International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

A Statistical Model for Automatic Error Detection and Correction of Assamese Words

M P Bhuyan, S K Sarma

Abstract*: *Digitization of local languages is getting importance in the present scenario and the Language Processing task is also becoming popular among the Linguistic and IT people. It is very common that most of the people are comfortable with their native mother tongue. Writing of corrected word-form is also an important task in the digital platforms for the future existence of a language. In this research work, the Assamese language is taken as a Natural Language which is processed in the experiments. The Assamese language is one of the Indian languages and the research & development of the Assamese language is going on; from the computational point of view, Assamese is in the development phase. In Assamese, there are some similar characters which are phonetically same but their glyphs are different these characters or symbols often cause confusion to the users while writing, these types of characters are specially taken into consideration in this research work. A list of 14 confusing characters pairs of Assamese letters is taken for experimental purpose. In addition, this research work has focused on errors of Assamese words, which are checked by using bigram and trigram models. Moreover, the proposed model also tries to find the erroneous character which causes the incorrectness and shows the suggestions for that incorrect character. A score based system is designed for the Assamese characters and each character is assigned a score from their probability of occurrences by using bigram and trigram language models. Different types of experiments are performed to check the correctness of the Assamese words and the proposed model is able to check the correctness of the Assamese word with accuracy ranging from 81% to 86%. Error rate in Assamese can be reduced by using this model in any digital platform where a user can type in Assamese.

Index Terms*: Assamese language, Assamese word, bigram, probability, score, and trigram.*

I. INTRODUCTION

Assamese is a language recognized by Govt. of India and the language is spoken in the North-East region of India. Assamese is an Indo-Aryan language [1], it is the major language of Assam. There are approximately 20 million users are found in this language. Assamese is one of the morphologically rich language and having free word order, one word can exist in many forms by adding different suffixes. Speakers of Assamese language are also found in Arunachal Pradesh, Nagaland, and Meghalaya. In addition, the language was spread so widely that Assamese scripts were found in the Rakhine state of Myanmar and in the Pashupati temple of Nepal [2]. Assamese language is written from left to right, some sister languages of Assamese are: Bengali, Manipuri, Rajbansi, Bishunupriya, Sylheti, Rohingya, Maithili, etc. [2], Existence of Assamese language was found before the 7th century like the other eastern Indo-Aryan languages. In 17th century Assamese language was used a court language in the Ahom Kingdom. In Assamese, there are 11 vowels, 41 consonants, and 143-174 two-phoneme and 21-27 three-phoneme clusters of conjugate letters [3].

Since Assamese is an official language of Assam, there are many newspapers, magazines, books, etc. are printed in Assamese and also, in the digital platforms like Facebook, WhatsApp, YouTube people can write in Assamese by using the Assamese font, it is seen that people are writing knowingly or unknowingly some erroneous words/sentences in those social media platforms which may lead to a serious problem for the originality of the Assamese language. So, it is necessary to enrich the language by designing various computational tools for the further betterment of the Assamese language in the digital world. For the development of digital contents of Assamese language, it is utmost important to check the correctness of the written words. This work is a sub-part of spell checker, spell-checkers are used to detect errors and suggest word by applying method like minimum edit distance, n-gram, etc. In this proposed method, erroneous words of Assamese are checked in character level and if any error is found in the word then possible suggestions for the examined letter/character which causes error in the word are produced and the proposed model can identify the point where the error has occurred and this proposed model can show the correct part/portion of the erroneous word, such feature will help the learners of the Assamese language. This kind of feature is not found in the present error detection and correction models. In addition, the present error detection and correction models report the whole word as erroneous and suggest the relevant words to the user and the user has to find the error point of the word. Remaining sections of the paper is organized as follows:

Section II describes the related works in this research field, Section III explains the proposed model, Section IV analyzes and discusses the experimental results, Section V concludes the present work and shows the future path of this research work.

Revised Manuscript Received on 30 July 2019.

