https://doi.org/10.26438/ijcse/v6si1.1925

IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-6, Special Issue-1, Feb 2018 E-ISSN: 2347-2693

# Factored Language Modeling

**A.R. Babhulgaonkar[1*], S.P. Sonavane[2]**

[1*]Department of Computer Science & Engineering, Walchand College of Engineering, Sangli, India
[2] Department of Information Technology, Walchand College of Engineering, Sangli, India

[*]*Corresponding Author:* arbabhulgaonkar@dbatu.ac.in, *Tel.: +91-9511701854*

**Available online at: www.ijcseonline.org**

***Abstract***— Language modeling is a technique for finding the next most probable word in a sentence. It is first and essential task for successful implementation of some natural language processing applications like machine translation and speech recognition. It ensures for correctness and fluency of the target output in these applications. N-gram is a traditional way to implement language model in which only previous words in the sentence are used to predict the probable next word in the sentence. Factored language modeling is a method to utilize linguistic knowledge of the word along with the word itself for constructing the language model. The paper describes the factored language modeling technique and compares the results obtained against the traditional n-gram technique using perplexity as a measure.

***Keywords***— Language model, Perplexity, Factored language model, Backoff.

## I. INTRODUCTION

Machine translation still produces poor output for highly inflectional languages as compared to less inflected languages. The reason may be rich morphology of these languages. Due to many morphological variations larger vocabulary is required for translation. The larger vocabulary reduces the out of vocabulary words problem but at the same time increases the size of language models and training corpora. A technique used to deal with morphologically rich languages is the use of grammatical and lexical information of these languages in some form. The factored language modeling provides an approach to incorporate the grammatical and lexical information of the language in language model.

A language model is a formalism that shows which words are more or less likely to be generated during some conversation in any natural language. This intuition is also used to predict what the next word would be, given the history of words appeared so far. The language model assigns probability to each possible next word and selects the highest one. It is also used to assign a probability to an entire sentence. If W = $w_1, w_2, \ldots w_n$ is a sentence then language models assign a probability estimate $p(W)$ to this sentence subject to $p(W) = 1$. It provides guidance and restricts the search of alternative word hypotheses during recognition. A language model for Hindi could suggest that 'राम खाता है' is more correct than 'राम खाती है' from grammatical point of view after considering gender knowledge of noun 'राम'. In short, Model that calculates probability of sequence of words in a sentence is called as language model. Apart from on words, it can also be constructed on characters, signs or symbols which form sequences.

Remaining part of the paper is structured as follows: Section II explains the techniques of language modeling, Section III contains the related work of factored language modeling, Section IV gives the methodology used for implementation, Section V provides a discussion on results obtained through experimentation and Section VI concludes research work and also gives directions for future work.

## II. TECHNIQUES OF LANGUAGE MODELING

### Standard n-gram language modeling

The leading method of language modeling is n-gram language modeling. Here, n is degree of language model. N-gram language models are based on statistics of how likely words are following each other. It shows that, how many previous words along with the next word are considered for predicting the next word in a sentence. Ideally n-gram models use the previous n − 1 words to represent the history $h$ of the next word to be predicted. Thus, probability of next word $w_n$ depends on history $h$ of previous words $w_1$, .. $w_{n-2}$, $w_{n-1}$ in sentence.

$$p(w_n \mid h) = p(w_n \mid w_1, ..., w_{n-2}, w_{n-1}) \quad (1)$$

For example, Suppose the history *h* is "राम आम खाता " and then the probability that the next word is है can be represented by using conditional probability as:

$$p(\text{है} \mid \text{राम आम खाता}) \quad (2)$$

One way to calculate this probability is by using relative frequency counts of the sentence. Count how many times राम आम खाता is there in corpus and how many times out of that it is followed by word है in corpus.

$$p(\text{है} \mid \text{राम आम खाता}) = \frac{C(\text{राम आम खाता है})}{C(\text{राम आम खाता})} \quad (3)$$

Using the large enough corpus like web, the counts of this sequence of specific words can be obtained and conditional probability (equation 3) can be estimated. This way of finding the probability distribution of a sequence in the given corpus is called as maximum likelihood estimation (MLE). In this method, the counts of sequence of specified words in the corpus are obtained and then normalized it to limit them between 0 to 1 range of probability. In generalised form this can be represented as:

$$p(w_n \mid w_1, ..., w_{n-2}, w_{n-1}) = \frac{C(w_1,....w_{n-2},w_{n-1}\ w_n)}{C(w_1,.....,w_{n-2},w_{n-1})} \quad (4)$$

In equation 4, the N-gram probability is estimated by dividing the total observed frequency count of a particular sequence of words in the corpus by the total observed frequency count of the given prefix of words for next word.

