns of $\{C_{i,j}\}_{N \times K}$, if single '1' or '0' found assign respective karaka to noun head. */to allocate single karaka to word/head of group*
3. If single '1' or '0' not found after scanning all rows and columns, scan rows again till all karaka assignment to all n_i
    - a. If a row has '1' and '0', assign karaka with value '1' to n_i *//priority set to '1'*
    - b. Else if a row has '1' and '1', assign initial karaka to n_i *//priority set to initial karaka*
    - c. Else if a row has '0' and '0', assign initial karaka to n_i *//priority set to initial karaka*

   End if
4. If karaka not assigned to all n_i , suggest an error. Else if declare sentence as "Grammatically correct".

Illustration:

Consider Marathi sentence, "चंदू शाळेत रमाचा डबा खातो".

Table V: Transformation rules for a class with TAM label 'तो'

Vibhakti Marker	Karaka Relation
Null	karta, karma
स, ला, ना	karta, karma, sampradan
ने, शी	karta, karan, Adhikaran
ऊन, हन	karan, apadan
त, ई, आ	Adhikaran

Using the proposed system, steps to check grammaticality of the sentence are as follows:

Tokenization: (चंदू) (शाळेत) (रमाचा) (डबा) (खातो)

Morphological Analysis: (चंदू) (शाळेत) (रमाचा) (डबा) (खातो)

Parts of Speech Tagging: (चंदू Noun) (शाळेत Noun) (रमाचा Adjective) (डबा Noun) (खातो Verb)

Word Grouping: (चंदू) (शाळेत) (रमाचा डबा) (खातो). In word group, (रमाचा डबा), डबा will play the role of a group head.

Verb-Karaka Mapping: Root verb 'खा' is obtained after pre-processing steps. To get optionality of karaka relation of root verb 'खा' refer Table IV. From TAM label 'तो' of verb 'खा', the respective class is assigned and permitted vibhakti markers are fetched. Table V gives vibhakti markers for a class with TAM label 'तो' and we get following karaka relations for each word and group head, and karaka relations are assigned as follows:

Word/Group Head	**Karaka Relation**
चंदू	karta, karma
शाळेत	Adhikaran
डबा	karta, karma

Computational Matrix method: Initially, computational matrix formed as follows.

	कर्ता	कर्म	अधिकरण
चंदू	1	1	-
शाळेत	-	-	0
डबा	1	1	-

By applying algorithm depicted in section V, resultant computational matrix will formed as:

	कर्ता	कर्म	अधिकरण
चंदू	1	-	-
शाळेत	-	-	0
डबा	-	1	-

We get karaka relation to each word/ group head and can conclude that the sentence is grammatically correct.

VI. RESULT ANALYSIS

Dataset considered for the proposed method is discussed in section IV.A. So far, we have tested the proposed method for simple Marathi sentences. As per the description in section IV.A, total 500 simple sentences are taken into consideration which is formed using 119 types of verbs of different categorization (table III) consisting 400 grammatically correct sentences and 100 grammatically incorrect sentences verified by a linguist. A document consisting of 500 simple sentences feed to the system as an input. The accuracy of the system needs to be measured using metrics such as '*Precision'* and '*Recall'*. For our proposed approach, both can be calculated using the following formulae.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Where,

$\boldsymbol{True\ Positive}$
$= proposed\ method\ detect\ actaul\ grammatically\ correct\ sentences\ as\ a\ correct$
$\boldsymbol{False\ Positive}$
$= proposed\ method\ detect\ actual\ grammatically\ incorrect\ sentences\ as\ an\ \ incorrect$
$\boldsymbol{False\ Negative}$
$= proposed\ method\ detect\ actual\ grammatically\ correct\ senetnces\ as\ an\ incorrect$

Document tested on the proposed system and results were analysed. We have tested results for all types of sentences mentioned in table II and results are depicted in table VI.

The overall system then evaluated using evaluation metrics *precision* and *recall*, mentioned in table VII.

Results for the proposed approach are promising. We investigated outcome of proposed method and found prominent reasons for short falling. These reasons are as a) number of karaka and noun head mismatch, b) agreement failure within a group and c) agreement failure between groups. Error analysis is shown in fig. 1.