* Correspondence Author

M P Bhuyan*, Department of Information Technology, Gauhati University, Guwahati, India.

S K Sarma, Department of Information Technology, Gauhati University, Guwahati, India.

II. RELATED WORKS

In [3], LuitPad a Unicode based Assamese writing software was designed. The authors have mentioned two main typing options, one was on the basis of the sounds of the words and the second one was on the basis of sounds of the characters. LuiPad was enabled with on-line spell checking facility and also suggested the user with relevant words for the erroneous words. In LuitPad, Assamese characters or words were retrieved using the English characters with a phonetic mapping of the Assamese characters. LuitPad was having two input modes one was character by character and the second one was word by word. For spell checking, a list of 60,000 words dictionary was used and a dictionary search was performed for every word of the user. The suggestions were generated for the erroneous word by calculating the Levenshtein distance using a dynamic programming approach. Spell checker accuracy was around 93% and for the new user, it was effectively user-friendly.

In [4], a model was proposed to check the correctness of Bangla words using N-gram based models like bigram, trigram, and quadgrams. Authors have designed a corpus of the size of around 1 million words which was used to train their system. For each input word, they have calculated the probability in bigram, trigram and quadgram model and if the probability of the word was greater than a predefined threshold value then the system declared the word as correct; otherwise wrong. They have collected 50K valid/correct words from the Bangla e-newspapers. They have performed two types of experiments on the 50K data set. In their first experiment, they have tested all the correct words using bigram, trigram and quadgram model and the accuracies values obtained were 95.20%, 95.12% and 95.26% in bigram, trigram and quadgram model respectively. In the second level experiment, they have introduced some errors in the 50K words by inserting, deleting, substituting one character; accuracies values obtained in their second experiment were 97.13%, 97.14% and 97.16% for bigram, trigram and quadgram model respectively. The error rate of their proposed model was 3.83%, authors have explained that this error rate was due to the size of the training corpus, which could be enhanced by adding more data on the training corpus.

In [5], authors have designed a spell-checker for Assamese language and integrated the spell-checker to Open Office Writer. Authors have used a dictionary of root words whose size was 16K to check the spelling. Suggestions for misspelled words were calculated by using the minimum edit distance technique. For the result analysis, authors have calculated precision and recall and compared their results with Hunspell. The results obtained by the authors were below 70%.

In [6], authors have tried to find the real-word errors in the Assamese language. They have performed experiments on context-sensitive spell checking using n-gram based model like bigram and trigram with add one smoothing technique. Their model detects real-word errors and generates suggestions for possible corrections. Authors have designed a confusion set for each word by doing minimum edit distance operations and the words formed in minimum edit operations were compared with standard dictionary words so that only valid words were kept as a suggestion in the suggestion list. Finally, the probability of the sentence was calculated by placing different words of the confusion set and the list of words giving the highest sentence probabilities was displayed as a suggestion. Performance of the spell checker was 76% against a corpus size of 220,743.

In [7], authors have designed a word prediction model for Assamese language using bigram, trigram and quadrigram models for spontaneous sentence completion in Assamese. In their prediction model if a user wrote a character on the editor window then the prediction system starts predicting the word related to that character and once a word was written, the predictive model has tried to show the next word as a suggestion. Authors did two different types of experiments to evaluate the performance of their system, one was on the basis of pre-configured data and the other was on user input data. They have recorded the keystroke saving value for each of the sentence during the writing. The maximum level of keystrokes saving in pre-configured data was 74.04% and in user input data was 48.28%.

In [8], authors have designed a method to detect and correct real-word error using bigram and trigram in Bangla language. They have proposed a localized real word error detection and correction model where the scores of the left and right neighbors of the target/erroneous word along with the trigram formed by considering the left and right neighbors were calculated. They have assumed a single character error i.e. a word W is erroneous then the actual correct word will be found in a set S which is formed by doing a single edit operation on W. A weighted combined score of bigram and trigram were calculated for all the members of the set S. The words of the set S were ranked in decreasing order of their score. Using a rule-based approach and the rank of the word, error-decision was taken on any target word and necessary suggestions were generated accordingly. The error detection and correction model was able to give comparable accuracy with the other existing models. Since only left and right neighbors multiple errors could also be detected and corrected in their score calculation process.