**Factored Language Model (FLM)**

Factored language model treats each word as a collection of K factors. So every word in the sentence can be represented as a set of k features. So $w_t = \{f_{t1}, f_{t2}, \ldots, f_{tk}\}$. Factor is some additional information about word and it can be anything such as morphological class of word, stem, root, part of speech tag of the word or any other linguistic information that can be associated with the word. Each word itself is also considered as a factor. For highly inflected languages, it will be more useful as more and more factors of a word can be obtained in it. The factored representation is applicable to any language as semantic features available in that language can be considered as factors. Thus, FLM is a probabilistic language model over both words and its associated factors, as shown in equation 5. The FLM utilises these factors of the word to produce a statistical model over the individual factors.

$$p(f_{1:t}^{1:k}) \quad (5)$$

Using n-gram like representation it can be represented as:

$$p\ (\ f_t^{1:k} \mid f_{t-n+1:t-1}^{1:k}) \quad (6)$$

This says that, probability of any factor can be obtained by using all factors available in the history of the word in the context. In this representation, FLM opens up many possibilities for modeling options in addition to standard n-gram model over words.

After applying the chain rule of probability this joint distribution becomes:

$$\prod_k p(\ f_t^k \mid f_t^{1:k-1}\ ,\ f_{t-n+1:t-1}^{1:k}) \quad (7)$$

Where t represents any word in history and k represents any factor of that word in the history. Equation 7 shows a single possible ordering among many orderings of the factors by chain rule. Apart from this, any order of factors can be selected and any set of factors can be used for calculation of the probability of the next factor. The possible number of factor permutations are K! and the number of subsets of factors are $2^{nK}$, hence the FLM can represent large number of statistical language models which is equal to K! $2^{nK}$. The standard n-gram is the simplest case of the FLM in which only one factor such as a word itself is considered. Same is the case with class-based language model, where class is one factor and the class of the word depends only on the previous class.

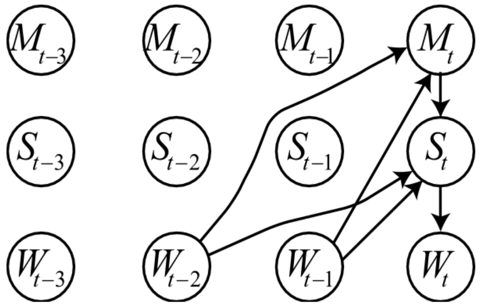

Figure 1. Graphical representation of factored language model.

The FLM can be shown as graphical model over a set of factors as shown in figure 1. The graphical model given in figure 1 shows following conditional probability model:

$$p(W_t \mid S_t , M_t )\ p(S_t \mid M_t , W_{t-1} , W_{t-2} )\ p(M_t \mid W_{t-1} , W_{t-2})$$

(8)

Where, $W_t$ is word factor, $M_t$ is the word's morphological tag factor and $S_t$ is the word's stem factor. This is showing a factored class-based language model, where the word is represented by using three factors such as word itself $W_t$, a stem $S_t$ and morphological tag $M_t$.

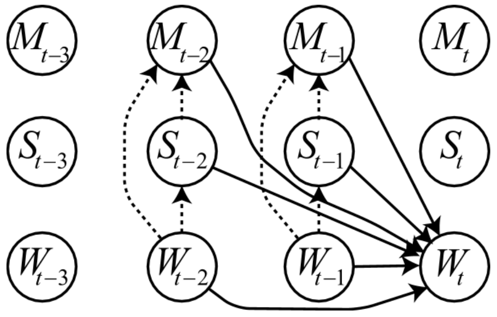

Figure 2. A factored language model for a word with word, stem and morphological tag as parents.

The FLM shown in figure 2 is graphical form of the following model:

$$p(W_t \mid W_{t-1}, W_{t-2}, S_{t-1}, S_{t-2}, M_{t-1}, M_{t-2}) \quad (9)$$

This model is same as word only trigram model but along with previous words, the two previous morphological tags and stems are also used as parents. In this model the word factor $W_t$ is the child and the factors $W_{t-1}$, $W_{t-2}$, $S_{t-1}$, $S_{t-2}$, $M_{t-1}$ and $W_{t-2}$ are parents. The dashed lines represent deterministic variables. This model represents that, the probability of the next word can be predicted by using previous two words, two morphological tags and stems.