Overlapping of vibhakti markers with noun last word, overlapping of rules, and rule mismatch are found to be some minor reasons.

Table VI. Result Analysis of various types of Simple sentences

Sentence Type	Data set	*Precision* in %	*Recall* in %
Copular	30	85	95
Transitive	50	86	93
Intransitive	60	81	91
Ditransitive	60	79	97
Causative	70	79	89
Impersonal	70	87	89
Dative	50	81	91
Passive	60	83	96
Modal	50	85	95

Table VII. Overall Result Analysis

Sentence Type	Data set	Precision	Recall
Simple sentences	500	83%	93%

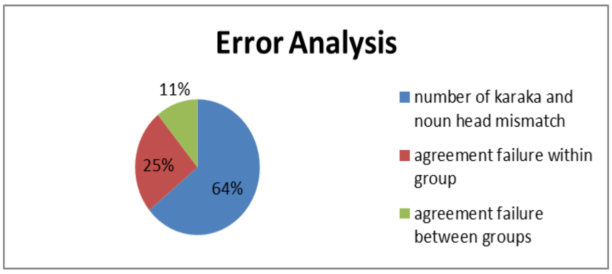


Fig. 1 Error Analysis

VII. CONCLUSION

Proposed computational matrix method uses minimal linguistic heuristic method for checking syntactical and shallow semantical correctness of Marathi simple sentences which make use of postpositions of words prominently rather applying the general algorithm of grammar checker which require heavy lexical resources. Proposed system tested on real-world handcrafted Marathi sentences and it has been observed that system detect the error at a various level ranging from word level to sentence level. Proposed method covers checking of Marathi sentence structures which is base of grammaticality of a sentence. A heuristic method is a key processing step under system suggests syntactical as well as shallow semantic errors in a sentence with the help of matrix comparison type operations. The performance is evaluated based on *precision* and *recall* metrics, giving *precision* 83% and *recall* 77% for simple sentences respectively. Typically using proposed method, documents of closed domain such as legal, educational, official, medical etc. can be checked grammatically using minimal lexical resources. Due to its lightweight nature can be easily ported to handheld device platforms. In the conclusion, proposed method is capable of checking grammaticality of simple Marathi sentences, using minimal lexical information with coverage of most of grammatical errors.

REFERENCES

1. Misha Mittal, Dinesh Kumar, Sanjeev Kumar Sharma, “Grammar Checker for Asian Languages: A Survey”, International Journal of Computer Applications & Information Technology Vol. 9, Issue I, 2016
2. Blossom Manchanda, Vijay Anant Athvale, Sanjeev Kumar Sharma, “Various Techniques used for Grammar Checking”, Internation Journal of Computer Application & information Technology, Vol. 9, Issue 1
3. Bharati, A., Chaitanya, V., & Sangal, R. (1995). *Natural Language Processing: A Paninian Perspective*. New Delhi: Prentice Hall of India
4. Anil Kumar Singh, Samar Husain, Harshit Surana, Jagadeesh Gorla, Chinnappa Guggilla, Dipti Misra, “Disambuguating Tense, Aspect and Modality Markers for Correcting Machine Translation Errors”, Hyderabad : s.n., 2007. IIIT/TR/2007/75.
5. Damle, M.K. *शास्त्रीय मराठी व्याकरण* (Shastriya Marathi Vyakran). Pune : s.n., 1965.
6. Wanjari N, Dhopavkar G M and Zungre N B. (2016) *Sentence Boundary Detection For Marathi Language*, Procedia Comput Sci. 78: 550–555.
7. Hiremath, R.G. “Marathi Vyakaran Parichay”. Pune : Mehata Publishing House, 1993.

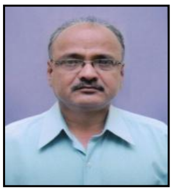
- Nivedita Bhirud, R.P. Bhavsar, B.V. Pawar, “Grammar Checker for Natural Languages: A Review”, Chennai : AIRCC, 2017, Vol. 6, pp. 10-18. 2278-1307.
- Tesfaye, Debela, “A rule-based Afan Oromo Grammar Checker”, Jimma, Ethiopia : s.n., 2011, Vol. 2.