In [9], authors have designed a method to detect real-word errors in Bangla language; they have chosen homophonic real word error. They have prepared a set of confusing words of homophones and seven different corpora for testing their model. Bigram and trigram models were used for checking the word errors and using bigram and trigram models they have calculated the probability for the final decision about the candidate word. A million words were tested in their model and the accuracy obtained was above 96%. In their proposed model, they have considered up-to two left and two right neighbors of any candidate word and evaluated a score for each candidate word by combining these left and right neighbors.

In [10], authors have studied different types of error detection and correction techniques in various Natural Languages.

They have shown the accuracies of the various spell-checkers and also highlighted the error detection techniques like N-gram models, Dictionary lookup, Edit distance, etc. They have raised the importance and presence of an efficient spell-checker for the Assamese language.

In [11], authors have designed a dictionary-based method for non-word error detection and correction in Assamese which they integrated to Microsoft Office. To detect the errors, every word was searched in the dictionary if the word was found in the dictionary then the word was correct otherwise wrong. They have used minimum edit distance operations for generating the suggestions of the misspelled words in their error detection and correction model.

In [12], authors have designed a spell-checker for Assamese. Initially, the spell-checker searched the dictionary for every word of the user and if the search failed to find the user's word in the dictionary a suggestion list was generated by using non-word error detection module, Soundex code generation module and Isolated-word error correction module. The dictionary which was used for error detection and correction was consisting of Soundex code of the Assamese words along with word meaning, pronunciation, and grammatical category. Soundex code of the Assamese words was able to speed-up the spell-checking process.

In [13], authors have designed a spell-checker for Khasi language. Khasi language is used in Meghalaya, a neighbor state of Assam. They have designed spell-checker for the typographic error of Khasi words i.e. the Non-word errors. They have detected the errors by using string cosine similarity and a dictionary.

In [14], authors have done a study on different spell-checking techniques for Indian languages. They have mentioned about the typographic errors (non-word error) and cognitive errors (real-word errors). In case of error detection, they have explained about the n-gram based model and dictionary-based approach and to correct errors. They have focused on edit distance, n-gram, rule-based, similarity keys, etc. The authors' aim was to design a spell-checker for Dogri and Urdu language.

From the above discussion, it is seen that development in the Assamese language is going on along with the other languages and a few Assamese spell-checkers were designed but their availability is very poor and support is also not found. In addition, increasing demand for social media platforms and rapid use of local languages in social media platforms results in transferring the local languages from the printed document/speech to digital platforms. During this language transfer, the major problem is erroneous writing by some users who have little knowledge about the language. So, it is necessary to make the software intelligent enough to take care of such erroneous writing. Keeping in mind the design of intelligent software, it is necessary to work on spell-checker for the Assamese language with an enhanced and better way by adding various intelligent features in the software. The present proposed model is described in *Section III* which is a new approach towards the development of spell-checker in the Assamese language.

III. PROPOSED MODEL

N-gram model is a type of statistical model which tries to predict the next event by observing the earlier events. This model is extensively used in Speech Processing, Natural Language Processing, etc., let us understand the n-gram model with an example:

$$sent = Please\ come\ \underline{\quad\quad}.$$

In the above sentence '*sent*' anyone can fill the blank space by writing the word '*here*'. This is possible because of the previous words, i.e. the earlier words or history. The working of n-gram model is also similar to this example. On the other hand, in computer, n-gram cannot be implemented as the computers are having limited memory so the n-gram is reduced to unigram, bigram, trigram, 4-gram, etc. Unigram model knows only the present event and there is no previous history in unigram model, Bigram model is one which can remember only one previous event or word, similarly, trigram model can remember two previous words and 4-gram or quadgram or quadrigram model can remember three previous words and so on. In this research, character level n-gram model is implemented where every word is taken into consideration and the word is broken into its characters and probabilities of the characters are calculated. Let us understand the character level n-gram model, for example, consider the word $W =$ *"bharat"*