**Backoff and Smoothing**

Backoff is the method of obtaining probability distribution of n-grams when the higher order n-grams do not have sufficient probability mass. In other words backoff suggests that, when data is not sufficient to estimate a high order n-gram conditional probability table estimate only a portion of the table and construct the remaining table using a lower-order n-gram model. For example, when the trigram $p(W_t \mid W_{t-1},W_{t-2})$ count is not sufficient, move down to the bigram $p(W_t \mid W_{t-1})$. In this process, the probability mass is taken away from the higher order n-gram model and is distributed to the lower order n-gram model by maintaining valid probability distribution totalling to unity. The process is then recursively applied down up to unigram.

The backoff model for trigram $P_{BO}(W_t \mid W_{t-1}, W_{t-2})$ can be defined as given in equation 10.

$$P_{BO}(w_t \mid w_{t-1}, w_{t-2}) = \begin{cases} d_{N(w_t,\, w_{t-1}, w_{t-2})}\, P_{ML}(w_t \mid w_{t-1}, w_{t-2}) \\ \qquad \text{if } N(w_t,\, w_{t-1}, w_{t-2}) > \tau_3 \\ \alpha(w_{t-1}, w_{t-2})\, P_{BO}(w_t \mid w_{t-1}) \text{ otherwise} \end{cases} \quad (10)$$

It says that, trigram distribution is used if trigram count is greater than threshold value $\tau_3$ otherwise go to the bigram model after applying the discount function $d_{N(w_t,\, w_{t-1}, w_{t-2})}$ on the trigram distribution. $d_{N(w_t,\, w_{t-1}, w_{t-2})}$ is a number that is generally between 0 and 1. For discounting, different smoothing techniques can be used such as Good Turing smoothing, Kneser-Ney smoothing, absolute discounting, etc. The quantity $\alpha(w_{t-1}, w_{t-2})$ ensures that the entire distribution still sums to unity after applying backoff.

Figure 3 shows the backoff right from 4-gram to the unigram distribution. In this backoff model, if 4-gram sequences are not in sufficient amount then backoff to 3-gram sequences. If 3-gram sequences are also not enough then go for bigram count and same way up to unigram distribution.

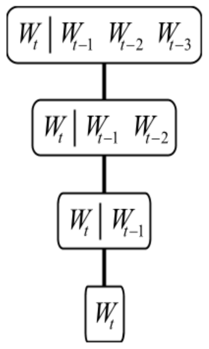
$W_t \mid W_{t-1}\ W_{t-2}\ W_{t-3}$
$W_t \mid W_{t-1}\ W_{t-2}$
$W_t \mid W_{t-1}$
$W_t$

Figure 3. Backoff graph for a 4-gram model.

**Generalized Backoff for FLM**

For standard n-gram language model, backoff procedure drops the word which is most distant in the contextual history of the next word to be predicted. Then the next most distant word is dropped and then the procedure goes on till the unique neighbouring previous word of the next word is dropped and a unigram distribution is obtained. In the graphical representation shown in figure 3, first the most distant parent node $W_{t-3}$ is dropped. Then the next distant node $W_{t-2}$ is dropped and so on till the immediate parent node $W_{t-1}$ is dropped. The graphical representation of the backoff procedure shown in figure 3 is called as backoff graph. The backoff graph in figure 3 shows a backoff path for a 4-gram language model. It starts from a node with all three previous words are present and goes down to the node with only single immediate previous word is present and then last unigram node.

But for factored language model, any factor of the current word can be predicted based on any combination of factors of the previous words in the contextual history of the word. As represented earlier in equation 7, the factored language model distributions are of the form:

$$\prod_k \mathrm{p}( f_t^1 \mid f_t^{1:k-1} , f_{t-n+1:t-1}^{1:k} )$$

For simplicity, it is represented as follows:

$$\mathrm{p}\ (\mathrm{F}/F_1, F_2, F_3 \ldots\ldots F_N) \qquad (11)$$

In FLM, due to availability of many factors and various combinations of the factors in the contextual history of the word, many options are available for dropping during the backoff procedure as shown in figure 4.