8. Mandeep Singh, Gurpreet Singh, Shiv Sharma “A Punjabi Grammar Checker”, Punjab University : s.n., 2008.
9. Bal Krishna Bal, Prajol Shrestha, “Architectural and System Design of the Nepali Grammar Checker”, www.panl10n.net/english/.../Nepal/Microsoft%20Word%20-%208_OK_N_400.pdf
10. “A prototype of a grammar checker for Icelandic”, available at www.ru.is/~hrafn/students/BScThesis_Prototype_Icelandic_Grammar Checker.pdf
11. H. Kabir, S. Nayyer, J. Zaman, and S. Hussain, “Two Pass Parsing Implementation for an Urdu Grammar Checker”.
12. Jahangir Md; Uzzaman, Naushad; Khan, Mumit. “N-Gram Based Statistical Grammar Checker For Bangla And English”. Center for Research On Bangla Language Processing. Bangladesh, 2006
13. Vaishali B. Patil, B.V. Pawar, “Developing Links of Compound Sentences for Parsing Through Marathi Link Grammar Parser”,International Journal on Natural Language Computing, Vol. 03, No. 5/6 , dec. 2014
14. Vaishali B. Patil, B.V. Pawar, ,” Developing Links for Marathi Noun Phrase Morphology for Proposed Link Grammar Parser”,KJCS.Vol. 06. Issue 06.Oct. 2012 . PP 291-299
15. R.P.Bhavsar, B.V. Pawar, “Rule Based Word Morphology Generation Framework”,International Journal of Computer Science. Vol. 08. Issue 03. No. 02. May 2011
16. B.V.Pawar & N.S. Chaudhari, “Marathi Language Grammar Parsing using Tomita’s Approach”, Indian Linguistics, Journal of the Linguistic Society of India. Vol. 61, Numbers 1-4, 2000, PP 69-96.
17. B.V.Pawar, “Comparison of Results of Tomita’s Algorithm and LA Algorithm for parsing Marathi Sentences.”, Indian Linguistics, Journal of the Linguistic Society of India. Vol. 65, Numbers 1-4, 2004, PP 133 -140.

AUTHORS PROFILE

Mrs. Nivedita S. Bhirud finished her B.E. (Computer Engineering) from Pune University (2010), followed by M.E. (Computer Engineering) from Mumbai University (2013). She has 7 years of teaching experience in the field of Computer Engineering. Currently, she is working as Assistant Professor at Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune. Her research interest include NLP, Machine Learning. She has published over 10 journal papers and 5 conferene papers. She is life time member of International Association of Engineers (IAENG) e-mail: nivedita.bhirud@viit.ac.in

Dr. R. P. Bhavsar obtained his B.C.S. degree from Pune University (1992) followed by MCA in (1995) from Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. He was awarded Ph.D. in Computer Science in 2016 from the same university. He has varied experience of 23 years which includes 03 years as Scientist at C-DAC (Pune), 07 years as System Analyst, 13 years of teaching. Currently, he is working as Associate Professor in School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. His research area includes Machine Translation, Applied Natural Language Processing, Capacity Building, Machine Learning, IT & Networking infrastructure setup and has handled software development projects. Presently, 3 research scholars are pursuing their Ph. D. under his supervision. He is life time member of Computer Society of India (CSI)
e-mail: rpbhavsar@nmu.ac.in



Prof. B. V. Pawar finished his B. E. (Production Engineering) from reputed VJTI Mumbai (1986), followed by M.Sc. (Computer science) from Mumbai University (1988). He was awarded Ph.D. degree in Computer Science by Kavayitri Bahinabai Chaudhari North Maharashtra University (formally North Maharashtra University), Jalgaon, India. He has huge experience of 30 years teaching and research in the field of computer science. Presently, he is working as Professor at School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. His research interests include NLP, Web computing, Information Retrieval and Machine Translation, Machine Learning. He has guided 12 research Scholars for their doctoral research. He has published over 75 research /journal papers and 113 conference papers presentations and 05 invited talks. He is life time member of Computer Society of India (CSI), Linguistic Society of India (LSI) and International Association of Engineers (IAENG)
e-mail: bvpawar@nmu.ac.in