$$\text{Unigram} = ['b','h','a',r','a','t']$$
$$\text{Bigram} = ['bh','ha','ar',ra','at']$$
$$\text{Trigram} = ['bha','har','ara',rat']$$

From the characters probabilities, it is possible to derive the word probability. Let us consider the word W containing $c_1c_2c_3c_4...c_n$ characters, probability of W can be calculated as follows:

In the Bigram model:

$$p(c_1c_2c_3...c_n) = p(c_1)\times p(c_2\,|\,c_1)\times p(c_3\,|\,c_2)\times...\times p(c_n\,|\,c_{n-1})$$

In the Trigram model:

$$p(c_1c_2c_3...c_n) = p(c_1)\times p(c_2\,|\,c_1)\times p(c_3\,|\,c_1c_2)\times...\times p(c_n\,|\,c_{n-2}c_{n-1})$$

A. Confusion set

In Assamese, there are some letters which are pronounced in the same way but their glyphs are different, such types of letters are kept in confusion set and special care has been taken. The following Table I shows these confusion letters.

TABLE I. SET OF CONFUSION LETTERS

Sl No.	Similar letters/ markers				pronunciation_in_word
	Letters-1	*Letter_Name*	*Letters-2*	*Letter_Name*	
01	□	Hroswo-i	□	Dirgho-i	i
02	□	Hroswo-u	□	Dirgho- u	u

03	□	Prothom-so	□	Ditiya-so	cha
04	□	Borgiya-jo	□	Ontosto-jo	ja
05	□	Murdhoniya-ta	□	Dontiya-ta	ta
06	□	Murdhoninya-tha	□	Dontiya-ta	tha
07	□	Murdhonaya-da	□	Dontiya-da	da
08	□	Murdhanya-dha	□	Dontiya-dha	dha
09	□	Murdhonoya-na	□	Dontiyana	na
10	□	Murdhanoya-xo	□	Taloibya-xo	xo
			□	Dontiya-xo	
11	□	Taloibya-xo	□	Murdhanoya-xo	xo
			□	Taloibya-xo	
12	□	Dontiya-xo	□	Taloibya-xo	xo
			□	Murdhanoya-xo	
13	□	horswoikar	□	dirghoikar	i_marker
14	□	hroswoukar	□	dirghoukar	u_marker

B. N-gram Formation

To check the error in a word or correctness of a word, left bigram, right bigram and trigrams are calculated for each letter of an Assamese word. For example if the character is c_i then left bigram, right bigrams and trigrams are generated as follows:

Left bigram: $c_{i-1}c_i$

Right bigram: $c_i c_{i+1}$

Trigram: $c_{i-1}c_i c_{i+1}$

Occurrences of each bigram and trigram are counted and recorded. Probabilities are calculated as follows:

$$p_1(c_i \mid c_{i-1}) = \frac{count(c_{i-1}c_i)}{count(c_{i-1})} \quad (1)$$

$$p_2(c_i \mid c_{i+1}) = \frac{count(c_{i+1}c_i)}{count(c_i)} \quad (2)$$

$$p_3(c_i \mid c_{i-1}c_{i+1}) = \frac{count(c_{i-1}c_i c_{i+1})}{count(c_{i-1}c_{i+1})} \quad (3)$$

In the equation (1) probability of the character is calculated by using the left bigram and in equation (2) probability of the same character is calculated by using the right bigram and in the equation (3) probability of the character is calculated by using the trigram. Finally, the probabilities calculated in the equations (1), (2) and (3) are combined and weighted combinations of these three are calculated and a score is assigned to each character. Following equation describes the evaluation of the score of an Assamese character.