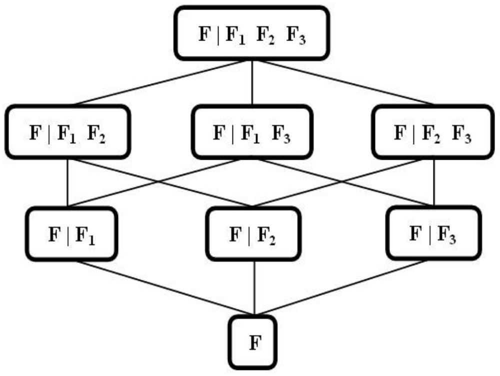

Figure 4. Backoff graph in FLM.

Figure 4 shows all possible backoff paths for a 4-gram factored model p (F / F1, F2, F3). Each path in the backoff graph is a distinct backoff model. Hence, which backoff path should be selected is a decision problem.

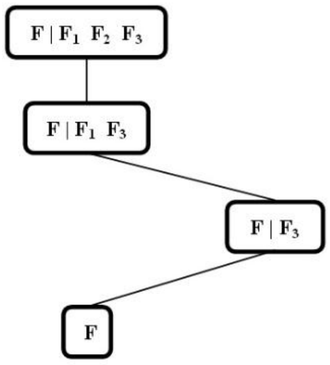

Figure 5. Backoff path in FLM.

Figure 5 shows only a specific backoff path in the backoff graph given in figure 4. The generalised backoff method solves this decision dilemma. Generalised backoff method suggests that instead of selecting a fixed path while going from higher order model to lower order model in a backoff graph, select multiple different paths dynamically at run time. The generalised backoff can be represented as in equation 12.

$$\mathrm{P_{GBO}}\,(\mathrm{f} \mid \mathrm{f_1}, \mathrm{f_2}, \mathrm{f_3}) = \begin{cases} \mathrm{d}_{\mathrm{N(f, f_1, f_2, f_3)}}\ \mathrm{P_{ML}}\,(\mathrm{f} \mid \mathrm{f_1}, \mathrm{f_2}, \mathrm{f_3}) \\ \qquad\qquad \text{if } \mathrm{N(f, f_1, f_2, f_3)} > \tau_4 \\ \alpha(\mathrm{f_1}, \mathrm{f_2}, \mathrm{f_3})\ \mathrm{g(f, f_1, f_2, f_3)} \quad \text{otherwise} \end{cases} \qquad (12)$$

Where, $g\big(f, f_1, f_2, f_3\big)$ is the non-negative backoff distribution. $\alpha(f_1, f_2, f_3)$ is used to make it sure that the entire distribution sums to unity. There are many possible ways how the function $g\big(f, f_1, f_2, f_3\big)$ can be calculated such as max normalized counts, min normalized counts, geometric mean, weighted mean, average mean and their variations. Each different way of calculation results into the different backoff strategy.

SRILM [6] language modeling toolkit supports both generalized backoff and factored language models. Essentially, the SRILM provides a way to implement graphical models like syntax in terms of code. The code syntax can contain the factor to be used as child node and other required factors as its parent node. It also provides a way to specify each possible node that can be dropped to determine the backoff path in the backoff graph. Different options can be specified at each node for smoothing, threshold value required for backoff decision making. The SRILM accepts a factored language data file and produces one count file containing counts of each n-gram and one language model file showing probabilities of n-grams. The language model is generated as per the specification given in the language model description file.

For building a standard n-gram language model the language data required is of the following form;

"Sham went to the school"

But for building factored language model the language data must be modified and factors must be added to the words in the data as given below.

"W-Sham:P-noun W-went:P-verb W-to:P-connective W-the:P-article W-school:P-noun"

Here, W represents word factor and P represents part of speech factor of the word.

The language model description file will contain specifications about what kind of language model is to be generated. It will have following format;

```
## Factored trigram model
W: 2  W(-1) W(-2) tri.count.gz tri.lm.gz 3
W1,W2  W2 ukndiscount gtmin 2 interpolate
     W1  W1 ukndiscount gtmin 1 interpolate
      0     0   ukndiscount gtmin 1
```

## represents the comment. Second line specifies child node and parent nodes along with the files to be used for data and language model. The integer 3 represents number of backoff nodes in the model. Third line onwards specifies the nodes in backoff path and their respective smoothing parameters such as algorithm, threshold value etc.