$$Score(c_i) = \mu_1 p_1(c_i \mid c_{i-1}) + \mu_2 p_2(c_i \mid c_{i+1}) + \mu_3 p_3(c_i \mid c_{i-1}c_{i+1}) \quad (4)$$

The coefficients µ1, µ2, and µ3 can be calculated by doing some trial and error method, finally, it is found that if the contribution from higher-order n-gram i.e. trigram is more than the other two bigrams then score is more stable; since the higher-order n-grams are more sensitive on the context. Following relationship gives a relatively good result in the present proposed model.

$\mu_1 = 0.2$, $\mu_2 = 0.2$ and $\mu_3 = 0.6$

$Score(c_i)$ is also lie in the range 0 to 1.

C.Correctness Checking Procedure

Let us consider the following assumptions:

c_i = Any character in the word W.

CFS = Confusion set of Assamese characters as shown in Table I.

as_alpha_list = Assamese alphabet list.

c_i^p = Alternate letter(s) in the confusion set.

S_L = Suggestion list of the characters.

B_i = Bigram corpus.

T_i = Trigram corpus.

\# = comment line

Following algorithm describes the correctness of any Assamese word:

Algorithm-1:

Step-1: if $Score(c_i) = 0$

Step1.1: if c_i € *CFS*

Step 1.1.1: CALL S_L = *get_Score_list(CFS)*
 # Algorithm-2 called

Step 1.2: else

Step 1.2.1: Declare word is NOT correct at the character c_i

Step 1.2.2: my_list = as_alpha_list – $(c_i + CFS)$
#considering the remaining characters

Step 1.2.3: CALL S_L= *get_Score_*list(*my_list*)
 # Algorithm-2 called

Step-2: display S_L

Algorithm-2:

Step-1: get_Score_list(*my_list*):

Step 1.1: for each c_i in *my_list*

Step 1.1.1: Calculate the *score*(c_i) using equation (4)

Step1.1.2: Arrange my_list in decreasing order of the score of c_i and store in the suggestion list S_L

Step-2: return S_L

IV. RESULT AND DISCUSSIONS

For the experimental purpose, a corpus of size around 50K valid Assamese words is used. Bigrams and Trigram of the letters of these words were calculated. Errors were introduced in the words either by replacing the confusing character or inserting a new character or substituting by another character. For each word, the error was introduced by editing a single character. The model was tested for erroneous words ranging from 1000 to 5000 and the performance of the model is shown in *Section B*.

A. Experimental Results

Table II describes a portion of some confusing letters results and their score.

TABLE II. ASSAMESE WORDS CORRECTNESS CHECKING SCORES AND SUGGESTIONS FOR CONFUSION SET LETTERS

Sl No.	Word	Erroneous character and score	Suggested character and score	Correct word
01	□□□	['□', '0']	['□', '0.042']	□□□
02	□□□□	['□', '0']	['□', '0.008']	□□□□
03	□□□	['□','0']	['□', '0.33']	□□□
04	□□□□□	['□', '0']	['□', '0.074']	□□□□□
05	□□□□□	['□', '0']	['□', '0.22']	□□□□□
06	□□□	['□', '0']	['□', '0.036']	□□□
07	□□□□	['□', '0']	['□', '0.0022']	□□□□
08	□□□□□□	['□', '0']	['□', '0.62']	□□□□□□
09	□□□□	['□','0']	['□', '0.00051']	□□□□
10	□□□□□□□	['□□' ->'□', '0']	['□□'->' □', '0.42']	□□□□□□□

From the above Table II, it is seen that the proposed model is able to generate the relevant suggestions for the confusion set letters.

Table III demonstrates some non-confusion set letters and the suggestions generated for those letters.