## III. Related Work

Factored language models were initially used for speech recognition of Arabic languages [3]. Now it is also widely used for language modeling component of phrase based machine translation and natural language generation. K. Kirchhoff, J. Bilmes, K. Duh in their tutorial described in depth about factored language models and its implementation using SRILM toolkit [4]. In [5], author utilized factored language model for machine translation application. In [9], how FLM can be used for Portuguese text generation is explained. In [10], authors used FLM for generating a Romanian language model using linguistic factors. In [11], [12], [13], authors show how morphological factors can be utilized for FLM generation. In [14], [15], authors utilized factored language model for speech recognition application.

## IV. Methodology

For applying factored language model for any natural language processing task, two important decisions need to be taken in advance. First, a valid set of factors to be used in FLM must be identified and second the best statistical language model must be selected out of many possible models which may produce highest perplexity on the test data. Hence, following factors were selected for the factored language modeling of Hindi language.

1. Word itself (शब्द)
2. Part of speech tag of the word (शब्द भेद)
3. Gender of the word (लिंग)
4. Number of the word (वचन)
5. Stem of the word (मूलशब्द)

For language model training, the training corpus was preprocessed by adding these factors in the data. The available corpus is divided in two parts for training and testing purpose. For building factored language model SRILM language modeling toolkit is used [7]. SRILM provides various methods of smoothing such as Good-Turing smoothing, Kneser-Ney smoothing, Witten-Bell smoothing. A corpus containing 603 words was used for training and a test set containing 44 words was used for testing purpose. Baseline n-gram language models were also generated to compare the results against factored language model using perplexity as a measure of comparison. Many experiments were carried out by writing various scripts satisfying factored language model specification format for testing different backoff paths and smoothing strategies.

## V. Results and Discussion

For evaluating the performance of the language model there are two main techniques, extrinsic evaluation and intrinsic evaluation. In extrinsic method, the developed language model is applied in a particular natural language application and the performance is evaluated. An intrinsic evaluation method is a metric which measures the performance of a model without applying it to any application. The intrinsic metric used to evaluate the performance of language model is based on probability and is termed as perplexity. As perplexity is the inverse of probability, the higher conditional probability of the word sequence produces lower perplexity. In other words minimizing perplexity leads to maximizing the test set probability for the language model. Hence, lower the perplexity, the better the language model is.

Table 1. Perplexity of individual models.

| Sr. No. | Model Description | Perplexity |
|---|---|---|
| 1 | 3-gram with Good-Turing smoothing | 35.3702 |
| 2 | 3-gram with Kneser-Ney smoothing | 37.4103 |
| 3 | 3-gram with Witten-Bell smoothing | 42.3994 |
| 4 | Factored Language Model with Good-Turing smoothing | 19.4299 |
| 5 | Factored Language Model with Kneser-Ney smoothing | 26.0954 |
| 6 | Factored Language Model with Witten-Bell smoothing | 19.5216 |

In table 1, the standard baseline n-gram model and factored language model are compared using perplexity as a measure. It is observed that, on the available test data, the perplexity of factored language model with Good-Turing smoothing is 19.4299 and is lowest in all the models. It reduces the perplexity by 45% over standard n-gram language model with Good-Turing smoothing whose observed perplexity was 35.3702. In all developed factored language models, interpolation is used to estimate probabilities at the backoff node. The gtmin of 1 and gtmax of 3 are used during training of factored language model with Good-Turing smoothing. As factored language model not only uses previous words but also the linguistic knowledge of the previous words for predicting the next word, this more information adds more robustness in the model. It is observed that the perplexity of factored models is about 40% less than standard n-gram language models.

## VI. CONCLUSION and Future Scope

This paper describes a method of language modeling which uses n-gram framework as a backbone methodology. The paper also discusses how grammatical and lexical features of the language can be used to predict the next word more accurately than a baseline n-gram language model using perplexity as a measure. The disadvantage of the model is that it requires pre-processing of data where factors need to be added in the data. Also, computational complexity gets increased due to many possible backoff paths availability. Future scope of the work includes identifying a group of factors which further improves perplexity of the model.

## REFERENCES

[1] R. Rosenfeld, "*Two decades of statistical language modeling: where do we go from here?*", In the Proceedings of the 2000 IEEE Intenational conferance, Vol. 88, Issue. 8 pp. 1270–1278, 2000.

[2] S. F. Chen, J. Goodman, "*An Empirical Study of Smoothing Techniques for Language Modeling*" , In the Proceedings of the 1996 Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, San Francisco, pp 310-318, 1996.

[3] J.A. Bilmes, K. Kirchhoff, "*Factored Language Models and Generalized Parallel Backoff* ", In the Proceedings of the 2003 HLT/NAACL, pp 4-6, 2003.