TABLE III. ASSAMESE WORDS CORRECTNESS CHECKING SCORES AND SUGGESTIONS FOR NON-CONFUSION SET LETTERS

Sl No.	Word	Erroneous character and score	Suggested character and score	Correct word(s)
01	□□□□□	['□', '0']	['□', '0.55'] ['□','0.36'] ['□', 0.09]	□□□□□, □□□□□, □□□□□
02	□□□□□	['□', '0']	['□', '1']	□□□□□
03	□□□□□	['□','0']	['□', '1']	□□□□□
04	□□□□□	['□', '0']	['□', '0.61'] ['□', '0.21'] ['□', '0.18']	□□□□□, □□□□□, □□□□□
05	□□□□□	['□', '0']	['□', '0.71'] ['□', '0.29']	□□□□□, □□□□□

From Table III it is cleared that the proposed model is detecting and correcting the errors and producing relevant suggestions.

B. Performance Evaluation

The performance of the system is evaluated by using the following parameters:

True Positive (TP), True Negative (TN) False Negative (FN) and *False Positive (FP)* detection by the model.

Now, *Precision, Recall and Accuracy* can calculated as follows:

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP} \tag{7}$$

Following Table IV shows the performance of the present proposed model on increasing the number of erroneous words.

TABLE IV. PERFORMANCE OF THE PROPOSED MODEL

Sl. No.	No. of Erroneous Words	Precision	Recall	Accuracy
1	1000	90.36%	87.72%	81.50%
2	2000	93.37%	89.86%	86.15%
3	3000	92.32%	90.07%	84.50%
4	4000	93.06%	89.83%	84.90%
5	5000	91.52%	92.55%	85.80%

Following Fig. 1 shows the graphical representation of the results of Table IV. From Fig. 1, it is clearly visible that accuracy and precision are highest in erroneous words size 2000. Recall is highest in erroneous words size 5000.

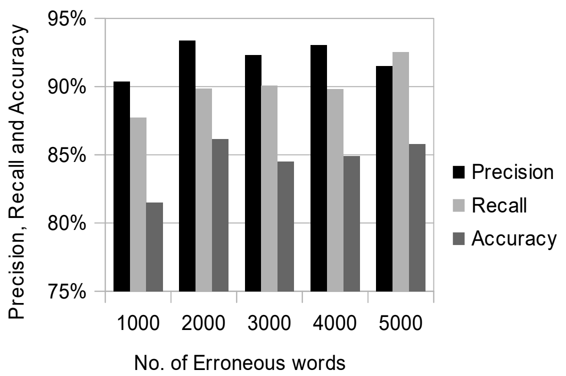

Fig. Change of *Precision*, *Recall* and *Accuracy* against the No. of Erroneous words.

V. CONCLUSION AND FUTURE WORKS

Error detection and correction is important in any typing tool and this is the preliminary task for further development of the language in computational field or digitization of the language. This research work has focused on the confusing letters or makers of Assamese language along with the other Assamese letters or characters; because in most of the cases, people often do spelling mistakes with these confusing letters. After doing the different level of testing it was found that the proposed model not only detects the erroneous words but also the point where the error has occurred in an efficient way.

Moreover, the performance measuring parameters like *Precision, Recall, and Accuracy* are relatively stable during the experiments with the erroneous words ranging from 1000 to 5000.

This research work will definitely help the Assamese medium school teachers to introduce a topic like Language Learning using Computer as a practical component in their Curriculum. Such type of model will produce a thrust among the students to learn the language efficiently and correctly. Besides, other users of the language will get benefit from this research work. The overall performance of the system is quite satisfactory on moderate data size.

In the future, to enhance the experimental results, corpus size can be increased. Secondly, more than one character errors can be explored and analyzed. Thirdly, the coefficients $\mu 1$, $\mu 2$ and $\mu 3$ of the equation (4) can be analyzed more precisely and deeply. Lastly, the same model can be applied to other Indian languages.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers of this manuscript and MeitY Govt. of India for the Visvesvaraya PhD Scheme.

REFERENCES

1. G. C. Goswami, "STRUCTURE OF ASSAMESE," *Dept. of Publication, Gauhati University*, Guwahati, 1982.
2. Wikipedia: The free encyclopedia. (2019, June 26). *Assamese Language*, Inc. Retrieved July 1, 2019, from https://en.wikipedia.org/wiki/Assamese_language#cite_note-17
3. N. Saharia, K. M. Konwar, "LuitPad: a fully unicode compatible Assamese writing software," in *Proceedings of the 2nd Workshop on Advances in Text Input Methods (WTIM 2)*, Mumbai, India, 2012, pp. 79-88.
4. N. H. Khan, G. C. Saha, B. Sarker and H. Rahman, "Checking the Correctness of Bangla Words using N-Gram", *International Journal of Computer Applications*, 89:11, 2014, pp. 1-3.
5. K. Kashyap, H. Sarma and S. K. Sarma, "Luitspell: Development of an Assamese Language Spell Checker for Open Office Writer," *European Journal of Advances in Engineering and Technology*, 2:5, 2015, pp. 135-138.
6. R. Choudhury, N. Deb, K. Kashyap, "Context-Sensitive Spelling Checker for Assamese Language," In: J. Kalita, V. Balas, S. Borah, R. Pradhan(eds) *Recent Developments in Machine Learning and Data Analytics. Advances in Intelligent Systems and Computing*, vol 740. Springer, Singapore, 2019, pp. 177-188.
7. M. P. Bhuyan and S.K. Sarma, " An N-gram based model for predicting of word-formation in Assamese language," *Journal of Information and Optimization Sciences*, 40:2, DOI: 10.1080/02522667.2019.1580883, 2019, pp. 427-440.
8. P. Samanta and B. B. Chaudhari, "A simple real-word error detection and correction using local word bigram and trigram," *Proceedings of the Twenty-Fifth Conference on Computational Linguistics and Speech Processing (ROCLING 2013)*, 2013, pp. 211-220.
9. M. Rana, M. T. Sultan, M. F. Mridha, E. A. Khan, M. Ahmed and A. H. Hamid, "Detection and Correction of Real-word Errors in Bangla Language," *International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018, pp.__.
10. S. Kalita, R. R. Deka, M. P. Bhuyan, S. K. Sarma, "Real – Word and Non – Word error detection & correction in NLP: A Survey," *Proceedings of TEQIP III sponsored Joint National Conference on Emerging Computing Technologies & its Applications,* 2019, pp. 209-212.
11. B. Sarma, D. Goswami and G. Goswami, "Assamese Spell Checker Design and Implementation," *International Journal of Modern Trends in Engineering and Research*, 03:02, 2016, pp. 44-47.
12. M. Das, S. Borgohain, J. Gogoi and S. B. Nair "Design and Implementation of a Spell Checker for Assamese," *Proceedings of the Language Engineering Conference (LEC'02)*, 2002.
13. M. S. Nonghuloo and K. Krishnamurthi, " Spell Checker for Khasi Language," *International Journal of Software Engineering,* 2017, pp. 1-12.
14. R. Kumar, M. Bala and K. Sourabh, "A study of spell checking techniques for Indian Languages," *JK Research Journal in Mathematics and Computer Sciences,* 1:1, 2018, pp. 105-113.

AUTHORS PROFILE

M P Bhuyan is a research scholar in the department of Information Technology, Gauhati University, Guwahati, India. He did BE in Computer Science & Engineering and MTech in Information Technology. His research areas are Artificial Intelligence, Natural Language Processing, and Machine Learning.
Email: mpratim250@gmail.com

S K Sarma is currently working as a professor in the department of Information Technology, Gauhati University, Guwahati, India. He is a computer scientist and published various research papers in renowned conferences and journals. His research areas are Natural Language Processing, Artificial Intelligence, Machine Learning, Computer Networks, etc. He has guided many PhD students.