[4] K. Kirchhoff, J. Bilmes, K. Duh, "*Factored Language Models Tutorial*", University of Washington, 2016.

[5] A. E. Axelrod, "*Factored Language Models for Statistical Machine Translation* ", University of Edinburgh, 2006.

[6] A. Stolcke, "*SRILM- an Extensible Language Modeling Toolkit*", In the Proceedings of the 2002 International Conference on Spoken Language Processing, Denver, Colorado, September 2002.

[7] A. Stolcke, J. Wheng, W. Wang, V. Abrash, "*SRILM at Sixteen: Update and Outlook*", In the Proceedings of the 2011 IEEE Automatic Speech Recognition and Understanding Workshop, Waikoloa, 2011.

[8] K. Duh, K. Kirchhoff, "*Automatic Learning of Language Model Structure*", In the Proceedings of the 2004 International Conference on Computational Linguistics (*COLING)*, 2004.

[9] E. M. deNovais, "*Portuguese Text Generation Using Factored Language Models*", J. Brazilian Computation Society, Vol. 19, Issue. 2, pp 135–146, 2013.

[10] M. Lazˇar, D. Militaru, "*A Romanian Language Modeling Using Linguistic Factors*" , In the Proceedings of the 2013 7th Conference in Speech Technology and Human - Computer Dialogue (SpeD), Cluj-Napoca, , pp. 1–6, 2013.

[11] I. Kipyatkova, A. Karpov, "*Study of Morphological Factors of Factored Language Models for Russian ASR*", In the Proceedings of the 2014 SPECOM 2014, Novi Sad, pp. 451–458, 2014.

[12] H. Sak, M. Saraçlar, T. Güngör, "*Morphology Based and Sub Word Language Modeling for Turkish Speech Recognition*", In the Proceedings of the 2010 ICASSP, Dallas, pp. 5402–5405, 2010.

[13] A. Mousa, M. Shaik, R. Schlüter, H. Ney, "*Morpheme Based Factored Language Models for German LVCSR*", In the Proceedings of the 2011 INTERSPEECH, Florence, pp. 1053–1056, 2011.

[14] Z. Alumae, "*Sentence Adapted Factored Language Model for Transcribing Stonian Speech*", In the Proceedings of the 2006 ICASSP, Toulouse, pp. 429–432, 2006.

[15] T. Hirsimaki, J. Pylkkonen, M. Kurimo, "*Importance of High-Order N-Gram Models in Morph-Based Speech Recognition*", IEEE Trans. Audio, Speech, Lang. Process. , Vol. 17, Issue. 4, pp. 724–732, 2009.

[16] H. Adel, NT. Vu, K. Kirchhoff, D. Telaar, T. Schultz, "*Syntactic and Semantic Features for Code-Switching Factored Language Models*", IEEE/ACM Trans. Audio, Speech, Lang. Process, Vol. 23, Issue. 3, pp. 431–440, 2015.

## Authors Profile

Mr. A.R. Babhulgaonkar pursed Bachelor of Engineering from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India and Master of Technology from Dr. Babasaheb Ambedkar Technlogical University, Lonere, Maharashtra, India. He is currently pursuing Ph.D. and working as assistant professor in Department of Information Technology, Dr. Babasaheb Ambedkar Technlogical University, Lonere, Maharashtra, India since 2004. He is a lifetime member of ISTE since 2008. He has published 06 research papers in reputed international journals and conferences including IEEE. His main research work focuses on Natural Language Processing, Machine Learning. He has 15 years of teaching experience.

*S. P. Sonavane* pursed Ph.D. in 2010 in Computer Science and Engineering at Walchand College of Engineering, Sangli (Government aided autonomous institute) affiliated to Shivaji University, Kolhapur, Maharashtra, India. With two years of industry experience, she opted teaching as a career profession.

Her Ph.D. work is supported under young scientist, research scheme by Department of Science and Technology, New Delhi, India. Dr. Shefali received best teacher award in 2008 and is a member of many professional organisations.

Currently, she is working as an associate professor in Department of Information Technology at Walchand College of Engineering, Sangli. She has received research funds from DST and AICTE for various technical projects promoting work in the area of Computer Vision and Information Security. She has a good number of publications in journals and participation in conferences with few IPR credentials at her account. She has extended her research interest further in the fields of Machine Learning and Big Data. She is an active member towards the implementation of Outcome Based Education (OBE) in engineering with special efforts in revamping the teaching methodology and its assessment.